Signal- and Physics-Based Sound Synthesis of Musical Instruments

Balázs Bank[†], János Márkus[†], Attila Nagy[‡] and László Sujbert[†]
[†]Dept. of Measurement and Information Systems
[‡]Dept. of Telecommunications
Budapest University of Technology and Economics
H-1521 Budapest, Hungary
Email: bank@mit.bme.hu, markus@mit.bme.hu,
nagyab@hit.bme.hu, sujbert@mit.bme.hu

Abstract

In this paper signal-based and physics-based sound synthesis methods are described, with a particular emphasis on our own results achieved in the recent years. The applications of these methods are given in the case of organ, piano, and violin synthesis. The two techniques are compared based on these case studies, showing that in some cases the physics-based, in other cases the signal-based realization is more advantageous. As a theoretical result, we show that the two methods can be equivalent under special circumstances.

Keywords

digital signal processing, sound synthesis, musical acoustics, signal modeling, physical modeling, organ, piano, violin

1 Introduction

Musicians and music students – especially those playing organ, piano or other large instruments – would require of having small size, economic and light musical instruments for portable, stage or home applications. Composers would like to try all kind of instruments they otherwise do not play for searching new expressions. Thus, traditional instrument models are required to satisfy these requirements. Naturally, the sound quality of these artificial instruments needs to be comparable to that of the original ones. By modeling traditional instruments (like guitar, piano, organ, strings, winds, brass, etc.) and modifying the model parameters, novel, never-heard sounds can be generated. In addition, with more insight and better description of the physical operation of these instruments, new and efficient algorithms can be developed from which other fields of digital signal processing can benefit.

Sound synthesis methods can be classified in many ways. Here we divide them into three groups, by unifying two groups of the classifications found in [1, 2].

The first group is the family of abstract methods. These are different algorithms which can easily generate synthetic sounds. Methods like frequency modulation [3] and waveshaping [4, 5] belong to this category. Modeling real instruments with these methods is fairly complicated as the relationship between the parameters of the technique and those of the real instruments cannot be easily formulated. Thus, these methods are beyond the scope of this paper.

The second group (*signal modeling*) is the one which models the *sound* of the musical instruments. In this case, the input to the model is only the waveform or a set of waveforms generated by the instrument and the physics of the sound generation mechanism is not examined in details. Synthesis methods like PCM (Pulse Code Modulation) [6] and SMS (Spectral Modeling Synthesis) [7] belong to this category. The corresponding groups in the taxonomy of [1, 2] are processing of pre-recorded samples and spectral models.

The third group (*physical modeling*) is the one which instead of reproducing a specific sound of an instrument, tries to model the instrument physical behavior itself. Usually, the physical system (such as a string on an instrument or the skin of a drum) can be described with a set of difference equations and transfer functions. Given the excitation of the instrument (such as bowing the string or hit the drum), the difference equations can be solved (or the general solution can be applied for the given input), and the output of the model is expected to be close to the output of the real instrument. One well-known method in this category is the waveguide synthesis [8] which efficiently models the vibration of a one-dimensional string, based on the solution of the wave-equation.

In this paper, the signal- and physical-model based synthesis methods are examined, based on our own results achieved in the last years. In Sec. 2 an efficient signal model based synthesis method is introduced and applied for modeling the sound of organ pipes. Then Sec. 3 describes an extended digital-waveguide based physical model with the application to modeling the sound of the piano and the violin. Finally, in Sec. 4, the equivalence of the two methods for a given excitation is proven, and detailed comparison is given from the viewpoint of efficiency and applicability. The results are summarized in Sec. 5.

2 Signal Modeling

Nowadays, the most commonly used signal model-based synthesis method is the Pulse Code Modulation (PCM). The method simply samples the sound of the instrument to be modeled and stores the samples in a digital memory. To reduce the required memory for a waveform, usually the quasy-steady state of the sound (after the transient) is stored as one period, and this period is repeated continuously at playback. To be even more effective, usually not all possible sound is sampled (e.g. all the 88 keys of a piano), but only a few, and the missing waveforms are generated by resampling the stored ones.

It can be readily deducted from the above discussion that the PCM synthesis technique has some limitations. One limitation is the lack of controllability. As the method simply plays back the wavetables, the musician does not have the right tool to modify the characteristics of the sound. Other limitation is the absence of random effects. Most instrument (especially bowed string instruments and wind instruments) produce different transients at the start of the sound and random effects exist also in the stationary state (typical the windnoise for wind instruments).

Thus, a signal model has to take into account all these effects. In the following, first the organ and its characteristics from a signal modeling viewpoint is described. Then a conceptual signal model and its application to the sound synthesis of the organ pipe is introduced which is flexible enough to model all the required parameters.

2.1 The Sound Characteristic of the Organ

The pipe-organ is one of the largest musical instruments. A small, efficient and high fidelity instrument substituting the church organ is long awaited by the organ players and students. Accordingly the organ is among the most intensively studied instruments.

The sound generators of the organ are the flue and the reed pipes. As in a real organ flue pipes are dominant (small organ does not even have reed pipes), only the most important properties of the sound generated by the flue pipes are examined in the following.

It is well known that a significant and also easy to measure part of a musical instrument's sound is the stationary spectrum. Accordingly, the organ stops have also different characters, and the spectrum is strongly depending on the pipes' physical parameters [9].

However, only the synthesis of the main components of the spectrum is not enough for a good-quality reproduction. The attack and decay transients and the modulations on the harmonics, or other quasi-steady processes are important part of a musical sound, too. Some examinations prove that without the attack and decay transients some instrument cannot be identified [10], and in some cases only the specific modulations of an instrument on a sine wave are enough to recognize the instrument itself [11]. Hence, a good synthesis has to implement both the transient and the quasi-steady processes.

Another important property of some musical instrument is the effect of the ambience of the sound-source. The organ normally sounds in a church or in a hall, far away from the listeners. Closer to the pipes (or without this effect) the organpipe-sound sound unfamiliar without this reverberation [12]. Another external effect is the sometimes observable coupling mechanism of two pipes [13]. The localization of the sound-sources (which originates from the different positions of the pipes) falls also under this category [14].

2.2 Model Structure

The main concept of the proposed synthesis method is the periodic signal model that has been already applied in several other technical applications [15]. This model – a conceptual signal generator – is based on the Fourier-expansion of the periodic signals. According to the sampling-theorem, such a generator can generate a band-limited periodic signal, consisting of N complex components. In sound synthesis it realizes directly the discrete spectrum components of the instrument according to the previous discussion, and it is usually referred to as additive synthesis [6].

In this concept the attack and decay transients have effect mainly on the partials. During the transients, the adjustable parameters of each partial can be the magnitude, the frequency and the relative phase. In this paper only the amplitude-variation is examined, as this is the most relevant.

The organ pipes, as the most of the other wind instruments have a special characteristic, the so-called wind-noise. In some stops, this is the component which characterize the pipe, thus it needs to be modeled. The noise is a wideband component of the sound, with a typical spectral shape (see in Fig. 3). To integrate it into the signal model, the periodic generator has to be completed with a special noise-generator. Naturally, during the transients the envelope of the noise has to be changed as well.

The applied periodic signal model for sound synthesis is displayed in Fig. 1. The periodic signal generator has two main parameters – the fun-

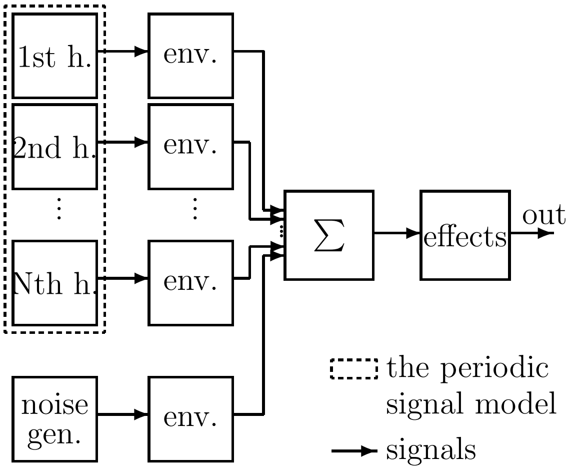

Figure 1: The integrated signal model

damental frequency and the volume – and each harmonic component has further parameters, the relative magnitude and the phase. The noise generator produces filtered white noise which is added to the magnitude-modified outputs of the periodic generator. At the end the summarized output is modified by the external effects discussed above.

