A Modal-Based Real-Time Piano Synthesizer

Balázs Bank, Stefano Zambon, and Federico Fontana, *Member, IEEE*

***Abstract*—This paper presents a real-time piano synthesizer where both the transverse and longitudinal motion of the string is modeled by modal synthesis, resulting in a coherent and highly parallel model structure. The paper applies recent developments in piano modeling and focuses on the issues related to practical implementation (e.g., numerical stability, aliasing, and efficiency). A strong emphasis is given to modeling nonlinear string vibrations, and a new variation of earlier synthesis techniques is proposed which is particularly well suited for modal synthesis. For soundboard modeling, the possibilities of using fast Fourier transform-based fast convolution and parallel second-order filters are discussed. Additionally, the paper describes the details of the software implementation and discusses the computational complexity of each model block. The piano model runs on current computer hardware with full polyphony in real time.**

***Index Terms*—Modal synthesis, physics-based sound synthesis, piano.**

I. Introduction

THE piano is a particularly important instrument for sound synthesis applications because of many reasons. First, the piano has a large prominence in western music, while it is large, heavy, and expensive. Thus, there is a great demand for an electronic substitute for home or stage use. Second, because the piano is a keyboard instrument, its control interface is relatively simple, unlike that of the guitar or the violin. That is, there is no physical obstacle on the control side to fulfill this demand.

Most of the current digital pianos are based on sample playback. Because of the simple interface (the player controls the key velocity only), a high level of realism can be achieved. The variation of the sound as a function of key velocity can be taken into account by linear filtering or by crossfading between samples recorded at different dynamic levels. However, the dynamic interactions of the different parts of the piano, like the restrike of the same string or the coupling between the strings, cannot be faithfully reproduced.

The remedy of the problems of sampling synthesis is the use of physical modeling, where, instead of reproducing the sound of the instrument, the entire sound production mechanism is modeled. That is, a complete virtual instrument is running on the computer or dedicated hardware. Because of the very nature of physical modeling, the interaction of different parts (such as the coupling of strings) is automatically modeled. Moreover, the user (player or sound designer) controls a relatively small set of meaningful parameters similar to those of real pianos, like hammer hardness, string mistuning, etc.

The first step of physics-based sound synthesis is to understand how the instrument works, that is, the equations describing the main parts of the instrument have to be developed and the interactions of the different parts have to be revealed. However, the resulting computational complexity of numerically solving an instrument model that incorporates all the details we know about a specific instrument is usually too high for real-time implementation. Therefore, some simplifications have to be made. This is usually done by neglecting some perceptually less relevant phenomena in such a way that this should not lead to significant audible differences compared to the full model. Next, efficient signal processing algorithms are developed for computing the solution of this simplified physical model.

The first physics-based instrument model was presented as early as 1971 by Hiller and Ruiz [1], [2]. Interestingly, the first piano model has followed 16 years later, which was the digital waveguide-based piano of Garnett [3]. In 1995, Smith and Van Duyne [4], [5] proposed a piano model based on commuted synthesis. As part of a collaboration between the University of Padova and Generalmusic S.p.A., Borin *et al.* [6] presented the first complete real-time piano model in 1997. Bank [7] introduced a similar physical model in 2000, with slightly different implementation, and its real-time version was implemented in 2002. A concise overview of these two models and some additional results have been presented in [8]. In the recent years, several improvements were proposed by Bensa and coworkers, concentrating on the modeling of coupled piano strings and on a parametric excitation model [9]–[11]. Bank has examined the generation of longitudinal motion in piano strings and proposed various modeling methods [12], [13]. Rauhala and their colleagues have concentrated on improved parameter estimation techniques and efficient, parametric implementations [14], [15]. We have started the implementation of our real-time modal-based piano synthesizer in 2007 as a part of collaboration between Viscount International and Verona University, and this forms the topic of the present paper.

Although physics-based piano synthesis has a more than two decades tradition in academic research, it has been applied in commercial products only very recently, due to its relatively high computational cost. A software piano, Pianoteq, was introduced by Modartt in 2006 [16], and the first digital piano employing physical modeling was presented by Roland in 2009 [17]. It is expected that physical modeling will be the current trend in digital pianos.

Manuscript received April 01, 2009; revised December 03, 2009. Current version published April 14, 2010. This work was supported by the joint project of Viscount International SpA and Verona University. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Julius Smith.

B. Bank was with the Department of Computer Science, Verona University, 37134 Verona, Italy. He is now with the Department of Measurement and Information Systems, Budapest University of Technology and Economics, 1111 Budapest, Hungary (e-mail: bank@mit.bme.hu).

S. Zambon and F. Fontana are with the Department of Computer Science, Verona University, 37134 Verona, Italy. (e-mail: stefano.zambon@univr.it; federico.fontana@univr.it).

Digital Object Identifier 10.1109/TASL.2010.2040524

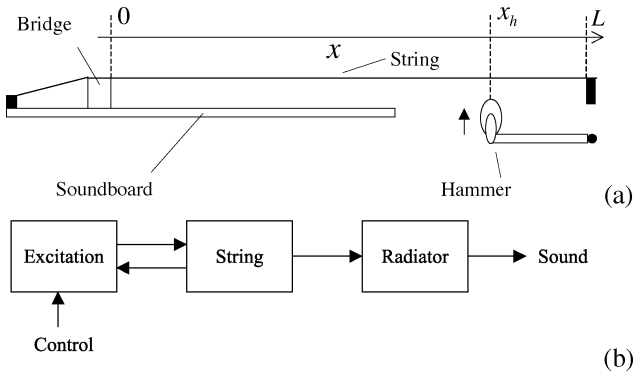

Fig. 1. General structures. (a) Schematic representation of the instrument with the coordinate system for string modeling, (b) Basic model structure.

In this paper, we describe a complete piano synthesizer based on modal synthesis. Concerning each model block, the chosen approach is briefly compared to other possibilities, and its advantages are discussed. A strong emphasis is given to the connection of the different blocks and on practical issues related to numerical stability, aliasing, and efficient implementation. For modeling longitudinal vibrations, a new variation of previous approaches is proposed which is particularly well suited for modal synthesis. Finally, the real-time software implementation is discussed, and the computational complexity of the different blocks is estimated.

II. Model Structure

Since the physical modeling approach simulates the sound production of the instrument, the blocks in the piano model resemble the parts of a real piano, as displayed in Fig. 1. The first model block is the excitation, the hammer strike exciting the string. The string determines the fundamental frequency of the tone and its quasi-periodic output signal is filtered through a postprocessing block, covering the radiation effects of the soundboard. Fig. 1(b) shows that the hammer-string interaction is bidirectional, since the hammer force depends on the string displacement. The string–soundboard coupling is also bidirectional in real instruments, because the soundboard, besides radiating the sound, provides a terminating impedance to the string. However, since the impedance of the soundboard is around a thousand times larger than that of the string, the only significant effect of the string–soundboard coupling is a change in the modal frequencies and decay times of the string. Therefore, it is customary to use a simplified modeling where the "impedance effects" of the coupling, i.e., the variation in the modal frequencies and decay times of the string, are taken into account by changing the parameters of the string block, and only the "radiation properties" are modeled in the radiator block.

Fig. 1 displays only the main parts of the instrument and the corresponding synthesis model—the blocks for modeling secondary effects are omitted for clarity. The full piano model is displayed in Fig. 2. The functions of the various blocks will be explained later in the corresponding sections.

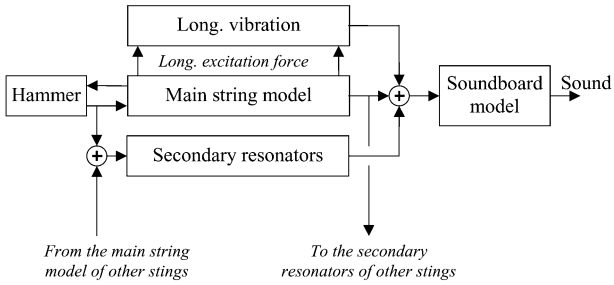

Fig. 2. Detailed model structure.

III. Main String Model

In reality, the string vibrates in two transverse and one longitudinal directions. First we consider the vibration of the main transverse polarization $y(x,t)$. The stiff and lossy string equation is

$$\mu\frac{\partial^2 y}{\partial t^2} = T_0\frac{\partial^2 y}{\partial x^2} - ES\kappa^2\frac{\partial^4 y}{\partial x^4} - 2R\mu\frac{\partial y}{\partial t} + d_y(x,t) \qquad (1)$$

which is the ideal wave equation extended by terms describing the stiffness and losses of the string [18]. The tension of the string is denoted T_0, and μ is the mass per unit length. The stiffness of the string is characterized by $ES\kappa^2$, where E is the Young's modulus, S is the cross-section area of the string, and κ is the radius of gyration. The internal losses of the string and the losses due to the impedance effect of the soundboard are approximated by the frictional resistance R. Note that in real instruments, losses are frequency dependent. This effect will be taken into account later in (3). External driving forces are included in the excitation force density $d_y(x,t)$, which has the dimension of force per unit length.

Equation (1) can be numerically solved in many ways; see [19] for an overview of the methods used in sound synthesis applications. We summarize only three commonly used approaches. The first is finite difference modeling, which is based on approximating the derivatives of the wave equation (1) by differences, leading to a difference equation that can be easily solved numerically. Finite difference piano models include [20]–[22]. Finite-difference modeling is especially well suited for modeling complex structures (such as the piano soundboard), where a closed-form solution cannot be derived.

However, for simple linear systems, such as the one-dimensional wave equation (1), the closed-form solution is known, and the computational complexity can be greatly reduced if the continuous-time solution is discretized instead of the wave equation itself. Digital waveguide modeling, introduced by Smith [23], [24], discretizes the traveling wave solution of the ideal wave equation. The losses and dispersion of real strings are lumped to one specific filter in the model. Thus, the full string model reduces to a delay line and a relatively low-order IIR filter in a feedback loop. Most of the piano models intended for sound synthesis applications [3], [6], [7], [10], [15] are based on the digital waveguide technique due to its efficiency.