2.3 Parameter Estimation

In order to determine the correct parameters of the signal model, original pipe-sounds were recorded at a sampling rate of 44,100 Hz, with a resolution of 16 bit. The records were processed off-line with MatLab, using a developed analysis process that can be seen in Fig. 2.

First, by defining magnitude-limits the stationary and the transient parts (the attack and the decay) were separated in the time-domain. From the stationary part the fundamental frequency and the magnitudes of the harmonic components were calculated via the discrete Fourier transform (DFT).

A novelty of the introduced method (first proposed in [16]) is that for data and computation-time reduction the attack and decay envelopes of the harmonics are implemented as step-responses of IIR-filters. Using this method, the ith harmonics at time step k can be computed as

$$y_{i,k} = h_{i,k} A_i \cos(2\pi(if_0/f_s)k + \varphi_i), \ (i = 1..N) \tag{1}$$

where $y_{i,k}$ is the harmonic component, A_i and φ_i are the relative magnitude and phase of the component, f_0 and f_s are the fundamental and the sampling frequency, respectively, and $h_{i,k}$ represents the samples of the step-response of the designed envelope-filter.

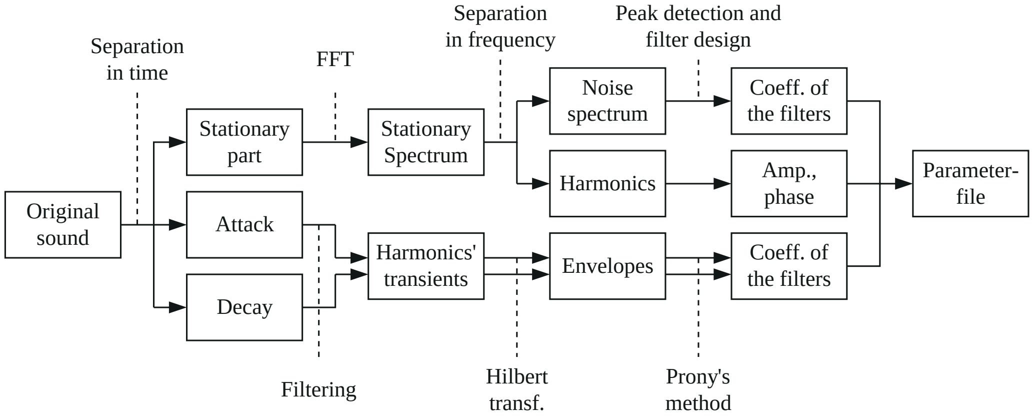

Figure 2: The analysis process

The parameters of these envelope-filters were determined in the time-domain. First, each harmonic component was filtered by a narrowband FIR filter designed by means of the windowing design method [17]. The trade-off between selectivity and filter length had to be balanced well in order not to suppress the transient of the harmonics by the transient of the filter.

Then the envelopes of the harmonics were determined as the absolute value of their analytical signal which is a complex signal originated from the original signals and their Hilbert-transform. To get the best time-domain result, the obtained envelopes were averaged, and a step-response of a 2nd or 3rd order IIR filter was fitted on each of them. The algorithm used the Prony's IIR-filter design method as initial step, then for better curve-fitting the Steiglitz-McBride iteration was used [18].

As mentioned previously, the spectrum of the organ pipe has also noise component. The noise-filter was designed as follows: subtracting the discrete components from the spectrum, 2nd order resonant filters were fitted to the specific peaks in the averaged noise spectrum. They can be designed easily having the center frequency, the gain level and the damping factor of the separated peaks. The resulted analog filter consists of 6-10 2nd order resonators, and the filter was converted to digital one by means of the bilinear transform [18].

The examined external effects were only the reverberation of the hall and the location of the pipes. This latter one can be modeled by intensity and time-delay stereo soundfield, while the reverberation can be simulated using hall-simulators.

2.4 Synthesis Results

The spectrum of two organ pipes and their models can be seen in Fig. 3. The first one is a c_4-pipe of a Bourdon register (closed, wood pipe), the second is a Diapason e_4-pipe, which is an opened organ-metal pipe. It can be seen clearly that both the original and model Bourdon pipe have more noise, and their odd harmonics have smaller magnitude, than those of the Diapason pipes. Furthermore, the metal pipe and its model have much more relevant components than the wood ones'. The modeling of the discrete spectrum is very good, and the synthesis of the main characteristic of the noise spectrum is also acceptable.

An example of the averaged original attack transients and the estimated 3rd order IIR filter step-responses can be seen in Fig. 4. The estimation is good for the lower harmonics with good signal to noise ratio (SNR) (see Fig. 3, Diapason pipe). The higher the order of the component, the smaller its SNR, this is why the modeling worse for higher order components. Note that their precise synthesis is not required accordingly to their small magnitude.

To examine the efficiency of the introduced synthesis method, it had been implemented off-line using MatLab, and real-time on a 16 bit fixed-point digital signal processor (DSP). Using these programs, some demonstrations have been recorded. For comparison, original records are also available using the measured organs. These original and synthesized samples are available through the Internet, at [19].

3 Physical Modeling

3.1 Model Structure

Since the physical modeling approach simulates the structure of the instrument, the parts of the model correspond to the parts of real instruments. In every string instrument, the heart of the sound production mechanism is the string itself. The string is excited by the excitation mechanism, which corresponds to the hammer strike in the case of the piano, or to the bow in the case of the violin. The string is responsible for the generation of the periodic sound by storing this vibration energy in its normal modes. One part of this energy dissipates and an other part is radiated to the air by the instrument body. The body can be seen as an impedance transformer between the string and the air, which increases the effectiveness of radiation significantly. The body provides a terminating impedance to the string, therefore it also influences the modal parameters of string vibration, i.e., partial frequencies, amplitudes, and decay times.

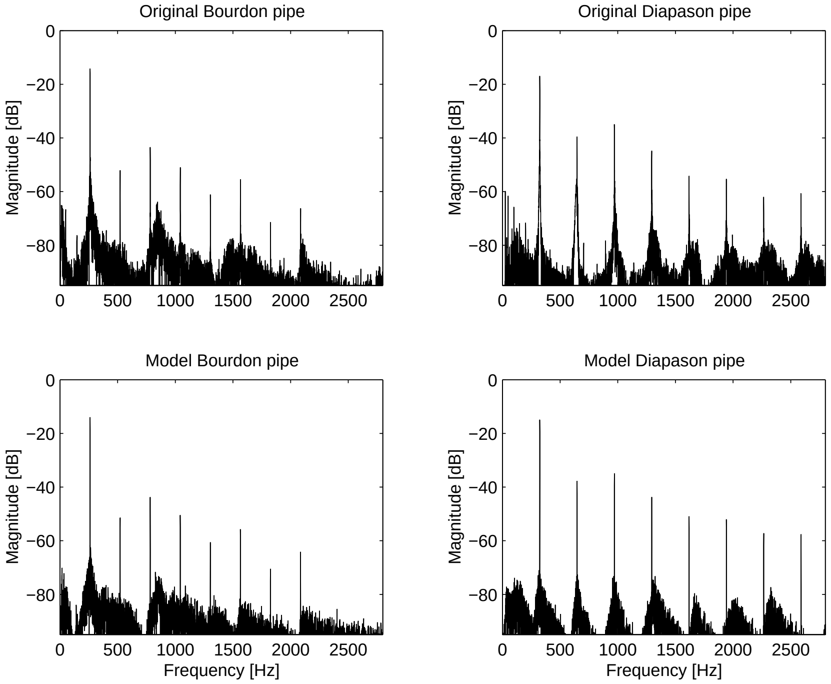

Figure 3: The stationary spectrum of two original pipes and their models

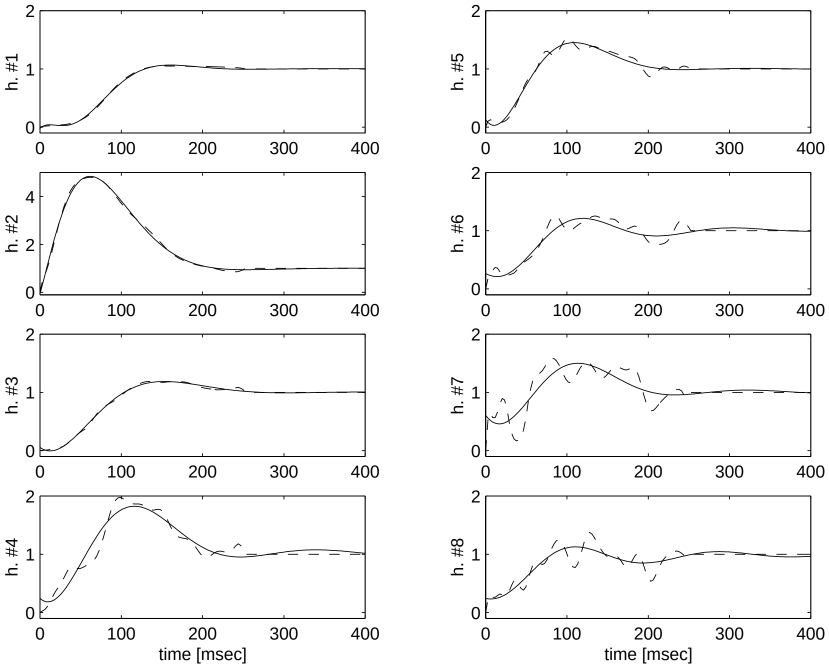

Figure 4: The envelopes of the first 8 harmonics of a Diapason pipe (dashed lines), and the fitted step-responses (solid lines)