In modal synthesis, the modal solution of the wave equation is discretized [25], [26]. Thus, the string is modeled by a set of second-order resonators, each implementing the response of one particular mode. The modal approach has been already used

for the simulation of nonlinear vibrations in piano strings [12], [13]. This paper presents a complete piano model that is based on modal synthesis.

The choice of the modeling method for the string has been made after carefully considering the advantages and drawbacks of digital waveguides and modal synthesis. The main difference of our piano model from the previous ones presented in the literature is the ability to model nonlinear string vibrations. Thus, the usual choice of digital waveguide modeling had to be revised. It has been shown in [13] that precise modeling is not possible with digital waveguides in their efficient form, because the modal shapes of the string model become non-orthogonal due to the lumped dispersion filter. Either the dispersion filter has to be distributed, increasing the model complexity significantly, or, some less physical "tricks" have to be used [13]. On the other hand, modeling the nonlinear vibrations in piano strings is very straightforward with modal synthesis, as will be shown in Section V.

While the computational complexity of our modal piano model is probably higher compared to earlier, digital-waveguide based piano models that neglect nonlinear string vibrations, it is still efficient enough to consume only one third of the resources of a today's PC, which we find an acceptable compromise. This efficiency can be achieved in spite of the relatively high number of required arithmetic operations because modal synthesis results in a fully parallel structure and simple code that is easier to optimize (see Section VIII for details). An additional advantage of modal synthesis is greater flexibility, since all the parameters of the partials can be set independently. This leads to simpler parameter estimation and the possibility of modifying the parameters in real-time.

A. Continuous-Time Equations

The modal solution of the string equation can be found in various acoustics textbooks. Nevertheless, summarizing it here is still instructive and helps understanding the different parts of the model. The following derivation is a slightly modified and more concise version of [13, Ch. 2].

If the string is rigidly terminated at $x = 0$ and $x = L$ with hinged boundary conditions, as displayed in Fig. 1(a), the string shape $y(x,t)$ can be expressed by the Fourier-like series

$$y(x,t) = \sum_{k=1}^{\infty} y_k(t) \sin\left(\frac{k\pi x}{L}\right) \quad x \in [0, L] \tag{2}$$

where $y_k(t)$ is the instantaneous amplitude of mode k.

The solution of (1) can be separated for the different modes if (2) is substituted into (1), then multiplied by the modal shape $\sin(k\pi x/L)$ and integrated over x from 0 to L. The resulting second-order differential equation governing the behavior of mode k is

$$\frac{d^2 y_k}{dt^2} + a_{1,k}\frac{dy_k}{dt} + a_{0,k} y_k = b_{0,k} F_{y,k}(t) \tag{3}$$

where

$$a_{1,k} = 2R_k \tag{4a}$$

$$a_{0,k} = \frac{T_0}{\mu}\left(\frac{k\pi}{L}\right)^2 + \frac{ES\kappa^2}{\mu}\left(\frac{k\pi}{L}\right)^4 \tag{4b}$$

$$b_{0,k} = \frac{2}{L\mu} \tag{4c}$$

$$F_{y,k}(t) = \int_{x=0}^{L} \sin\left(\frac{k\pi x}{L}\right) d_y(x,t)\,dx. \tag{4d}$$

In (3), $F_{y,k}(t)$ is the excitation force of mode k, and it is computed as the scalar product of the excitation force density and the modal shape [see (4d)].

Once the partial differential equation (1) is decoupled to a set of ordinary differential equations (3), it is possible to modify each of the parameters of (3). We take advantage of this fact by using R_k instead of R in (4a) in order to model frequency dependent losses, since now R_k can be different for the various modes having different modal frequencies.

The solution of (3) for $F_{y,k}(t) = \delta(t)$ with zero initial conditions is an exponentially decaying sine function given by

$$y_{\delta,k}(t) = A_k e^{-\frac{t}{\tau_k}} \sin(2\pi f_k t) \tag{5a}$$

$$A_k = \frac{b_{0,k}}{2\pi f_k} = \frac{1}{\pi L \mu f_k} \tag{5b}$$

$$\tau_k = \frac{2}{a_{1,k}} = \frac{1}{R_k} \tag{5c}$$

$$f_k = \frac{1}{2\pi}\sqrt{a_{0,k} - \frac{a_{1,k}^2}{4}} \approx f_0 k\sqrt{1 + Bk^2} \tag{5d}$$

where A_k is the initial amplitude, τ_k is the decay time, and f_k is the frequency of mode k. In (5d), f_0 is the fundamental frequency of the string and B is the inharmonicity coefficient. They are computed as

$$f_0 = \frac{1}{2L}\sqrt{\frac{T_0}{\mu}} = \frac{1}{2L}c_{\mathrm{t}} \quad \text{and} \quad B = \kappa^2 \frac{ES}{T_0}\left(\frac{\pi}{L}\right)^2 \tag{6}$$

where c_{t} is the transverse propagation speed.

A practical choice for the loss term R_k is the second-order function

$$R_k = \frac{1}{\tau_k} = b_1 + b_3 2\pi f_k \tag{7}$$

which implements the same type of frequency dependent losses as the finite-difference model of [20] and [21], or the usual one-pole loss filter in digital waveguide modeling (see [7, Appendix]). Example values for the b_1 and b_3 coefficients can be found in [21]. Note that (7) is just the simplest choice, since $R_k = 1/\tau_k$ can be chosen arbitrarily for the various modes, e.g., based on the analysis of real tones.

As (5) computes the impulse response of the system characterized by (3), the response to the excitation force $F_{y,k}(t)$ is obtained by the time domain convolution

$$y_k(t) = y_{\delta,k}(t) * F_{y,k}(t). \tag{8}$$

Now we can summarize the computation of the string shape $y(x,t)$ as a response to an external force.

1) Computation of the force $F_{y,k}(t)$ acting on mode k, which is the scalar product of the excitation-force density $d_y(x,t)$ and the modal shape $\sin(k\pi x/L)$, as in (4d).
2) Computation of the instantaneous amplitude $y_k(t)$ of mode k by the convolution of the mode excitation force $F_{y,k}(t)$ and the mode impulse response $y_{\delta,k}(t)$, as in (8).
3) Computation of the string shape by summing the modal shapes $\sin(k\pi x/L)$ multiplied by their instantaneous amplitudes $y_k(t)$, as in (2).

Besides the string shape, we are also interested in the force acting at the termination, because that will form the input of the soundboard modeling block. The bridge force $F_{\mathrm{b}}(t)$ at the termination $x=0$ is computed as follows:

$$F_{\mathrm{b}}(t) = T_0 \frac{\partial y}{\partial x}\bigg|_{x=0} = \frac{T_0 \pi}{L} \sum_{k=1}^{\infty} k\, y_k(t) \tag{9}$$

which is a weighted sum of instantaneous modal amplitudes $y_k(t)$.

B. Discretization

We have seen that either we are interested in the string deflection or the force at the bridge, the computation has the same structure: first projecting the external force to the input forces of the modes, then filtering with the impulse responses of the modes, and finally projecting back to the variable of interest (force or displacement). Since these projections do not depend on the time variable, they are implemented in discrete time without any modifications from the continuous-time case. The only task left is to discretize the impulse response of the modes (5).

The discretization with respect to time can be done by various methods. We have chosen the impulse invariant transform [27], because in this case the discrete-time impulse response of each mode will have a leading zero which results in a one-sample delay. Therefore, we will avoid the problem of delay free loops in excitation modeling that can arise in other discretizations, such as the bilinear transform (see Section IV for details). This useful initial zero in each modal impulse response occurs because we are modeling string displacement in response to a force excitation. With the impulse invariant transform, the discrete-time impulse response is obtained by simply sampling the continuous-time impulse response (5a), yielding

$$y_{\delta,k}[n] = y_{\delta,k}(t_n) = \frac{1}{f_{\mathrm{s}}} A_k e^{-\frac{t_n}{\tau_k}} \sin(2\pi f_k t_n) \tag{10}$$

where $t_n = nT_{\mathrm{s}}$, $T_{\mathrm{s}} = 1/f_{\mathrm{s}}$ being the sampling interval. Equation (10) differs from (5a) by a scaling factor of $1/f_{\mathrm{s}}$. This scaling is required because the discrete-time unit pulse has an area of $1/f_{\mathrm{s}}$, while the continuous-time Dirac impulse has unity area.

Taking the z transform of $y_{\delta,k}[n]$, after some algebra, gives

$$H_{\mathrm{res},k}(z) = \frac{b_k z^{-1}}{1 + a_{1,k} z^{-1} + a_{2,k} z^{-2}} \tag{11a}$$

$$b_k = \frac{A_k}{f_{\mathrm{s}}} \mathrm{Im}\{p_k\} \tag{11b}$$

$$a_{1,k} = -2\mathrm{Re}\{p_k\} \tag{11c}$$

$$a_{2,k} = |p_k|^2 \tag{11d}$$

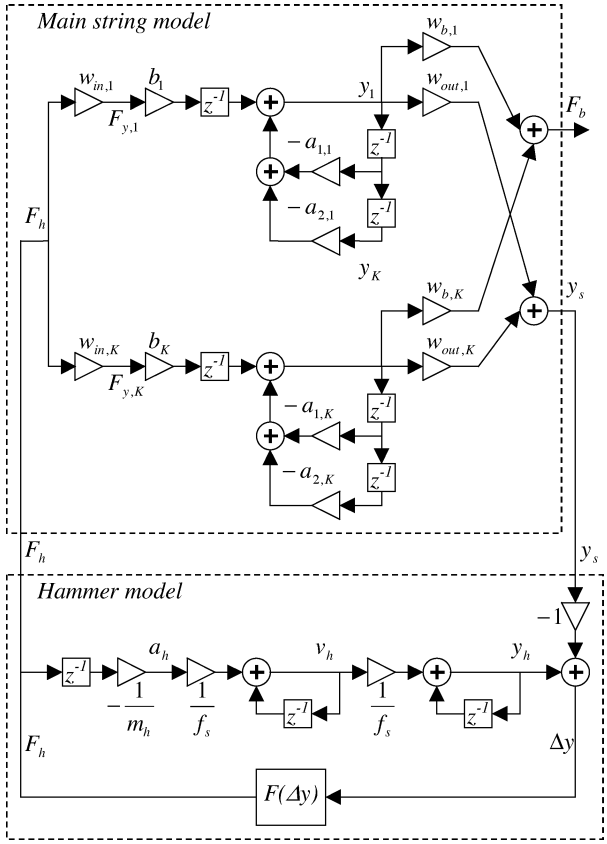

Fig. 3. String and hammer models, and their interconnection.

$$p_k = e^{j2\pi \frac{f_k}{f_{\mathrm{s}}}} e^{-\frac{1}{\tau_k f_{\mathrm{s}}}}. \tag{11e}$$

That is, each mode is implemented by a two-pole filter and a delay in series, as displayed in the top of Fig. 3 (the detailed description of the figure is left for Section IV).