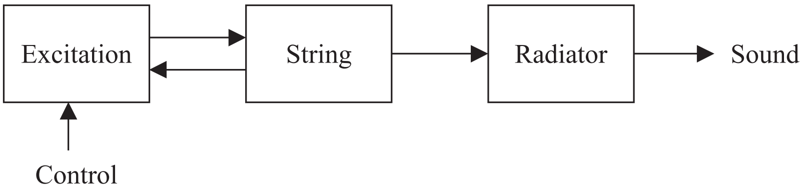

Figure 5: Model structure.

The model structure is displayed in Fig. 5. It can be seen that the interaction of the string and the excitation is bi-directional. This is because the interaction force depends on the previous state of the string too, and not only on the parameters of the excitation. This is taken into account by introducing a feedback from the string to the excitation model. As mentioned above, the body also influences the string vibration, so the interaction should be bi-directional also in this case. However, in our model the effect of the instrument body is split into two parts. The effect of providing a terminating impedance to the string is taken into account in the string model itself. The body model is responsible for the modeling of radiation properties. This way, the body can be modeled as a straightforward structure without feedback.

3.2 String Modeling

The wave equation of the ideal string is based on several simplifications: the length of the string is assumed to be infinite, its mass density μ and tension T is supposed to be homogenous and its displacement to be small with respect to string length, which means that its slope is very small ($dy/dx \ll 1$). Furthermore, only one transversal polarization of the string is taken into account. The result is the one-dimensional wave equation Eq. (2), which is similar to that of transmission lines or the longitudinal motion of bars, see, e.g., [20, 9].

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 y}{\partial t^2}, \quad c = \sqrt{\frac{T}{\mu}}, \tag{2}$$

where x is the position along the string, y is the transversal displacement, t stands for time, T for the tension, μ for linear mass density and c for the wave velocity. The equation shows that the acceleration of a small section of the string is proportional to the curvature of the string at that section. This equation can be directly discretized with respect to time and space, forming the algorithm called “finite differences” [21, 22]. The method has the advantage of being a purely physical approach, thus completely flexible (e.g., the mass density μ can vary along the string), but this complexity is not required for musical instrument modeling. For our purposes, a simpler, thus computationally less demanding method should be used.

3.2.1 The Digital Waveguide

A very efficient technique has been presented for string modeling in [23, 8]. The digital waveguide modeling is based on the discretisation of the solution of the wave equation, rather than the wave equation itself.

Every traveling wave which retains its shape is a solution of Eq. (2). Coming from the linearity of the string, the general solution is a superposition of two traveling waves; one of them going to the right, the other to the left direction [20, 9]:

$$y(x,t) = f^{+}(ct - x) + f^{-}(ct + x) \tag{3}$$

This equation holds for other wave variables (velocity, force, curvature) too. The digital waveguide model of the ideal string is obtained by sampling Eq. (3) temporally and spatially in a way that the two traveling waves move one spatial sampling interval during one time-instant [8]:

$$y(t_n, x_m) = y^{+}(n - m) + y^{-}(n + m) \tag{4}$$

This is implemented by two parallel delay lines, where the transversal displacement of the string is calculated by adding up the output of the samples of the two delay lines at the same spatial coordinate. This is illustrated in Fig. 6.

The termination of the string can be modeled by connecting the two delay lines at their endpoints. An ideally rigid termination corresponds to a multiplication of -1, meaning that the traveling waves are reflected with a sign change. In practice, the string is terminated by a frequency dependent impedance, introducing losses to the string vibration. This is taken into account by a digital filter $H_r(z)$ at the end of the delay line. Moreover, the distributed losses and dispersion of the string can also be approximated by the lumped reflection filter $H_r(z)$ [8]. Fig. 7 displays the digital waveguide model in its physical form, where M represents the length of the string in spatial sampling intervals, M_{in} denotes the position of the force input, and $H_r(z)$ refers to the reflection filter.

In practice, the four delay line structure of Fig. 7 can be implemented as one circular buffer, resulting in an extremely efficient realization. The computational complexity of the method depends on the order of the reflection filter $H_r(z)$, i.e., on the accuracy of the approximation of losses and dispersion, rather than on the number of simulated partials.

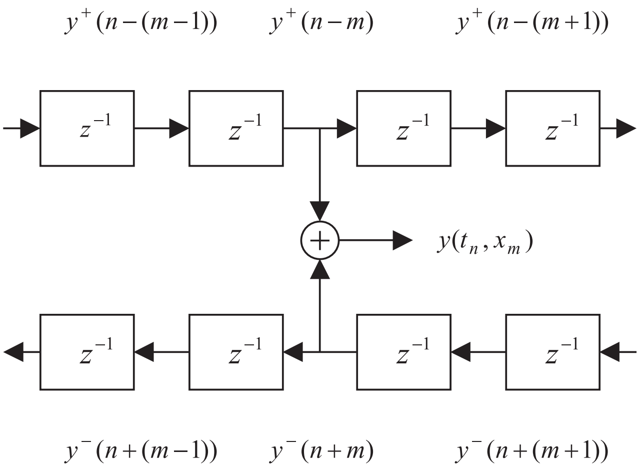

Figure 6: The principle of digital waveguide [23, 8].

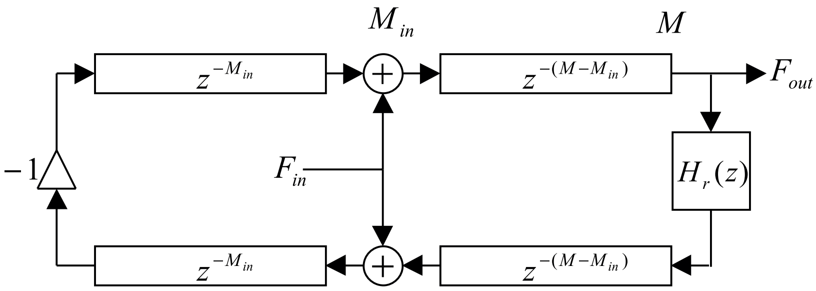

Figure 7: The digital waveguide with consolidated losses and dispersion

The impulse response of the digital waveguide is a quasi-periodic set of exponentially decaying sinusoids, whose frequencies and decay times can be controlled by the careful design of the reflection filter $H_r(z)$.

3.2.2 Reflection Filter Design

In practice, the model parameters are estimated from recorded tones, since that requires the measurement of one signal only. The partial frequencies produced by the digital waveguide of Fig. 7 are determined by the phase response of the reflection filter $H_r(z)$, together with the total length of the delay lines. On the other hand, the decay times of the partials are influenced by the magnitude response of the loss filter. Therefore, it is straightforward to split the design process into three independent parts: $H_r(z) = -H_l(z)H_d(z)H_{fd}(z)$, where $H_l(z)$ is the loss filter, $H_d(z)$ is the dispersion filter, and the fractional delay filter $H_{fd}(z)$ is required for fine-tuning the fundamental frequency of the string. Using allpass filters H_d(z) for simulating dispersion ensures that the decay times of the partials are controlled by the loss filter $H_l(z)$ only. The phase response of the loss filter is negligible compared to that of the dispersion filter. This way, the loss filter and the dispersion filter can be designed separately. Obviously, the dispersion filter $H_d(z)$ is implemented for those instruments only, where the inharmonicity is audible. In practice, this means that $H_d(z)$ is required for piano modeling only.

The string needs to be fine tuned because delay lines can implement only an integer delay and this provides too low a resolution for the fundamental frequencies. Fine tuning can be incorporated in the dispersion filter design or, alternatively, a separate fractional delay filter $H_{fd}(z)$ can be used in series with the delay line. In this study, we have used a first-order allpass filter for this purpose. Other type of fractional delay filters could be also used, [24] provides an exhaustive overview on their design.

Loss Filter Design

The role of the loss filter $H_l(z)$ is to set the decay times of the paretials. Therefore, the decay times of the recorded tone should be estimated, based on the amplitude envelopes of the partials [25]. The partial envelopes can be calculated as it is described in Sec. 2 about signal modeling. Alternatively, heterodyne filtering [26] or sinusoidal peak tracking utilizing Short Time Fourier Transform [25] could also be used. If the amplitude envelope of a partial is plotted in a logarithmic amplitude scale, the nearly exponential decay of the partial becomes approximately linear. Accordingly, the decay time and initial amplitude parameters can be estimated by linear regression

[25, 26].

The specification for the loss filter can be computed as follows:

$$g_k = \left| H_l \left(e^{j\vartheta_k} \right) \right| = e^{-\frac{k}{f_k \tau_k}}, \tag{5}$$

where τ_k is the decay time of partial k, f_k is the frequency of partial k and g_k is the desired amplitude value of the loss filter at the angular frequency ϑ_k. Fitting a filter to g_k coefficients is not trivial, even if the phase part of the transfer function is not considered. This is because of the special nature of the loss filter: the error in the decay time is a nonlinear function of the amplitude error. When the magnitude reponse of the loss filter exceeds unity, the stability of the feedback loop is at risk.

Designing the loss filter with respect to magnitude error has been extensively used in the literature, although it cannot overcome the problems mentioned above. The idea of optimizing the loss filter with respect to decay times was suggested in [23]. We have also developed filter design algorithms based on the decay-time error [27]. Decay-time based optimization assures that the overall decay time of the note is preserved and the stability of the feedback loop is maintained. This is because an unstable digital waveguide corresponds to negative decay times. Moreover, optimization with respect to decay times is perceptually more meaningful.

In [25, 26] a one-pole loss filter has been used for the syntheis of plucked instrument tones. The transfer function of such a filter is:

$$H_{1\mathrm{p}}(z) = g \frac{1 + a_1}{1 + a_1 z^{-1}} \tag{6}$$

where $-a_1$ is the pole of the filter and g refers to the DC gain. In [25, 26] such a filter was found to be adequate for simulating the acoustic guitar and other plucked string instruments. A great advantage of using a one-pole filter is that the stability of the waveguide loop can always be maintained by setting $a_1 < 0$ and $g < 1$. As for the design, [25, 26] used a simple algorithm for minimizing the weighted magnitude error in the mean squares sense. However, the overall decay time of the synthesized tone did not always coincide with the original one.

We have developed a more robust method for one-pole loss filter design [27]. The decay times of the partials produced by a digital waveguide with a one-pole loss filter can be calculated from the g and a_1 parameters of Eq. (6) as follows:

$$\begin{aligned} \tau &= \frac{1}{\sigma} \approx \frac{1}{c_1 + c_3 \vartheta^2} \\ c_1 &= f_0(1 - g) \\ c_3 &= -f_0 \frac{a_1}{2(a_1 + 1)^2} \end{aligned} \tag{7}$$

where ϑ is the angular frequency in radians, and f_0 is the fundamental frequency of the digital waveguide in Hz. It follows that from c_1 and c_3 coefficients g and a_1 can be easily calculated via the inverse of Eq. (7).

The goal is to minimize the mean-square error of the decay times, since that has been found to be a perceptually adequate criterion. The expression of the error e_τ is:

$$e_\tau = \sum_{k=1}^{K} (\hat{\tau}_k - \tau_k)^2 = \sum_{k=1}^{K} \hat{\tau}_k^2 \tau_k^2 \left(\frac{1}{\hat{\tau}_k} - \frac{1}{\tau_k} \right)^2 = \sum_{k=1}^{K} \hat{\tau}_k^2 \tau_k^2 (\hat{\sigma}_k - \sigma_k)^2 \qquad (8)$$

where $\sigma_k = 1/\tau_k$ are the prescribed, and $\hat{\sigma}_k = 1/\hat{\tau}_k$ are the approximated decay rates.

It can be noted from Eq. (7) that the decay rate $\sigma = 1/\tau$ is a second-order polynomial of ϑ. This means that its parameters c_1 and c_3 can be easily computed by means of polynomial regression. The parameters g and a_1 of the one-pole filter are calculated by the inverse of Eq. (7).

In most of the cases, tones synthesized using a one-pole loss filter sound realistic. However, the accuracy of the approximation can be increased by using higher order filters for $H_l(z)$. Computing analytical formulas for the decay times with high-order filters is a difficult task. A two-step procedure was suggested in [28].

We have presented a different approach [27], suggesting the transformation of the specification. Later, a simpler method was proposed for high-order filter design based on a special weighting function [29]. The resulting decay times of the digital waveguide are computed from the magnitude response $\hat{g}_k = |H(e^{j\vartheta_k})|$ of the loss filter by the inverse of Eq. (5):

$$\hat{\tau}_k = d(\hat{g}_k) = -1/(f_0 \ln \hat{g}_k) \qquad (9)$$

If the function $d(\hat{g}_k)$ is approximated by the first-order Taylor polynomial around the specification g_k, the mean-square error with respect to decay times is obtain by:

$$\begin{aligned} e_\tau &= \sum_{k=1}^{K} (\hat{\tau}_k - \tau_k)^2 = \sum_{k=1}^{K} (d(\hat{g}_k) - d(g_k))^2 \approx & (10) \\ &\approx \sum_{k=1}^{K} (d'(g_k)(\hat{g}_k - g_k))^2 = \sum_{k=1}^{K} w_k (\hat{g}_k - g_k)^2 & (11) \end{aligned}$$

which is a simple mean-squares minimization with weights $w_k = (d'(g_k))^2$, and can be done by any standard filter design technique.

The first derivate of $d(g_k)$ is $d'(g_k) = 1/(f_0 g_k (\ln g_k)^2)$, which can be approximated by $d'(g_k) \approx 1/(f_0(g_k - 1)^2)$. Since $1/f_0$ does not depend on k, it can be omitted from the weighting function. Hence, the weighting function becomes:

$$w_k = \frac{1}{g_k^2 (\ln g_k)^4} \approx \frac{1}{(g_k - 1)^4} \qquad (12)$$

The approximation of Eq. (10) is accurate only for $\hat{g}_k \approx g_k$, which means that the magnitude of the designed filter is close to the specification. On the contrary, the measured decay times τ_k have a great variance which cannot be followed by loss filters of reasonable order ($N < 20$). Therefore, it is worthwhile to smooth the decay time data τ_k, e.g., by convolving them with a triangular window $[0.25, 0.5, 0.25]$ before computing the specification g_k by Eq. (5). The phase specification of the loss filter is computed by the Hilbert transform [30] from the magnitude interpolated magnitude specification, corresponding to a minimum-phase filter.

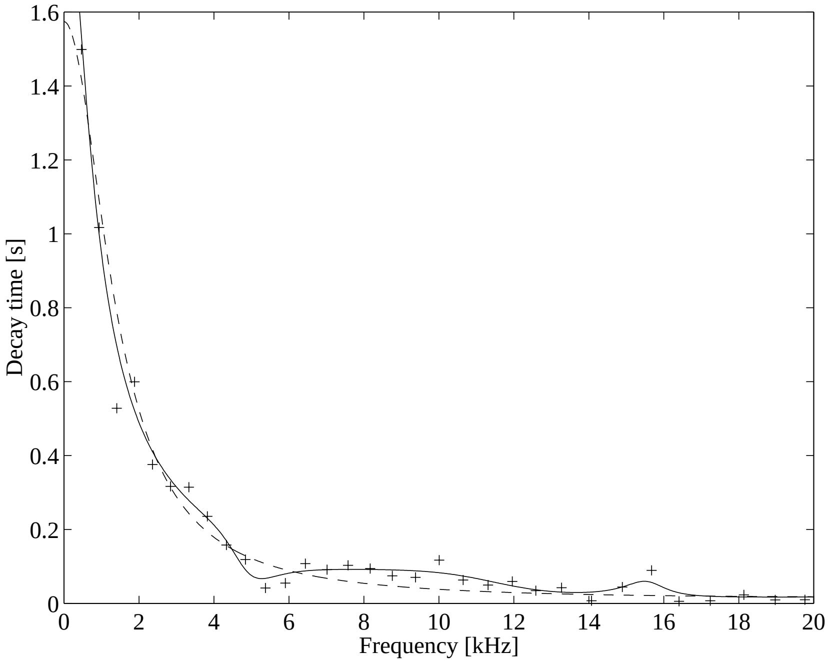

Figure 8: Loss filter design for $A_4^\sharp$ piano note: prescribed decay times (crosses), the decay times approximated by the one-pole loss filter (dashed-line), and by an 8th order loss filter designed by the method based on a special weighting function (solid line).