IV. Hammer Modeling

Generally, two distinct approaches are used for modeling the excitation of piano strings. One commonly employed approach is to model the excitation using a signal model [11], [15], where the perceptual parameters of the resulting tone (loudness of the partials) can be directly controlled. Another common approach is physical modeling [20], [21], [6], [7], [22]. We follow the physics-based approach in our work because we believe that a fully physical excitation model is necessary to provide the required responsiveness to the player. In addition, its parameters, like hammer mass and hardness, are close to those of a real piano, leading to an intuitive control for the user.

Accordingly, the piano hammer is modeled by a small mass connected to a nonlinear spring that contacts the string at a mathematical point x_{h} [28]. The equations describing the interaction are as follows:

$$F_{\mathrm{h}}(t) = F(\Delta y) = \begin{cases} K_{\mathrm{h}} (\Delta y)^{P_{\mathrm{h}}} & \text{if } \Delta y > 0 \\ 0 & \text{if } \Delta y \le 0 \end{cases}, \tag{12a}$$

$$F_{\mathrm{h}}(t) = -m_{\mathrm{h}} \frac{d^2 y_{\mathrm{h}}(t)}{dt^2} \tag{12b}$$

where $F_{\mathrm{h}}(t)$ is the interaction force, $\Delta y = y_{\mathrm{h}}(t) - y_{\mathrm{s}}(t)$ is the compression of the hammer felt, $y_{\mathrm{h}}(t)$ is the position of the

hammer, and $y_\mathrm{s}(t)$ is the position of the string at the excitation point x_h (i.e., $y_\mathrm{s}(t) = y(x_\mathrm{h}, t)$). The hammer mass is denoted m_h, K_h is the hammer stiffness coefficient, and P_h is the stiffness exponent. Example values for hammer parameters can be found in [21].

These equations can be easily discretized with respect to time. Equation (12a) is a static nonlinearity so it is implemented as is. Equation (12b) can be converted to a discrete-time system by the impulse invariant transform. Thus, the discrete-time version of (12) is the following:

$$F_\mathrm{h}[n] = F(\Delta y) = F(y_\mathrm{h}[n] - y_\mathrm{s}[n]) \tag{13a}$$

$$y_\mathrm{h}(z) = -F_\mathrm{h}(z)\frac{1}{m_\mathrm{h}}\left(\frac{1}{f_\mathrm{s}}\frac{1}{1-z^{-1}}\right)^2 z^{-1} \tag{13b}$$

which is displayed in the bottom of Fig. 3. The extra delay term z^{-1} is inserted in (13b) to avoid the delay-free loop that would arise because there is a mutual dependence between $F_\mathrm{h}[n]$ and $y_\mathrm{h}[n]$. That is, for the calculation of one of these variables, the other should be known. Inserting the delay comes from the assumption that the hammer position changes a little during one time step $y_\mathrm{h}[n] \approx y_\mathrm{h}[n-1]$, which is generally true because $y_\mathrm{h}[n]$ is computed as the double integration of $F_\mathrm{h}[n]$.

It is a bigger problem in general that there is no delay between the input (the string position $y_\mathrm{s}[n]$) and output (the hammer force $F_\mathrm{h}[n]$) of the hammer block. This means that for computing the hammer force, the string displacement should be known, which, in return, depends on the hammer force. This again leads to a delay-free loop which is usually resolved by assuming $y_\mathrm{s}[n] \approx y_\mathrm{s}[n-1]$, implemented by using $y_\mathrm{s}[n-1]$ instead of $y_\mathrm{s}[n]$ in (13a). However, this can lead to numerical instabilities at high impact velocities [8], because the string signal can have large jumps violating the assumption $y_\mathrm{s}[n] \approx y_\mathrm{s}[n-1]$.

Nevertheless, in our special case, because of the right choice of discretization of the string model in Section III-B, the delay-free loop does not arise because the string model contains a delay element in the signal flow (see (11a) and Fig. 3). Note that in the case of different discretizations leading to a delay-free loop, special measures are needed to maintain numerical stability [29], [8].

The only parameters in Fig. 3 that have not been described so far are the input and output weights of the string model. If we assume that the hammer contacts the string at an infinitely small area at position x_h, the excitation density for the string becomes $d_y(x, t_n) = \delta(x - x_\mathrm{h})F_\mathrm{h}[n]$. Computing the scalar product of the Dirac function and the modal shape according to (4d) leads to the following input weights of the resonators:

$$w_{\mathrm{in},k} = \sin\left(\frac{k\pi x_\mathrm{h}}{L}\right). \tag{14}$$

Not so surprisingly, computing the string displacement $y_\mathrm{s} = y(x_\mathrm{h}, t_n)$ at the hammer position x_h by (2) leads to the same weights as for $w_{\mathrm{in},k}$

$$w_{\mathrm{out},k} = \sin\left(\frac{k\pi x_\mathrm{h}}{L}\right). \tag{15}$$

The third set of weights $w_{\mathrm{b},k}$ are used to compute the bridge force $F_\mathrm{b}[n]$ which is the input of the soundboard model. From (9), these weights become

$$w_{\mathrm{b},k} = \frac{k\pi T_0}{L}. \tag{16}$$

Fig. 3 shows the hammer-string model in a form that corresponds to the above derivations, but some simplifications are possible to decrease computational complexity. Because of linearity, we may lump the effects of coefficients $w_{\mathrm{in},k}$, $w_{\mathrm{out},k}$, $w_{\mathrm{b},k}$, and b_k of Fig. 3, so finally only two of them have to be implemented. It is beneficial to lump the coefficients in such a way that $w_{\mathrm{b},k} = 1$ meaning that the bridge force is computed as a simple summation. The hammer model runs during the hammer–string contact only (first few ms); therefore, the remaining two scalings have to be computed for a limited amount of time. Additionally, the z^{-1} terms coming after the b_k coefficient in Fig. 3 can be also lumped; therefore, only one delay element has to be implemented for each string model.

We note that piano hammers are not fully characterized by the model of (12), as the felt has a hysteretic behavior [28], [30], and the vibration of the hammer shank can also have some influence on the hammer force [31]. We neglect these secondary phenomena in our hammer model.

V. Longitudinal String Vibrations

In the piano string at fortissimo levels, because of the relatively large amplitude of transverse vibration, the tension is not anymore constant, resulting in a nonlinear excitation of longitudinal modes. The perceptual effect is most important in the low note range where it greatly contributes to the metallic character of the tone (the interested reader may listen to the examples at the companion page of [12]). The importance of these spectral components has been recognized long ago by piano builders [32]. In addition to the longitudinal modal frequencies, a second series of partials with lower inharmonicity coefficient has also been found in piano tones [33], which was named "phantom partials" later [34]. It has turned out that phantom partials are also generated by the longitudinal motion of the string, as a response to the tension variation coming from the transverse string motion [12]. In other words, longitudinal modal peaks and phantom partials are the free and the forced response of the same system, respectively. Therefore, a single model can describe both phenomena, which will be outlined in this section. The related theory has been developed in [12] and in [35] independently.

A very efficient way of modeling the perceptual effect of phantom partials has been presented in [36], but it has a loose connection to physical reality. In [12] and [13, Ch. 6] various, more physically meaningful techniques were presented. We propose a new variation of these physics-based techniques, which provides greater accuracy at a negligible increase in computational complexity.

A. Continuous-Time Equations

The wave equation for the longitudinal motion takes the following form [13]:

$$\mu\frac{\partial^2\xi}{\partial t^2} = ES\frac{\partial^2\xi}{\partial x^2} - 2R_\xi\mu\frac{\partial\xi}{\partial t} + \frac{1}{2}ES\frac{\partial\left(\frac{\partial y}{\partial x}\right)^2}{\partial x} \tag{17}$$

where R_ξ is the frictional resistance of the longitudinal polarization (frequency dependency will be introduced later in (21) by using different R_ξ for the various modes). Equation (17) is a linear wave equation with a nonlinear forcing term depending on the transverse slope. By comparing (17) with (1) it can be noticed that the two equations are of the same form, and they differ only in their parameters: T_0 is substituted by ES, and the dispersion term (fourth-order spatial derivative) is missing. The external force density $d_y(x,t)$ is replaced by

$$d_\xi(x,t) = \frac{1}{2}ES\frac{\partial\left(\frac{\partial y}{\partial x}\right)^2}{\partial x} \tag{18}$$

which represents the excitation from the transverse polarization.

1) Longitudinal Motion: The formal similarity to (1) means that the results of Section III can be directly used. By assuming infinitely rigid terminations, the longitudinal displacement can be written in its modal form

$$\xi(x,t) = \sum_{k=1}^{\infty}\xi_k(t)\sin\left(\frac{k\pi x}{L}\right). \tag{19}$$

Accordingly, the instantaneous amplitude $\xi_k(t)$ of the longitudinal mode k is obtained as

$$\xi_k(t) = F_{\xi,k}(t) * \xi_{\delta,k}(t) \tag{20a}$$

$$F_{\xi,k}(t) = \int_0^L d_\xi(x,t)\sin\left(\frac{k\pi x}{L}\right)dx, \tag{20b}$$

$$\xi_{\delta,k}(t) = \frac{1}{\pi L\mu f_{\xi,k}}e^{-\frac{t}{\tau_{\xi,k}}}\sin(2\pi f_{\xi,k}t) \tag{20c}$$

where $F_{\xi,k}(t)$ is the excitation force acting on the longitudinal mode k. The time-domain impulse response of longitudinal mode k is denoted by $\xi_{\delta,k}(t)$.

For small frictional resistance, the longitudinal modal frequencies $f_{\xi,k}$ and decay times $\tau_{\xi,k}$ are

$$f_{\xi,k} = kf_{\xi,0} = \frac{k}{2L}\sqrt{\frac{ES}{\mu}} \quad \text{and} \quad \tau_{\xi,k} = \frac{1}{R_{\xi,k}} \tag{21}$$

where $f_{\xi,0} = f_{\xi,1}$ is the fundamental frequency of the longitudinal vibration. In practice, the fundamental frequency of the longitudinal vibration in piano strings is around 16 to 20 times higher than that of the transverse vibration [32].

The first step in calculating the longitudinal motion is the computation of the excitation force $F_{\xi,k}(t)$ by (20b), which is the scalar product of the excitation-force density $d_\xi(x,t)$ and the longitudinal modal shape. If the transverse vibration is expressed in the modal form of (2), from (18) and (20b) it turns out that a longitudinal mode with mode number k is excited by such transverse mode pairs m and n only, for which either the sum $m+n$ or the difference $|m-n|$ of their mode numbers equal to k [12].

The two cases can be computed separately by defining $F_{\xi,k}(t)$ as a sum of two components, i.e., $F_{\xi,k}(t) = F_{\xi,k}(t)^+ + F_{\xi,k}(t)^-$. The component originating from the transverse modes that satisfy $m+n=k$ is

$$F_{\xi,k}(t)^+ = -ES\frac{\pi^3}{8L^2}k\sum_{n=1}^{k-1}n(k-n)\,y_n(t)y_{k-n}(t). \tag{22a}$$

The component coming from $|m-n|=k$ becomes

$$F_{\xi,k}(t)^- = -2ES\frac{\pi^3}{8L^2}k\sum_{n=1}^{\infty}n(k+n)\,y_n(t)y_{k+n}(t). \tag{22b}$$

That is, the various terms of the longitudinal excitation force are the products of the instantaneous amplitudes of two transverse modes, $y_m(t)$ and $y_n(t)$, meaning that the longitudinal modes will be excited at the sum and difference frequencies of transverse modes.

The force exciting the first longitudinal mode $F_{\xi,1}(t)$ is displayed in Fig. 4(a) by a solid line, computed by the discrete-time implementation of the modal model described by (22) and (20). The simulation example is a G_1 piano string. Note that the excitation force has an odd-like partial series and a lower inharmonicity compared to the spectrum of the transverse bridge force [12], which is displayed by dots to show the transverse modal frequencies as a reference. The dashed line indicates the Fourier transform of the impulse response of the first longitudinal mode $\xi_{\delta,1}(t)$, amplifying the frequencies around 690 Hz. Fig. 4(b) shows the excitation-force spectrum of the second longitudinal mode for the same example. It can be seen that here the excitation spectrum contains even partials only and that the peak of the longitudinal mode (dashed line) is located at a higher frequency (1380 Hz in this case). It is also true for all other longitudinal modes that odd modes are excited by an odd-like spectrum and even longitudinal modes by an even-like. However, the frequencies of the excitation components will be slightly different even within the odd and even modes, because of the inharmonicity of the transverse vibration [12].

The longitudinal motion is the sum of the motion of different modes. This means that spectra similar to Figs. 4(a) and (b) should be superimposed with only slightly shifted excitation frequencies and very different longitudinal modal frequencies. The result is similar to formants on a quasi-harmonic spectrum but here the peaks are somewhat smeared as they are made up of many close frequencies [12].

2) Bridge Force: If the instantaneous amplitudes $y_n(t)$ of the transverse modes are known, the longitudinal displacement can be directly computed by the use of (22), (20) and (19). However, we still need to compute the force $F_{\mathrm{b},\xi}(t)$ acting on the bridge in the longitudinal direction, since that is led to the input of the soundboard model. The bridge force $F_{\mathrm{b},\xi}(t)$ equals the tension at the termination [12]

$$F_{\mathrm{b},\xi}(t) = T_0 + ES\left.\frac{\partial\xi}{\partial x}\right|_{x=0} + \frac{1}{2}ES\left(\left.\frac{\partial y}{\partial x}\right|_{x=0}\right)^2. \tag{23}$$

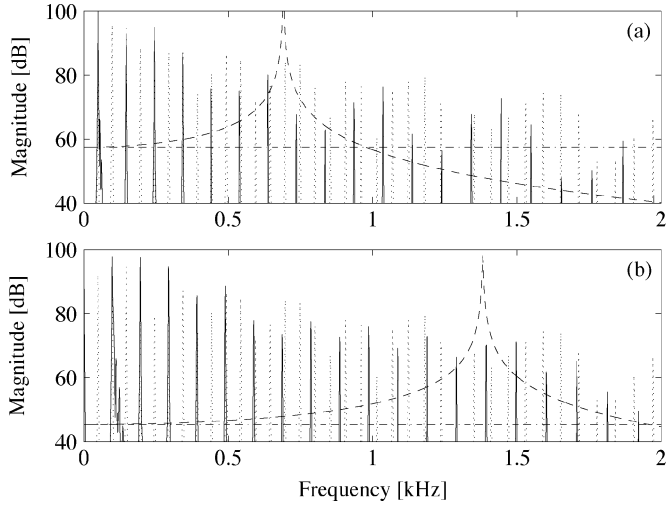

Fig. 4. Force spectrum exciting the first (a) and the second (b) longitudinal modes ($F_{\xi,1}(t)$ and $F_{\xi,2}(t)$) of a simulated G_1 piano string (solid line). The transverse bridge force (dotted line) is displayed to show the transverse modal frequencies. The dashed lines show the frequency response of the first (a) and the second (b) longitudinal modes. The horizontal dash-dotted lines are the low frequency approximations of the corresponding frequency responses. The relative levels of the signals are arbitrary.

Equation (23) shows that the force $F_{\mathrm{b},\xi}(t)$ depends not only on the longitudinal motion but on the transverse vibration as well. Due to the second-order nonlinearity, the component coming from the transverse motion has the same sum- and difference frequency terms as the component arising from the longitudinal motion, but their amplitudes and phases are different.

It can be seen in Fig. 4 (dashed lines) that the longitudinal modes have a constant gain under their resonance frequency. Therefore, let us assume for the moment that all the longitudinal modes are excited below their resonance. The transfer function of longitudinal mode k is the Laplace transform of (20c)

$$\mathcal{L}\{\xi_{\delta,k}(t)\} = \frac{2}{L\mu}\frac{1}{s^2 + \frac{2}{\tau_{\xi,k}}s + \frac{1}{\tau_{\xi,k}^2} + 4\pi^2 f_{\xi,k}^2} \tag{24}$$

from which the low frequency response $\bar{\xi}_{\delta,k}(t)$ of the resonator can be approximated as a constant gain by assuming $s \to 0$ and $1/\tau_{\xi,k} \ll f_{\xi,k}$

$$\bar{\xi}_{\delta,k}(t) = -\frac{2}{L\mu 4\pi^2 f_{\xi,k}^2}\delta(t). \tag{25}$$

The transfer function corresponding to (25) (which is a constant gain) is displayed for longitudinal mode 1 and 2 in Fig. 4. If (25) holds for all the longitudinal modes, that is, all the longitudinal modes are excited significantly below their resonance frequency, most of the transverse and longitudinal terms cancel out in (23) and only the double frequency terms remain [12], [13]. This means that the tension is spatially uniform along the string, and the bridge force becomes

$$F_{\mathrm{b},\xi}(t) = \bar{T}(t) = T_0 + \frac{\pi^2 ES}{4L^2}\sum_{n=1}^{\infty} n^2 y_n^2(t) \tag{26}$$

which is a simple squared sum of the instantaneous modal amplitudes $y_n(t)$ of the transverse polarization. Note that this corresponds to the case when the first and second time derivatives in (17) are zero (inertial and viscosity effects are negligible); thus, the dynamics of the longitudinal modes play no role [12]. As a result, the longitudinal motion simply follows the transverse one due to the fact that tension along the string tries to reach an equilibrium state. We may call this as a "static motion" of longitudinal modes.

Equation (25) holds for most of the longitudinal modes. Nevertheless, it does not hold for the lowest K modes, which are also excited around or above their resonant frequency. For these modes a correction is made by subtracting their static dc response [which is already included in the tension computed by (26)] and adding their real, frequency-dependent response

$$F_{\mathrm{b},\xi}(t) = T(x,0) = \bar{T}(t) + ES\frac{\pi}{L}\sum_{k=1}^{K} k\,\tilde{\xi}_k(t) \tag{27}$$

where $\tilde{\xi}_k(t)$ is the "dynamic response" of mode k, i.e., that portion of the motion which is in addition to the static stretching. It is computed as

$$\tilde{\xi}_k(t) = F_{\xi,k}(t) * (\xi_{\delta,k}(t) - \bar{\xi}_{\delta,k}(t)). \tag{28}$$

Note that the terms $\xi_{\delta,k}(t) - \bar{\xi}_{\delta,k}(t) = \tilde{\xi}_{\delta,k}(t)$ correspond to a resonant second-order high-pass filter, since they are the difference of a low-pass and a constant response.