Fig. 8 depicts the results of loss filter design for a filter order of 8 with solid line. The measured decay times of the piano note $A_4^\sharp$ are displayed with crosses. The resulted decay times using a one-pole loss filter designed by polinomial regression are displayed with dashed line. It can be noted that the decay times of the first ten partials are modeled precisely already by the one-pole loss filter, and the general trend of decay times are followed. When compared to having a frequency independent loss filter $H_l(z) = r$, which would mean equal decay times for all the partials, the difference is dramatic. When higher order loss filters are used, the sound quality is increased only slightly. However, in some cases, it is still worth to use a filter with a higher order. For example, above 8 kHz the original decay times are the double of the ones calculated with the one-pole filter (dashed line in Fig. 8), which can be overcome by a high-order loss filter (solid line in Fig. 8).

In practice, we have used 3$^{\text{rd}}$ order loss filters for piano modeling, and one-pole loss filters for the modeling of the violin. This distinction is motivated by the fact that the piano has a decaying tone, therefore the decay rates have a great perceptual importance. On the contrary, the violin is a continuously excited instrument, where the precise rendering of the decay rates are less significant for the listeners.

Dispersion Filter Design

In the case of piano modeling, the audible effect of string dispersion cannot be neglected. Dispersion denotes an increase in wave velocity for higher frequencies. This can be modeled by having a "shorter" delay line for the higher partials than for the lower ones. For that, a filter with a non-constant phase delay is required. Since the magnitude response of the reflection filter $H_r(z) = -H_l(z)H_d(z)H_{fd}(z)$ should only be affected by the loss filter $H_l(z)$, it is straightforward to use an allpass filter as dispersion filter $H_d(z)$. For the design, we use the method presented in [31, 32].

The desired phase delay D_k at the partial frequency f_k is given by

$$D_k = \frac{f_s k}{f_k} - N_{wg} - D_l(f_k) \tag{13}$$

where N_{wg} is the total length of the waveguide delay line and $D_l(f_k)$ is the phase delay of the already designed loss filter. The designed method proposed in [32] minimizes the weighted squared phase error between the desired phase response $\Phi_k = \vartheta_k D_k$ and the phase response of the dispersion filter Φ_d. We first compute the quantities:

$$\beta_k = -\frac{1}{2}(\Phi_k + N\vartheta_k) \tag{14}$$

and solve the equations for a_i:

$$\sum_{i=1}^{N} a_i \sin(\beta_k + k\vartheta_k) = -\sin\beta_k \qquad k = 1, 2, \ldots M \tag{15}$$

where a_i are the coefficients of the filter denominator, with $a_0 = 1$.

Since the number of prescribed phase values M is higher than the filter order N, the set of equations is overdetermined, thus it cannot be precisely solved. However, when the equation error is minimized in the mean-squares sense, the solution is easily computed. During the minimization a weighting function has to be used which depends on the magnitude response of the denominator of $H_d(z)$. Accordingly, the least-squares optimization has to be run iteratively. In practice, a filter order $N = 16$ is required for the lowest piano notes to provide good results, while for the middle register a fourth order dispersion filter has been found to be enough.

3.2.3 Modeling Beating and Two-Stage Decay

In real pianos, except for the lowest octave, the vibration of two or three strings are coupled through the bridge, when one note is played. This produces beating and two-stage decay in the sound [33]. This effect can be simulated by having two coupled waveguides in parallel [34], but this leads to high computational cost and complicated parameter estimation.

Instead, we suggest to use some second-order resonators $R_1 \ldots R_K$ parallel with the string model $S_v(z)$ [27, 35]. This is depicted in Fig. 9. The transfer function of the resonators $R_k(z)$ are as follows:

$$R_k(z) = \frac{Re\{a_k\} - Re\{a_k\overline{p_k}\}z^{-1}}{1 - 2Re\{p_k\}z^{-1} + |p_k|^2 z^{-2}}$$

$$a_k = A_k e^{j\varphi_k} \qquad p_k = e^{j\frac{2\pi f_k}{f_s} - \frac{1}{f_s \tau_k}} \tag{16}$$

where A_k, φ_k, f_k, and τ_k refer to the initial amplitude, initial phase, frequency and decay time parameters of the k$^{\text{th}}$ resonator, respectively. The overline stands for complex conjugation, the Re sign for taking the real part of a complex variable, and f_s is the sampling frequency.

The resonators are tuned close to the frequency of the distinct partials of the digital waveguide. Thus, every partial corresponds to two slightly mistuned sinusoids with different decay times and amplitudes, and their superposition produces beating and two-stage decay. The efficiency of the structure comes from the fact that only those partials have parallel resonators, where the beating and two-stage decay are prominent. The others have simple exponential decay determined by the digital waveguide model $S_v(z)$.

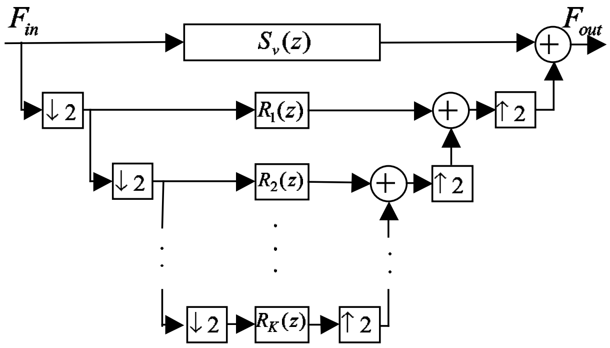

Figure 9: The multi-rate resonator bank.

The efficiency is further increased by running these resonators at a lower sampling rate [35]. The input signal F_{in} is of a lowpass character, therefore, when downsampled without prior lowpass filtering, and only the lower half of the downsampled frequency band is used, just a small aliasing occurs. This is acceptable, since a small aliasing at the input of the resonators only changes the initial amplitudes and phases of the sinusoids. Therefore, no anti-aliasing filters are required prior to the downsampling operation. The interpolation filters cannot be neglected. However, since only the half of the frequency bands are used, their specification can be simple (passband: $0 \leq \vartheta \leq 0.25\pi$, stopband: $0.75\pi \leq \vartheta \leq \pi$), which results in a 7$^{\mathrm{th}}$ order FIR filter for 60 dB stopband rejection. The interpolation filters used after upsampling operation can be common for all the notes played in the same time. Note that this is not depicted in Fig. 9.

The parameter estimation of the model is done by first estimating the partial envelopes of measured sound-pressure signals of pianos. Then, the deviation is computed from the general exponential decay. This is followed by fitting an exponentially decaying or growing sinusoid on the deviation, which completely characterizes the two-mode model [27].

This method provides significant computational savings compared to having a second waveguide in parallel (5–10 operations/cycle instead of 30–40). Moreover, the parameter estimation simplifies to finding the parameters of the mode-pairs. The stability problems of a coupled system are also avoided. As an example, Fig. 10 displays the first 8 partial envelopes of a recorded $A_4^\sharp$ note (466Hz). Fig. 11 shows the output of the synthesis model using the structure of Fig. 9 with 5 resonators. It can be seen that the envelopes of the first five partial are precisely rendered. Above the sixth partial, where the resonators are not implemented, the amplitude envelopes have simple exponential decay.