Despite the quite complicated math we had to go through, we end up with a relatively simple series of steps for computing the longitudinal bridge force as a function of transverse motion.

1) Computation of the excitation force $F_{\xi,k}(t)$ acting on longitudinal mode k as pairwise products of transverse mode amplitudes $y_n(t)$ and $y_m(t)$ by (22).
2) Computation of the spatially uniform part of the tension $\bar{T}(t)$ by the squared sum of $y_n(t)$ by (26).
3) Computation of the dynamic responses $\tilde{\xi}_k(t)$ for the first K longitudinal modes according to (28).
4) Correcting the spatially uniform tension $\bar{T}(t)$ by the weighted sum of the dynamic responses $\tilde{\xi}_k(t)$ of the first K longitudinal modes by (27).

This tension correction has to be done only for those longitudinal modes, which are excited above and under resonance. That is, the resonance frequency of each longitudinal mode should be compared with the bandwidth of $F_{\xi,k}$, which is the double of that of the transverse vibration due to the second-order nonlinearity. This leads to K taking on values between 2 to 10 in practice.

B. Discretization

1) Precise Model: The above steps can be easily discretized with respect to time. Again, the only part that needs consideration is the dynamics of the longitudinal modes. The impulse response $\xi_{\delta,k}(t)$ of mode k is discretized exactly in the same way as that of the transverse modes, resulting in a filter of the form (11). Then, the dynamic response of the longitudinal modes $\tilde{\xi}_k(t)$ is calculated by subtracting the static response of the modes, which is computed by multiplying $F_{\xi,k}$ with the dc gain of the longitudinal resonators.

However, because the excitation signal of longitudinal modes is obtained by multiplying the instantaneous amplitudes of transverse modes according to (22), aliasing should be

considered. It turns out that because modal synthesis is used, aliasing can be easily avoided (in contrast to, e.g., finite-difference string models). We just have to take care not to implement any of the products in (22) and (26), where the sum frequency would go beyond the Nyquist limit $f_s/2$. This way we also throw away some difference-frequency terms from (22) that would not be aliased, but this is of no concern because these low-frequency terms would be attenuated by the high-pass filters implementing the dynamic response of the longitudinal modes anyway. Since the transverse modal frequencies are known, it is easy to decide which terms should be discarded from (22) and (26). However, it is a reasonable simplification to limit the computation of (22) and (26) for such transverse modes whose frequencies are below the half of the Nyquist rate, that is, to compute (22) and (26) up to transverse mode N for which $f_N < f_s/4$.

2) Efficient Modeling: Since typically only $K < 10$ longitudinal resonances are within the audible range, the filtering needed to compute the dynamic response of the longitudinal modes $\tilde{\xi}_k(t)$ is negligible compared to that of transverse vibration, where hundreds of modes are computed. The computational complexity lies in (22), where hundreds of pairwise products of transverse amplitudes $y_n(t)$ has to be calculated. As an example, for $K = 10$ longitudinal and $N = 100$ transverse partials, this leads to 1000 multiplications and additions, so we may seek at some more efficient implementations.

As noted earlier, odd longitudinal modes are excited by an odd-like spectrum, while even ones with an even-like, and the odd and even spectra are similar (the frequencies are only slightly shifted due to the inharmonicity). Example spectra can be found in [13 Ch. 6]. Therefore, similarly to the synthesis models of [12], we compute a single excitation force for the odd longitudinal modes and another one for the even ones. For example, all the longitudinal modes with odd mode number k are excited by the same force as computed for mode 3 ($F_{\xi,k}(t) = F_{\xi,3}(t)$ for odd k), and all the longitudinal modes with even mode number k are excited by the same force as mode 4 ($F_{\xi,k}(t) = F_{\xi,4}(t)$ for even k). Naturally, other excitation force pairs (such as $F_{\xi,1}(t)$ and $F_{\xi,2}(t)$) can also be chosen. Additionally, only the $F_{\xi,k}(t)^-$ part of (22) is computed, since the $F_{\xi,k}(t)^+$ part contains low frequency components that are out of the passband of the high-pass filter implementing the dynamic response of the longitudinal mode $\tilde{\xi}_{\delta,k}(t)$.

It is worth noting that both in (26) and in (22b) the transverse terms appearing are always of the form $ny_n(t)$, which is the instantaneous amplitude of the transverse mode multiplied by its mode number. The same multiplication is needed in (9), therefore, no additional multiplications are needed here. For computing (22b) the transverse mode outputs are simply tapped before the bridge summation point and those outputs are pairwise multiplied which has a mode number difference of k, and the products are summed. From a computational viewpoint, since the transverse modal amplitudes are stored in a vector, this means correlating the vector and its shifted version. For (26) the squared sum of the same vector is computed.

Note that in the sound synthesis model of [12] the transverse contribution (last term of (23)) was simply neglected, which has led to the appearance of some unwanted low-frequency components that should otherwise cancel out, as described in Section V-A2. This is avoided in the proposed model, since now the effect of the transverse vibration on the longitudinal bridge force is also included. This is accomplished by the additional computation of the string tension (26), and by filtering the longitudinal excitation force by the dynamic part of the longitudinal response $\tilde{\xi}_{\delta,k}(t)$ (resonant high-pass filter) instead of the full response $\xi_{\delta,k}(t)$ (resonant low-pass filter). As a result, the new model is more precise, while the computational cost is increased only slightly by the additional tension computation of (26).

VI. String Coupling

Coupling effects in transverse string vibrations occur at two different levels: first of all, even a single string vibrates both in the horizontal and vertical plane, and these vibrations are coupled through the bridge. The situation gets even more complex for the piano because for most of the keys three slightly mistuned strings are sounded together. This results in beating and two-stage decay [37], where the first is a low-frequency amplitude modulation of partial envelopes, and the second means that the partial decay is faster in the early part than in the latter.

At a second level, and with lower efficiency, coupling between strings occurs not only for the two or three different strings belonging to the same key, but also between the strings of various keys. This effect (sometimes referred to as sympathetic resonances) is most prominent when the dampers of the strings are raised by the sustain pedal, leading to the coupling of all the strings of the piano. In fact, the bridge–soundboard system connects the strings together and acts as a distributed driving-point impedance for string terminations.

Both of these two phenomena are modeled by the "Secondary resonators" in Fig. 2.

A. Beating and Two-Stage Decay

The simplest way to model beating and two-stage decay is to use two string models in parallel for a single note. Varying by the type of coupling used, many different solutions have been presented in the literature [38], [39], [10].

A computationally more efficient, perception-based approach is to model the beating and two-stage decay for those partials only where the phenomenon is audibly prominent. One way of achieving this is to implement a few (5–10) resonators in parallel, driven by the excitation signal [7], [8]. Another option is the beating equalizer [15], where the partial envelope is modulated by a peak EQ tuned to the frequency of the partial.

In this paper, we are modeling beating and two-stage decay by adding a set of secondary resonators to the string model, driven by the hammer force. This is displayed in Fig. 2. Depending on the parameterization of the resonators, they can be seen as additional string models, or, they can be used for a perception-based efficient modeling, where only a few additional resonators are implemented. We have chosen to implement them as additional string models with a full set of modes (albeit with less modes than the main string), because the same structure will be used to model the sustain pedal effect (see Section VI-B). The resonators are implemented by second-order infinite-impulse response (IIR) filters, exactly as described in Section III-B, re-

sulting in a coherent model structure. The only difference from the resonators of the main string models that the string displacement does not have to be computed; thus, coefficients $w_{\mathrm{out},k}$ in Fig. 3 are missing, and only the bridge force is computed by $w_{\mathrm{b},k}$. Accordingly, the effect of $w_{\mathrm{in},k}, b_k$, and $w_{\mathrm{b},k}$ can be lumped to a single coefficient.

B. Sympathetic Resonances

It is relatively straightforward to model sympathetic resonances in digital waveguides with bidirectional coupling between the strings, when all the strings are connected to the same termination [3], [6]. Implementing numerically stable bidirectional coupling is more complicated in modal synthesis; therefore, we are applying a simplification where the coupling of the strings is unidirectional. This leads to a "structurally stable" solution, meaning that the coupled model is stable for any choice of the coupling parameters. For this, several different variations have been presented in the literature [38], [15].

Our choice is similar to [38], where the outputs of the main string models are summed and feed to the secondary resonators of all string models, as shown in Fig. 2. This way, energy is flowing from the main string models to the secondary ones, but not vice-versa. The simplest way of implementing this is to sum the outputs of all string models, multiply them with a constant coefficient, and then distribute this signal to all secondary resonator banks. However, it is also possible to control the strength of coupling between the strings, resulting in a more realistic behavior. This can be achieved by subdividing the strings into R different regions [40], where the cumulated string outputs are $\mathbf{F}_{\mathrm{b}} = [F_{\mathrm{b},0}, \ldots, F_{\mathrm{b},R}]$ and the inputs of the secondary resonator groups are $\mathbf{F}_{\mathrm{p}} = [F_{\mathrm{p},0}, \ldots, F_{\mathrm{p},R}]$. The coupling is controlled by a $R \times R$ gain matrix $\mathbf{B}$

$$\mathbf{F}_{\mathrm{p}} = \mathbf{B}\mathbf{F}_{\mathrm{b}}. \tag{29}$$

In our implementation, we have chosen a subdivision in $R = 8$ keyboard regions, requiring an additional 64 multiplications and additions.

VII. Soundboard Modeling

As already mentioned in Section II, only the radiation effects of the soundboard are implemented here, while the impedance effects are taken into account in the string model.

The computationally most efficient way of implementing the effect of the soundboard filtering is commuted synthesis [4], [5], where the order of the model blocks (hammer–string–soundboard) is commuted: the impulse response of the soundboard excites the strings and the effect of the hammer is taken into account as a filtering operation. Thus, the soundboard response is implemented as a wavetable, resulting in a low computational complexity. However, the method assumes linearity and time-invariance; therefore, some important effects, such as the restrike of the same string or the nonlinear vibration of strings, cannot be precisely modeled.