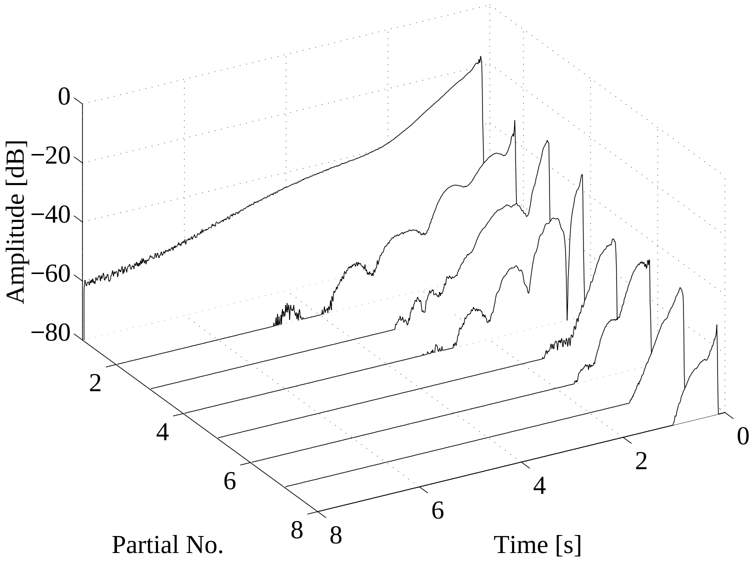

Figure 10: Partial envelopes of the original $A_4^\sharp$ note.

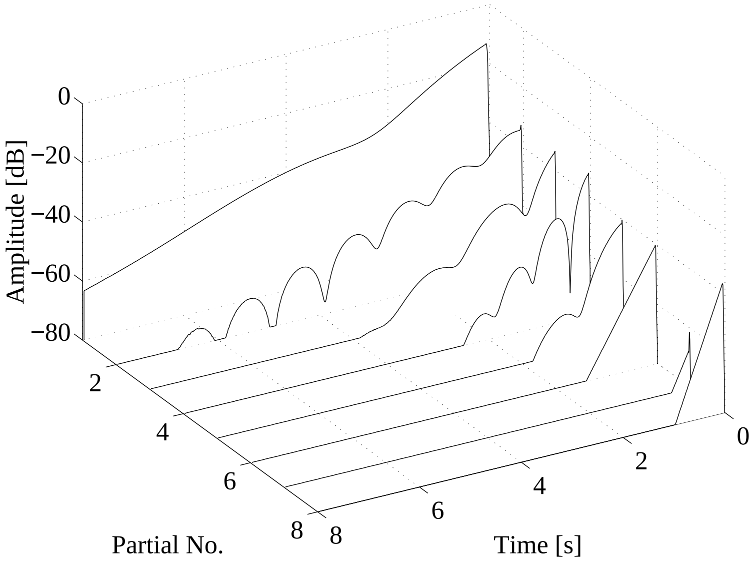

Figure 11: Partial envelopes of the synthesized $A_4^\sharp$ note.

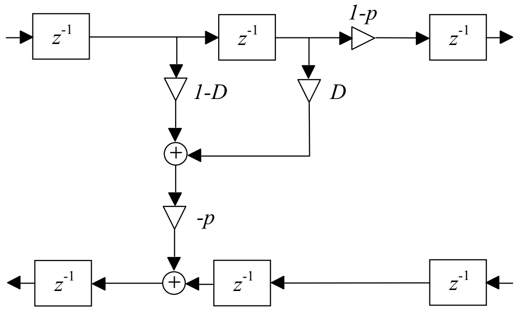

Figure 12: Simplified model of a string terminated by a finger

3.2.4 Finger Modeling

For the violin, not only the losses of the string and the termination have to be modeled, but the effect of the player's finger as well. On the piano, there is (at least) one string for each note. On the violin, the player has to use his fingers to change the length of the strings and thus to change the fundamental frequency of the tone. These note transitions are important in determining the character of the instrument. When playing flautato sounds, by touching the string slightly at a node of vibration, the basic frequency is damped leaving only higher harmonics.

Physically the finger acts like a damper attached to the string. In order to model it, a scattering junction with variable position and coupling should be inserted into the delay line. The frequency dependent low pass filtering effect of the finger can be realized within the reflection filter $H_r(z)$ as well. The scattering junction is similar to modeling finger holes in woodwinds [36].

In our experiments we used a simplified junction combined with a simple fractional delay for fine tuning the frequency of the sound (see Fig. 12). With the increase of the pressure of the finger (p), the right side of the delay lines gets less signal. Finally, when $p = 1$, the shortened left side string is terminated properly with -1 (and tuned with the fractional delay, D).

The finger model described above is not capable for the accurate modeling of the finger transitions. It models only the transitions from an open string note to a finger-terminated one and vice versa. However, in the most cases the player uses another finger to change from one note to the other, therefore two finger junctions need to be included in the model. In practice, two types of transitions have to be simulated depending on the direction of change. Changing to a higher note requires that the first finger is already on the string and second one is being used normally, with increasing finger pressure. Changing to a lower note may assume that the second finger is already in its place (behind the other) while pressure of the first finger is lowered to zero.

With the choice of the shape and the speed of pressure change several effects can be modeled.

Furthermore, also differences in the four strings of the violin can be considered to refine the model. Each string has its own properties (fundamental frequency, specific impedance, stiffness, etc.), thus, each has different tone. The player has the flexibility of choosing a string for a given note. The decision depends on the pitch of the actual note, the notes following and preceding the actual one and the timbre he wants to achieve. The same note played on a lower string results a more flat and soft tone than played on a higher one. When a new note is started on a new string, a previously excited open string or finger-terminated string might still vibrate, or the latter might change to open string (if the player lifts away his finger). When off-line synthesis is used, these subtleties can be set individually for each tone manually, or general transition rules can be formed to take them into account. On the contrary, In on-line synthesis only general rules can be used for easier controllability.

3.3 Body Modeling

The radiation of the soundboard or any instrument body is generally treated as a linear filtering operation acting on the string signal. Thus, body modeling reduces to filter design. Theoretically, this filter should be somewhat different for all the strings. This is feasible for the four strings of the violin, but for modeling the piano having hundreds of strings, this would lead to unacceptably high computational load. In practice, the string signals are summed and lead trough a single body filter to reduce the required computational complexity.

Unfortunately, instrument bodies exhibit a high modal density, therefore high order filters are needed for their simulation. In the case of the guitar body, the required filter order was about 500 [37]. We have found that the piano requires even higher filter orders. In the case of FIR filters, 2000 taps were necessary to provide high quality sound. Commuted synthesis [38] could overstep this problem, but that would require simplifications in the excitation model. Feedback delay networks [39] are capable of producing high modal density at a low computational cost, but due to the difficulties in parameter estimation, they have not been used for high-quality sound synthesis.

To resolve this problem, we have proposed a novel multi-rate approach for instrument body modeling [40]. The string signal F_s is split into two parts: the lower is downsampled by a factor of 8 and filtered by a high order FIR filter $H_{low}(z)$, precisely synthesizing the instrument body response up to 2 kHz. Above 2 kHz, only the overall magnitude response of the body is mod-

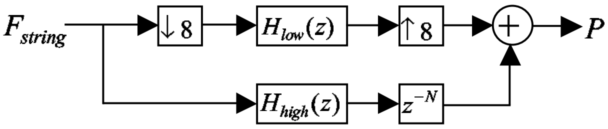

Figure 13: The multi-rate body model.

eled by a low order FIR filter $H_{high}(z)$. This signal is delayed by N samples to compensate for the delay of the decimation and interpolation filters of the low frequency chain. The crossover frequency of 2 kHz was determined by conducting informal listening tests. The body model is depicted in Fig. 13.

The decimation and interpolation filters are chosen to be the same filter $H_{di(z)}$ and designed with a loose specification (5 dB passpand ripple, 60 dB stopband rejection) by the Matlab's `remez` algorithm. This results in an order of 96, which corresponds to 12 instructions/cycle in a polyphase implementation.

The body filters $H_{low}(z)$ and $H_{high}(z)$ are designed from measurements. For the piano, the soundboard was excited by an impact hammer and the force and pressure signals were simultaneously recorded. Then, the force-pressure transfer function was calculated and a 2000 tap FIR target filter $H_t(z)$ was obtained by truncating the measured impulse response. The target response $H_t(z)$ is lowpass filtered and then downsampled by a factor of 8 to produce the desired low frequency filter $\tilde{H}_{low}(z)$. This has to be changed in order to correct the passband errors of the decimation and interpolations filters:

$$H_{low}(z) = \tilde{H}_{low}(z)\frac{1}{H_{di}^2(z^{\frac{1}{8}})} \tag{17}$$

This results in a 250 tap long FIR filter, which consumes 31.25 operations/cycle.

The high frequency filters are computed by subtracting the impulse response of the low frequency chain from the target response. This residual signal contains energy mainly above 2 kHz. The residual is then made minimum-phase and truncated to a length of 50 taps to produce the high frequency filter $H_{high}(z)$.