Another efficient way of implementing the effect of the soundboard is to apply a reverberation-like algorithm. Examples include the coupled digital waveguides in [3], and feedback delay networks in [6], [7]. A difficulty of these reverberator-based approaches is that only the statistical distribution and the overall damping of the body modes can be set by the available parameter estimation techniques.

In filter-based techniques, the measured force-pressure transfer function of a real piano soundboard is used as a target specification for filter design. The most straightforward approach is the use of a finite impulse response (FIR) filter obtained by windowing of the measured impulse response. The drawback is that long filters are required: for a good tonality, 1000–2000 tap FIR filters are needed at $f_{\mathrm{s}} = 44.1$ kHz, and for reproducing the characteristic knock sound of the middle and high notes, ten thousands of taps are needed. Therefore, a direct FIR implementation is not advantageous. Either a multi-rate approach can be used [8], [13], or the FIR filter should be implemented as a fast convolution algorithm. The latter has been implemented in our real-time piano model and will be outlined in Section VII-B.

Another option is the use of a specially designed infinite impulse response (IIR) filter, where a quasi-logarithmic frequency resolution can lead to large computational savings compared to traditional FIR and IIR filter designs. The most commonly used technique is frequency warping, where the z^{-1} elements of traditional filters are replaced by first-order allpass filters [41]. Kautz filters [42] can be seen as the generalization of warped FIR filters with more flexibility in determining the frequency resolution. Fixed-pole parallel filters [43] produce the same results as the Kautz filters but lead to a simpler filter structure. Soundboard modeling with parallel filters is summarized in Section VII-A.

The impulse response of the soundboards varies as a function of bridge excitation point. This can be simulated by having different soundboard filters for the different regions of the bridge. In addition, for each region, two filter outputs are needed for stereo effect. This complexity can be reduced by implementing only one high-order soundboard filter (or a filter pair for stereo output) and additional low-order shaping filters for the different regions of the bridge, to account for the tonal differences between them [7].

A. Partitioned Convolution by FFT

A very efficient way to implement the convolution required by soundboard filtering is adopting a partitioned fast Fourier transform (FFT)-based convolution, like the *overlap-and-add* (OLA) method, which is well documented in standard digital signal processing textbooks [44].

Both the input signal (i.e., the total force at the bridge) and the soundboard impulse response are partitioned in blocks which are N samples long. In a typical software implementation, N can be taken equal to the internal buffer size of the soundcard, thus avoiding any additional latency in the computation. For each output buffer, we need to take a $2N$ point FFT of the input buffer, then multiply the result with every block of the transfer function and finally compute the result by taking a $2N$ point inverse FFT (IFFT). If we denote with M the length of the impulse response, the total cost per buffer is $\mathcal{O}(N \log N)$ for the Fourier transforms and $4(N+1)\lceil M/N \rceil = \mathcal{O}(M)$ for the multiplication in the frequency domain. Therefore, the cost per input sample is $\mathcal{O}(\log N + M/N)$, which is typically dominated by the complex multiplication when N is small, as it is

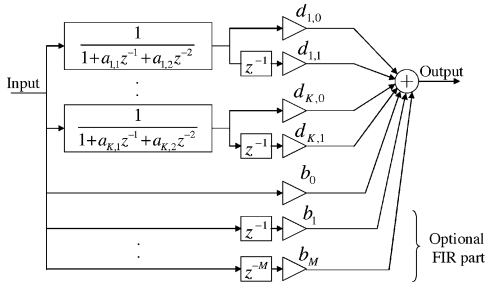

Fig. 5. Structure of the parallel second-order filter.

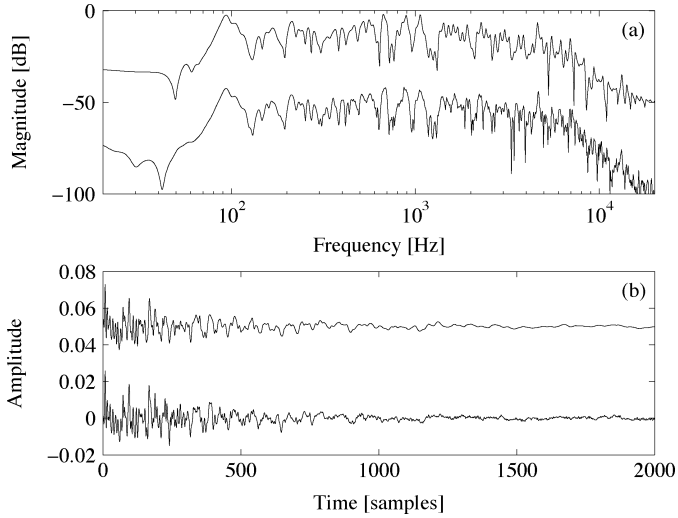

Fig. 6. (a) Frequency and (b) time-domain responses of a 200th-order parallel second-order filter design with logarithmic pole positioning. In (a) and (b), the top curve is the filter response, while the bottom is the target (the time- and frequency-domain filter responses are shifted for clarity).

the case when low-latency processing is needed. As an example, with a typical soundboard length of 20 000 taps and block size $N = 128$, the multiplication in the frequency domain requires 633 floating point operations at each time sample. This computational complexity could be decreased if necessary by combining the FFT-based convolution with multi-rate filtering, such as [8] and [13], at the expense of somewhat more complicated model structure and parameter estimation.

It is possible to avoid the tradeoff between computational load and processing delay by using a nonuniform partitioning scheme [45], [46]. With these algorithms, the principal issue is finding an optimal scheduling of the different computational tasks in order to get a constant load suitable for real-time implementations.

B. Fixed-Pole Parallel Second-Order Filters

Another option is to model the soundboard as a set of second-order parallel filters [43]. Implementing IIR filters in the form of parallel second-order sections has been used traditionally because it has good quantization noise performance and the possibility of code parallelization. The parameters of the second-order sections are usually determined from direct form IIR filters by partial fraction expansion [27]. Here the poles are set to a predetermined (e.g., logarithmic) frequency scale, leaving the zeros as free parameters for optimization. In the case of modeling a desired impulse response, the parallel filter uses the outputs of the second-order sections (exponentially decaying sinusoidal functions) as basis functions of a linear-in-parameter model. In addition to the second-order sections, it is beneficial to include a parallel FIR path for the modeling of non-minimumphase responses. Thus, the transfer function becomes

$$H(z^{-1}) = \sum_{k=1}^{K} \frac{d_{k,0} + d_{k,1}z^{-1}}{1 + a_{k,1}z^{-1} + a_{k,2}z^{-2}} + \sum_{m=0}^{M} b_m z^{-m} \tag{30}$$

where K is the number of second-order sections. The filter structure is depicted in Fig. 5

Since we are aiming at a logarithmic frequency resolution, the poles are set to a logarithmic scale

$$\vartheta_k = \frac{2\pi f_k}{f_\mathrm{s}} \tag{31}$$

$$p_k = R^{\vartheta_k/\pi} e^{\pm j\vartheta_k} \tag{32}$$

where ϑ_k are the pole frequencies in radians determined by the logarithmic frequency series f_k and the sampling frequency f_s. The pole radii form an exponentially damped sequence approximating constant Q resolution. The pole radius at $f_\mathrm{s}/2$ is set by the damping parameter R, similarly as proposed for Kautz filters in [47].

It can be seen in (30) and in Fig. 5 that since the poles determine the coefficients of the denominators $a_{k,1}$ and $a_{k,2}$, (30) becomes linear in its free parameters $d_{k,1}$, $d_{k,2}$, and b_m, which can be calculated in a closed form by standard least-squares equations from the measured soundboard response [43].

Fig. 6(a) shows the frequency response of a parallel second-order design with 100 logarithmically spaced poles with $R = 0.98$, giving a filter order of 200. The order of the FIR part M was set to zero, resulting in a canonical structure containing second-order IIR sections only. The time-domain response of Fig. 6 shows that the parallel filter can follow the long-ringing modes due to its logarithmic frequency resolution.

A benefit of using the parallel filter instead of the FFT-based convolution is that this results in a coherent model structure, a "fully modal" piano, since the structure of the soundboard model is essentially the same as that of the string model, the only difference is that now each second-order filter has a zero as well. Although the pole frequencies of the soundboard filter differ from the actual modal frequencies of the measured soundboard [the poles of the filter are set to a logarithmic scale by (31) and (32)], we may consider each second-order section as implementing one normal mode of the soundboard vibration. This analogy leads to interesting parameter modifications. The resonance frequencies of the second-order filters (pole angles) can be changed to simulate a bigger or smaller instrument body, and the decay times of the body modes can be influenced by varying the pole radii. Changing the overall magnitude response can be accomplished by scaling the feedforward coefficients $d_{k,0}$ and $d_{k,1}$ of Fig. 5, without an additional filter. These modifications can be carried out even in run-time for special effects. Multiple

outputs (e.g., for stereo effects) can be efficiently achieved by using the same set of poles for the different channels; thus, only the output coefficients of Fig. 5 have to be implemented for the channels separately [43].

According to preliminary comparisons, the computational complexity seems to be larger compared to the partitioned convolution for the same sound quality. On the other hand, the parallel filter requires significantly smaller amount of memory, has zero latency and results in a simpler code, which may be beneficial in some DSP implementations. Although it is not yet part of the current real-time piano implementation, we have good experience with offline simulations.

VIII. Software Implementation

We have implemented our model in the form of a real-time software, written in C++ using the *RtAudio* and *RtMidi* classes from the Synthesis Tool Kit (STK) [48]. The design of the software architecture was driven by the requirements of flexibility in the control and efficiency for real-time use.

Flexible control is achieved by imposing a strong modularity on the system. The different tasks are separated into three categories, depending on the rate at which they are executed at run-time. Obviously, the arithmetic-intensive operations required for the synthesis are performed for each audio sample, i.e., at the *audio rate* (sampling rate f_s). Polyphony management, hammer initialization and all the other operations that depend directly on player interaction are performed at *MIDI rate* $= f_s/N_a$, where N_a is the audio buffer size. Finally, run-time modifications of the synthesis parameters are handled at a *calibration rate*, which is the slowest one, usually between 20 and 50 Hz. Calibration parameters are received from an external program running a graphical user interface, or alternatively with MIDI control change messages. The processing of these parameters may be computationally expensive, but it is performed asynchronously at a low rate, so real-time performance is not affected.

In order to plan efficient optimizations on the structure, it is important to estimate the computational cost of each note, which can vary significantly from the bass to the treble range. The cost of each synthesis block is summarized in Table I in terms of the number of the primary resonators N, secondary resonators N_2, longitudinal resonators N_l and the number N_{exc} of transverse modes used for computing the excitation force for the longitudinal vibration. Fig. 7 shows typical values for the number of resonators in a practical calibration of the model. The number of resonators is derived by imposing an upper limit on their frequency and an additional absolute limit on the number of resonators.

Basically, we need to perform three MPOS (multiplications per output sample) per resonating filter if there is a nonzero input signal, and only two for the free evolution of the filter. Moreover, if the hammer is exciting the string, we also need to consider the cost for updating the string position at contact point (N MPOS), plus the cost of updating (13). Finally, longitudinal excitation force computation requires three MPOS for every transverse partial involved, one for the spatially constant tension term of (26), and two for the products of (22b) since the excitation force is computed for one odd and one even longitudinal mode. The estimated relative overall cost when all the notes are sounding is indicated in the third column of Table I. The computation assumes that the hammer is active only for a limited amount of time, as in normal playing conditions. Note that the number of multiplications do not precisely represent the resulting computational load in modern processors, since that will highly depend on the efficiency of memory access (cache), pipeline stalls, code and data parallelization, etc. However, they can still give an idea about the relative computational load of the various parts of the model.

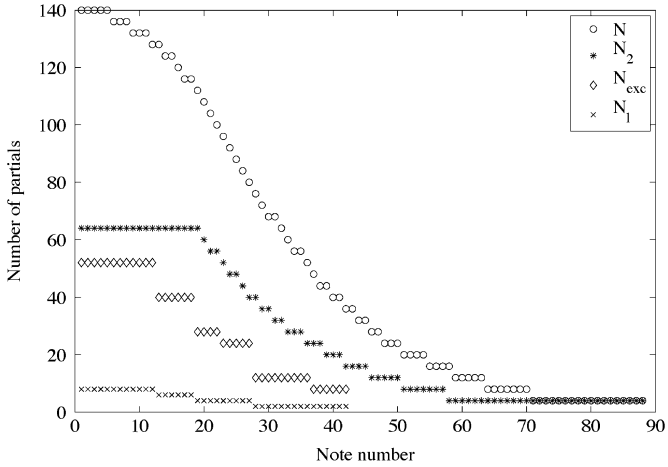

Fig. 7. Number of resonators for each note. N refers to primary resonators, N_2 to the secondary ones, N_l to the longitudinal modes and N_{exc} is the number of transversal partials used for longitudinal excitation force computation.

TABLE I
Computational Cost of String and Hammer Modeling

Synthesis block	MPOS for a single note	Overall cost
Transversal vibration	$2 \times N$ (free string)	40%
Hammer update	$2 \times N + 3$ (+static nonlinearity)	5%
Secondary resonators	$3 \times N_2$	30%
Longitudinal vibration	$3 \times N_{\text{exc}} + 3 \times N_l$	25%

One nice feature of the modal based approach is that parallelization is very straightforward for most of the components. In our particular implementation, we have taken advantage of the parallel capabilities of current x86 PCs, but similar approaches are applicable to most of shared-memory parallel architectures, which also include custom DSP boards. The parallel bank of filters of Fig. 3 can be easily implemented in a single instruction multiple data (SIMD) fashion. We used Intel's streaming SIMD extension (SSE) instructions [49], which can produce a speed-up factor of almost four times with single precision floating point data.

Soundboard filtering is implemented by a fast convolution engine with uniform partition size, using the efficient Fastest Fourier Transform in the West (FFTW) library [50]. By looking at the general diagram of Fig. 2, it is easy to notice that the soundboard radiation filter can be run in parallel with the rest of the model. In our implementation, we exploit the multicore capabilities of current processors by performing soundboard filtering in a separate thread, which communicates with the main synthesis thread through a ring buffer. The drawback

is that some additional latency is introduced, which can be avoided in hardware with finer synchronization capabilities of DSP boards. The computational load of soundboard filtering varies depending on the number of convolutions, the length of each filter and the implementation used. In our case, running four 20 000 tap convolutions requires around one fourth of the resources used by string and hammer modeling at full polyphony.

Our software is able to run at full polyphony, with a total number of 10 000 second-order resonators and four 20 000 tap convolutions, at an approximate load of 30% on a Intel's *Core 2 Duo*@2.4-GHz laptop. The computational cost can be further reduced by applying some sort of polyphony management. By knowing the approximate cost of each note, we have implemented a polyphony queue where the limit is imposed on the total computational cost and not on the number of the notes. It is possible to further extend this approach in many ways, for example by selectively deactivating the partials of a single note when they are not audible anymore. The only limit coming from the architecture of the model is that the secondary bank of resonators cannot be deactivated, since they are also used for simulating sympathetic resonances.

The interested reader may listen to the sound examples at http://www.mit.bme.hu/~bank/publist/taslp-piano.

IX. Conclusion and Future Research

This paper has presented a real-time piano model based on recent developments in string modeling. The main string model is based on the discretization of the modal solution of the wave equation, and it is connected to a nonlinear hammer model. The problem of delay-free loops is avoided by the appropriate choice of discretization. The effects of beating and two-stage decay are modeled by a secondary bank of resonators, driven by the excitation force of the main string model. Sympathetic resonances are synthesized by redistributing some parts of the outputs of the main string models to the secondary resonators. The bridge force arising due to nonlinearly excited longitudinal vibrations is computed by first assuming spatially uniform tension, then correcting this with the dynamics of longitudinal modes. The proposed method provides greater accuracy compared to earlier techniques, while does not increase the computational complexity significantly. For soundboard modeling, the FFT-based partitioned convolution and a filter structure similar to that of the string model was considered. The computational cost of the string parts was estimated based on the number of resonators, and the details of the software implementations were also given.

In the current piano model, dampers are implemented by simply decreasing the decay times of the resonators. For future work, a physics-based damper model would increase the realism of the note off sounds significantly and would allow the simulation of part pedaling. The soundboard models based on fast convolution and parallel second-order filters should be evaluated from a perceptual point of view. That is, the computational complexity of the two methods that is required for reaching the same subjective quality should be determined by listening tests. Finally, the combination of the two approaches could possibly lead to an even higher efficiency.

In our piano model, all the parts are real physical models except the soundboard, which is implemented as a digital filter and parameterized from measured transfer functions. In the far future, it might be possible to incorporate a physics-based soundboard model based on finite-difference or finite-element modeling. Alternatively, the soundboard impulse response could be computed at calibration rate by finite-difference or finite-element modeling, and implemented as a filtering algorithm, allowing the modification of the geometrical and material properties of the soundboard in run time.

Acknowledgment

The authors would like to thank Dr. László Sujbert, Zsolt Garamvölgyi, the Associate Editor Prof. Julius Smith, and the anonymous reviewers for their helpful comments.

References

[1] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part 1," *J. Audio Eng. Soc.*, vol. 19, no. 6, pp. 462–470, Jun. 1971.

[2] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part 2," *J. Audio Eng. Soc.*, vol. 19, no. 7, pp. 542–550, 1971.

[3] G. E. Garnett, "Modeling piano sound using digital waveguide filtering techniques," in *Proc. Int. Comput. Music Conf.*, Urbana, IL, 1987, pp. 89–95.

[4] J. O. Smith and S. A. Van Duyne, "Commuted piano synthesis," in *Proc. Int. Comput. Music Conf.*, Banff, AB, Canada, Sep. 1995, pp. 335–342 [Online]. Available: http://ccrma.stanford.edu/~jos/cs.html

[5] S. A. Van Duyne and J. O. Smith, "Developments for the commuted piano," in *Proc. Int. Comput. Music Conf.*, Banff, AB, Canada, Sep. 1995, pp. 319–326 [Online]. Available: http://ccrma.stanford.edu/~jos/cs.html

[6] G. Borin, D. Rocchesso, and F. Scalcon, "A physical piano model for music performance," in *Proc. Int. Comput. Music Conf.*, Thessaloniki, Greece, Sep. 1997, pp. 350–353.

[7] B. Bank, "Physics-based sound synthesis of the piano" M.S. thesis, Budapest Univ. of Technol. and Economics, Budapest, Hungary, May 2000 [Online]. Available: http://www.mit.bme.hu/~bank/thesis, published as Rep. 54 of HUT Lab. of Acoust. and Audio Signal Process.

[8] B. Bank, F. Avanzini, G. Borin, G. De Poli, F. Fontana, and D. Rocchesso, "Physically informed signal-processing methods for piano sound synthesis: A research overview," *EURASIP J. Appl. Signal Process.*, vol. 2003, no. 10, pp. 941–952, Sep. 2003.

[9] J. Bensa, "Analysis and synthesis of piano sounds using physical and signal models" Ph.D. dissertation, Universitá de la Máditárranáe, Marseille, France, May 2003 [Online]. Available: http://www.lma.cnrs-mrs.fr/~bensa

[10] M. Aramaki, J. Bensa, L. Daudet, P. Guillemain, and R. Kronland-Martinet, "Resynthesis of coupled piano string vibrations based on physical modeling," *J. New Music Res.*, vol. 30, no. 3, pp. 213–226, 2002.

[11] J. Bensa, K. Jensen, and R. Kronland-Martinet, "A hybrid resynthesis model for hammer-string interaction of piano tones," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 1021–1035, Jan. 2004.