As an example, the magnitude response of a piano soundboard model is depicted in Fig. 15. The magnitude response of the target filter $H_t(z)$ is depicted in Fig. 14 for comparison. It can be seen from the figures that the magnitude response is accurately preserved up to 2 kHz. Although not visible, but so is the phase response. Above 2 kHz, only the overall magnitude response is retained.

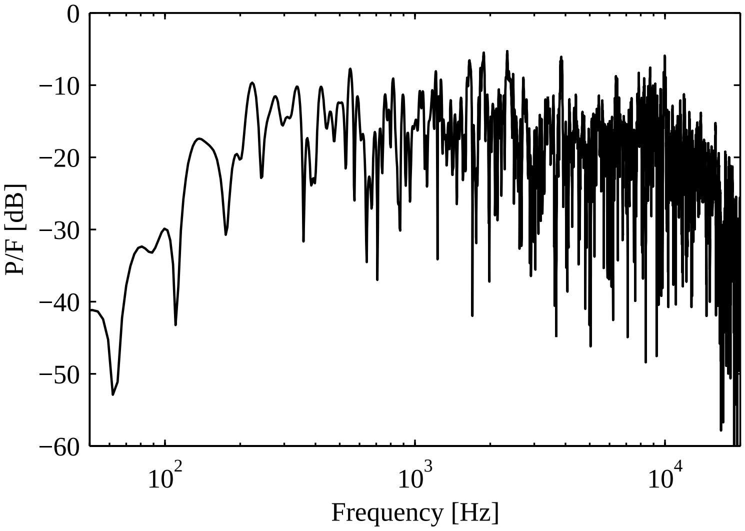

Figure 14: The mangnitude transfer function of the 2000 tap target filter $H_t(z)$.

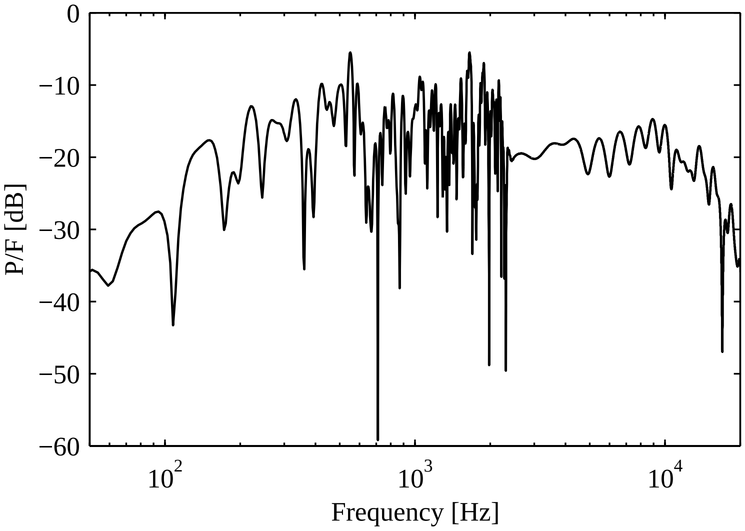

Figure 15: The magnitude transfer function of the multi-rate body model.

This structure is capable to produce high sound quality at around 100 instructions per cycle and provide a very similar sonic character compared to the reference filter $H_t(z)$. The only shortcoming of the model is that the attack of high notes sounds somewhat sharper compared to the target filter. This is because now the energy of the soundboard response in the high frequency range is concentrated to a very short time period, i.e., the resonances are not sharp enough above 2 kHz. As a straightforward solution, the downsampling factor of 8 in Fig. 13 can be decreased to 4, but that approximately doubles the computational load. Here again, a tradeoff has to be found between quality and efficiency. However, when the high frequency filter $H_{high}(z)$ would be implemented with a simple structure which is capable of producing high modal density, this limitation could be overcome. For that, the feedback delay network [39] could be a good candidate. Similar techniques can be applied for modeling the high modal density impulse response of the violin.

3.4 Excitation Modeling

The string and body models are of the same structure for the different string instruments, although they are parametrized in a different way for the piano and for the violin. On the contrary, for modeling the excitation, different model structures has to be developed. This is because the excitation mechanisms of the instruments are completely different, and their precise implementation is essential for rendering the sonic characteristics of these instruments.

3.4.1 The Hammer Model

The piano string is excited by a hammer, whose initial velocity is controlled by the player with the strength of the touch on the keys. The excitation mechanism of the piano is as follows: the hammer hits the string, the hammer felt compresses and feeds energy to the string, than the interaction force pushes the hammer away from the string. Therefore, the excitation is not continuous, it is present for some milliseconds only. The hardwood core of the hammer is covered by wool felt, whose structure is not homogenous. This is the reason why playing harder on the piano results not only in a louder tone, but also in a spectrum with stronger high frequency content.

The piano hammer is generally modeled by a small mass connected to a

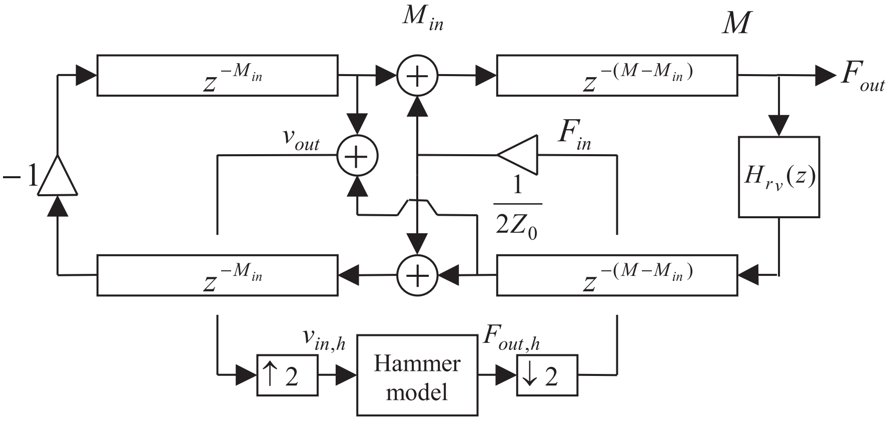

Figure 16: The string model connected to the hammer.

nonlinear spring [41]. The equations describing the interaction are as follows:

$$F(t) = f(\Delta y) = \begin{cases} K(\Delta y)^p & \text{if}\Delta y > 0 \\ 0 & \text{if}\Delta y \leq 0 \end{cases} \tag{18}$$

$$F(t) = -m_h \frac{d^2 y_h(t)}{dt^2} \tag{19}$$

where $F(t)$ is the interaction force, $\Delta y = y_h(t) - y_s(t)$ is the compression of the hammer felt, where $y_h(t)$ and $y_s(t)$ are the positions of the hammer and the string, respectively. The hammer mass is referred by m_h, K is the hammer stiffness coefficient, and P is the stiffness exponent.

These equations can be easily discretized with respect to time. However, as seen from Eqs. (18) and (19), there is a mutual dependence between $F(t)$ and $y(t)$, i.e., for the calculation of one of these variables, the other should be known. This is generally overcome by the assumption that the hammer force changes a little during one time step, that is $F(t_n) \approx F(t_{n-1})$. Although leading to numerical instabilities for high impact velocities, the straightforward approach is used in the literature (see, e.g., [42]. The numerical instabilities can be avoided by rearranging the nonlinear equations to known and unknown terms [43].

We have suggested to use a simpler approach for avoiding the numerical instability [44]. The proposed multi-rate hammer model is depicted in Fig. 16. The stability of the discrete system can always be assured with a sufficiently high sampling rate f_s, since for $f_s = \infty$, the discrete-time system will behave as the continuous-time equations. However, increasing the sampling rate of the whole model would lead to unacceptable computational

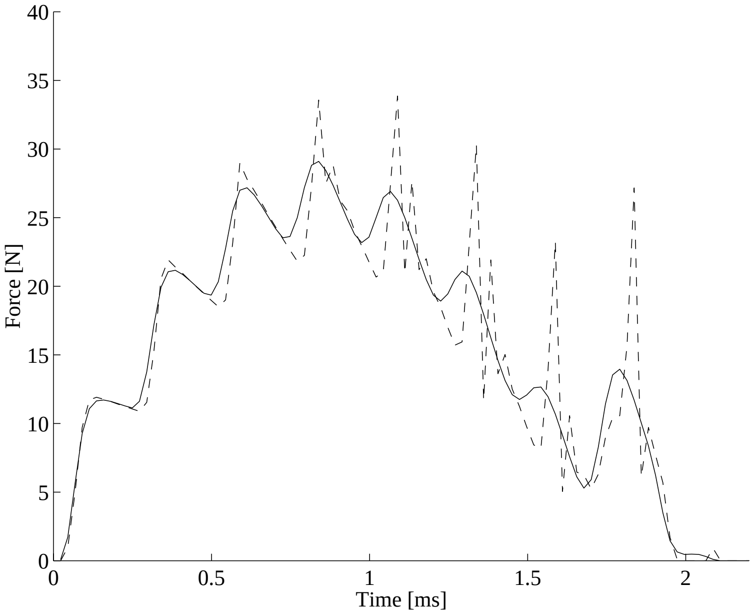

Figure 17: Simulated interaction forces for note C_5 (522 Hz), with an impact velocity of 6 m/s: straightforward approach with numerical instability (dashed-line) and the multi-rate hammer (solid line).

overhead. When only the sampling rate of the hammer model is increased, it leads to a small computational overhead, while still assures that $F(t_n)$ changes a little at every time-step. Implementing the hammer at a double sampling rate has been found to provide stable results. For downsampling (↓ 2 in Fig. 16) simple averaging, for upsampling (↑ 2 in Fig. 16) linear interpolation are used.