[12] B. Bank and L. Sujbert, "Generation of longitudinal vibrations in piano strings: From physics to sound synthesis," *J. Acoust. Soc. Amer.* vol. 117, no. 4, pp. 2268–2278, Apr. 2005 [Online]. Available: http://www.mit.bme.hu/~bank/jasa-longitud

[13] B. Bank, "Physics-based sound synthesis of string instruments including geometric nonlinearities" Ph.D. dissertation, Budapest Univ. of Technol. and Economics, Budapest, Hungary, Feb. 2006 [Online]. Available: http://www.mit.bme.hu/~bank/phd

[14] J. Rauhala, H. M. Lehtonen, and V. Välimäki, "Toward next-generation digital keyboard instruments," *IEEE Signal Process. Mag.*, vol. 24, no. 2, pp. 12–20, Mar. 2007.

[15] J. Rauhala, M. Laurson, V. Välimäki, H.-M. Lehtonen, and V. Norilo, "A parametric piano synthesizer," *Comput. Music J.*, vol. 32, no. 4, pp. 17–30, Winter 2008.

[16] S. A. S. Modartt, Pianoteq True Modelling. Ramonville, France, 2006 [Online]. Available: http://www.pianoteq.com/

[17] Roland V-Piano 2009 [Online]. Available: http://www.roland.com/products/en/V-Piano/, Roland Corp.

[18] P. M. Morse, *Vibration and Sound*, reprint, 1 ed. New York: McGraw-Hill, 1948, 1936.
[19] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, "Discrete-time modelling of musical instruments," *Rep. Progress in Phys.*, vol. 69, no. 1, pp. 1–78, Oct. 2006.
[20] A. Chaigne and A. Askenfelt, "Numerical simulations of piano strings. I. A physical model for a struck string using finite difference methods," *J. Acoust. Soc. Amer.*, vol. 95, no. 2, pp. 1112–1118, Feb. 1994.
[21] A. Chaigne and A. Askenfelt, "Numerical simulations of piano strings. II. Comparisons with measurements and systematic exploration of some hammer-string parameters," *J. Acoust. Soc. Amer.*, vol. 95, no. 3, pp. 1631–1640, Mar. 1994.
[22] N. Giordano and M. Jiang, "Physical modeling of the piano," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 7, pp. 926–933, Jun. 2004.
[23] J. O. Smith, "Techniques for digital filter design and system identification with application to the violin," Ph.D. dissertation, Stanford Univ., Stanford, CA, June 1983.
[24] J. O. Smith, "Physical audio signal processing for virtual musical instruments and audio effects," in *USA: Center for Computer Research in Music and Acoustics*. Stanford, CA: Stanford University, W3K Publishing, Jul., Aug. 2009 [Online]. Available: http://ccrma.stanford.edu/~jos/pasp
[25] J.-M. Adrien, "The missing link: Modal synthesis," in *Representations of Musical Signals*, G. de Poli, A. Piccialli, and C. Roads, Eds. Cambridge, MA: MIT Press, 1991, pp. 269–297.
[26] L. Trautmann and R. Rabenstein, *Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method*. New York: Kluwer Academic/Plenum, 2003.
[27] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.
[28] X. Boutillon, "Model for piano hammers: Experimental determination and digital simulation," *J. Acoust. Soc. Amer.*, vol. 83, no. 2, pp. 746–754, Feb. 1988.
[29] G. Borin, G. De Poli, and D. Rocchesso, "Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 8, no. 5, pp. 597–606, Sep. 2000.
[30] A. Stulov, "Hysteretic model of the grand piano felt," *J. Acoust. Soc. Amer.*, vol. 97, no. 4, pp. 2577–2585, Apr. 1995.
[31] N. Giordano and J. P. Winans II, "Piano hammers and their compression characteristics: Does a power law make sense?," *J. Acoust. Soc. Amer.*, vol. 107, no. 4, pp. 2248–2255, Apr. 2000.
[32] H. A. Conklin, "Design and tone in the mechanoacoustic piano. Part III. Piano strings and scale design," *J. Acoust. Soc. Amer.*, vol. 100, no. 3, pp. 1286–1298, Jun. 1996.
[33] I. Nakamura and D. Naganuma, "Characteristics of piano sound spectra," in *Proc. Stockholm Music Acoust. Conf.*, 1993, pp. 325–330.
[34] H. A. Conklin, "Generation of partials due to nonlinear mixing in a stringed instrument," *J. Acoust. Soc. Amer.*, vol. 105, no. 1, pp. 536–545, Jan. 1999.
[35] A. Watzky, "On the generation of axial modes in the nonlinear vibrations of strings," in *Proc. Acoustics'08 Paris Conf.*, Paris, France, Jun. 2008.
[36] J. Bensa and L. Daudet, "Efficient modeling of "phantom" partials in piano tones," in *Proc. Int. Symp. Musical Acoust.*, Nara, Japan, Mar. 2004, pp. 207–210.
[37] G. Weinreich, "Coupled piano strings," *J. Acoust. Soc. Amer.*, vol. 62, no. 6, pp. 1474–1484, Dec. 1977.
[38] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: From Karplus–Strong algorithm to digital waveguides and beyond," *Comput. Music J.*, vol. 22, no. 3, pp. 17–32, 1998.
[39] J. O. Smith, "Efficient synthesis of stringed musical instruments," in *Proc. Int. Comput. Music Conf.*, Tokyo, Japan, Sep. 1993, pp. 64–71 [Online]. Available: http://ccrma.stanford.edu/~jos/cs/cs.html
[40] S. Zambon, H.-M. Lehtonen, and B. Bank, "Simulation of piano sustain-pedal effect by parallel second-order filters," in *Proc. Conf. on Digital Audio Effects*, Espoo, Finland, Sep. 2008, pp. 199–204.
[41] M. Karjalainen and J. O. Smith, "Body modeling techniques for string instrument synthesis," in *Proc. Int. Comput. Music Conf.*, Hong Kong, Aug. 1996, pp. 232–239.
[42] T. Paatero and M. Karjalainen, "Kautz filters and generalized frequency resolution: Theory and audio applications," *J. Audio Eng. Soc.*, vol. 51, no. 1–2, pp. 27–44, Jan./Feb. 2003.
[43] B. Bank, "Direct design of parallel second-order filters for instrument body modeling," in *Proc. Int. Computer Music Conf.*, Copenhagen, Denmark, Aug. 2007, pp. 458–465 [Online]. Available: http://www.acoustics.hut.fi/go/icmc07-parfilt
[44] V. Madisetti and D. Williams, *The Digital Signal Processing Handbook*. Boca Raton: CRC, 1998.
[45] W. G. , "Efficient convolution without input—output delay," *J. Audio Eng. Soc.*, vol. 43, no. 3, pp. 127–136, Nov. 1995.
[46] G. Garcia, "Optimal filter partition for efficient convolution with short input output delay," in *Proc. 113th AES Conv.*, Los Angeles, 2002, Preprint No. 5660.
[47] M. Karjalainen and T. Paatero, "Equalization of loudspeaker and room responses using Kautz filters: Direct least squares design," *EURASIP J. Adv. Sign. Process., Spec. Iss. on Spatial Sound and Virtual Acoustics*, vol. 2007, pp. 13–, 2007, article ID 60949, doi:10.1155/2007/60949.
[48] G. Scavone, "RtAudio: A cross-platform C++ class for realtime audio input/output," in *Proc. Int. Comput. Music Conf.*, Gothenburg, Sweden, Sep. 2002, pp. 196–199.
[49] R. Gerber, A. Bik, K. Smith, and X. Tian, *The Software Optimization Cookbook*. Santa Clara, CA: Intel, 2002.
[50] M. Frigo and S. Johnson, "FFTW: An adaptive software architecture for the FFT," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Seattle, WA, May 1998, vol. III, pp. 1381–1384.

Balázs Bank received the M.S. and Ph.D. degrees in electrical engineering from the Budapest University of Technology and Economics, Budapest, Hungary, in 2000, and in 2006, respectively.

During his M.S. thesis work, he was with the Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, Espoo, Finland. From 2000 to 2006, he was a Ph.D. student and Research Assistant at the Department of Measurement and Information Systems, Budapest University of Technology and Economics. In 2001, he visited the Department of Information Engineering, University of Padova, Padova, Italy. In 2007, he returned to the Acoustics Laboratory of the Helsinki University of Technology for a year, by the support of an FP6 Marie Curie EIF individual fellowship. In 2008, he was with the Department of Computer Science, Verona University, Verona, Italy, working on the piano model described in this paper. Currently, he is a Postdoctoral Researcher at the Budapest University of Technology and Economics, supported by the Norway and EEA Grants and the Zoltán Magyary Higher Education Foundation. His research interests include physics-based sound synthesis and filter design for audio applications.

Stefano Zambon graduated in computer science at the University of Verona, Verona, Italy, in 2007, where he is currently pursuing the Ph.D. degree.

He worked as Core Developer for the piano model that forms the topic of this paper, writing most of the C++ code for the real-time synthesizer. His main research interest is in physics-based sound synthesis for musical instruments and virtual environments.

Federico Fontana (M'03) received the Laurea degree in electronic engineering from the University of Padova, Padova, Italy, in 1996 and the Ph.D. degree in computer science from the University of Verona, Verona, Italy, in 2003.

During the Ph.D. degree, he was a Research Consultant in the design and realization of real-time audio DSP systems. He is currently an Assistant Professor in the Department of Computer Science teaching sound processing and nonvisual interaction at the University of Verona. In 2001, he was Visiting Scholar at the Laboratory of Acoustics and Audio Signal Processing, Helsinki University of Technology, Espoo, Finland. He was team member in several national and international research projects. He was the Coordinator of the Joint Project that resulted in the piano model covered in this paper. He coordinates the EU project 222107 NIW under the FP7 ICT-2007.8.0 FET-Open call. His current interests are in sonic interaction design, sound and music computing, and computational modeling of nonlinear systems.

Dr. Fontana is a Guest Editor of this special issue of the Transactions.