In Fig. 17 the interaction force is shown for note C_5 (522 Hz). For the simulation, an ideal digital waveguide model was used, without any dispersion or losses. The parameters of the hammer were taken from [42], C_4 hammer. The impact velocity was $v_{h0} = 6$ m/s. The dashed line refers to the single-rate hammer model with $f_s = 44.1$ kHz. The solid line is the force of the multi-rate implementation, by using $f_s = 44.1$ kHz for the waveguide model. It can be seen that the straightforward technique operating at normal sampling rate goes unstable, while the output of the multi-rate hammer model produces well behaving force signals, similar to those that can be measured for real hammers.

3.4.2 The Bow Model

In the case of bowed instruments the excitation is based on the sticking friction between the string and the bow hairs. The bow, moving perpendicular to the string, grips the string (gripping phase). This friction force is highly nonlinear. Due to the increasing displacement of the string, the elastic returning force is also increasing until its level reaches the sticking friction. At this point the bow releases the string, the string swings back (release phase) and then vibrates freely. This vibration is damped partly by the own losses of the string and partly by the slipping friction that develops between the string and the bow hairs. This state lasts as long as the bow grips the string again, which occurs only when the velocity of the bow and the string equals. In this case, their relative velocity is zero, the frictional force is maximal. This alteration of the stick and slip phases is the so-called Helmholtz motion. The excitation is periodical and generates a sawtooth shape vibration (while at the piano the excitation by the hammer is impulse like).

The excitation depends on several control variables. As the primary control variable is the velocity of the bow, the traveling-wave component in the digital waveguide is the velocity for modeling bow instruments. Other important variables are the force of the bow exerted on the string and the position of the bow along the string. Less important factors are the angle between the bow and the string, the size of the contact surface of the bow, and the grip of the bow hair (which can be increased by rosin). In order to keep the model manageable and realizable, usually only the primary and some other important variables (such as the bow force and position) are taken into account.

The bow-string interaction is usually modeled by a scattering junction [45] (Fig. 18). This junction is controlled by differential velocity (v_{Δ}^{+}), which is the difference of the bow velocity and the current string velocity. The position of bowing determines the insertion point of the junction into the delay lines. Other control variables (bowing force and angle, etc.) are changed by modifying the parameters of the reflection function ($\rho(v_{\Delta}^{+})$). This function also depends on the characteristic impedance of the string and on the friction coefficient between the bow and the string.

At a given point of the string, the string velocity is obtained by adding the right-going velocity sample to the left-going velocity sample at the same point of the delay lines:

$$v_s = v_{s,l}^{+} + v_{s,l}^{-} = v_{s,r}^{+} + v_{s,r}^{-}, \tag{20}$$

where v^{-} and v^{+} are the traveling wave components within the string going towards a termination (resulting from the bow-string interaction) and coming

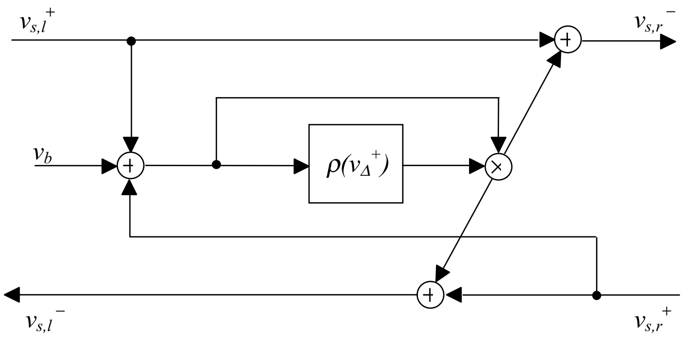

Figure 18: The scattering junction for modeling the bow-string interaction

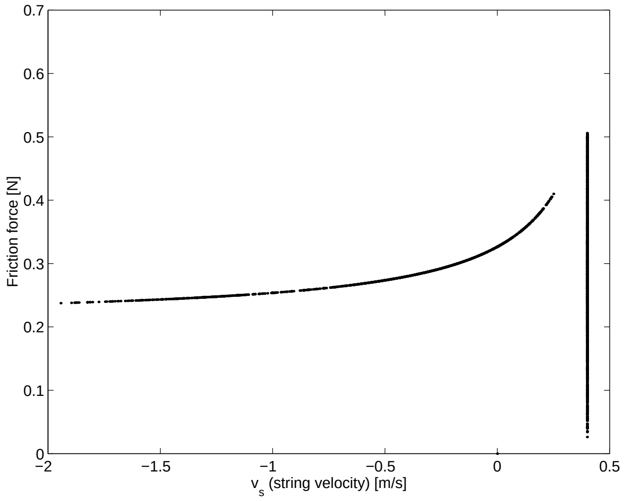

Figure 19: The nonlinear function of friction versus string velocity; simulation results with $v_b = 0.4$ m/s; $f_b = 0.7$ N

from (reflected by) a termination, respectively.

The bow-string interaction can be described as follows:

$$\begin{aligned} v_{s,r}^{-} &= v_{s,l}^{+} + \rho(v_\Delta^{+})v_\Delta^{+} \\ v_{s,l}^{-} &= v_{s,r}^{+} + \rho(v_\Delta^{+})v_\Delta^{+}, \end{aligned} \tag{21}$$

where ρ is the reflection function. The nonlinear function of friction versus string velocity can be seen on Fig. 19. We note that this model was refined in [46] by deriving the interaction from adhesion between two bodies in contact.

Besides modeling the bow-string interaction the player has to be modeled as well. The problem of modeling the left hand was discussed in Sec. 3.2.4. An exact model of the right (bowing) hand should provide enormous degrees of freedom using interactive controllers. However, this would result again an unmanageable instrument, and/or it would require a real violin player at the control keyboard/violin. Similarly to the proposed finger model, this problem can also be resolved by an automatic system based on real playing styles on bowed instruments. For each bowing style the time variations of the primary control variables can be represented by characteristic envelopes, so only one parameter needs to be adjusted for a given style. A MIDI based implementation of this idea can be found in [47].

4 Comparison of the Two Synthesis Methods

Here we compare the two methods described in this paper, namely the signal modeling based on envelope-filters and the physical modeling based on digital waveguides. When mentioning signal modeling and physical modeling throughout this section, we are referring to these two models covered in the paper. As our signal model describes the partial envelopes by linear filters, even theoretical connections can be found between the two methods. The theoretical investigations are followed by practical considerations.

4.1 Theoretical Connections

We show that the impulse response of both formulations can be expressed as a sum of exponentially decaying sinusoids, which can be realized as a resonator bank. Naturally, the resonator bank implementation is not an efficient realization, its only purpose is to serve as a common base for the comparison of the two methods. We show that for certain circumstances the two modeling approaches produce the same output signal.

4.1.1 The Signal Model

Recalling (1), the signal model was based on the idea of switching on a sine wave when a note is played and multiplying it with the attack and decay envelope of the given harmonics:

$$y_{i,k} = h_{i,k} A_i \cos(2\pi(if_0/f_s)k + \varphi_i) = h_{i,k} x_{i,k}, \ (i = 1..N). \tag{22}$$

Here the attack envelope $h_{i,k}$ is realized as step responses of 2nd or 3rd order filters.

The step response can be further rewritten as

$$h_{i,k} = w_{i,k} * \varepsilon_k, \tag{23}$$

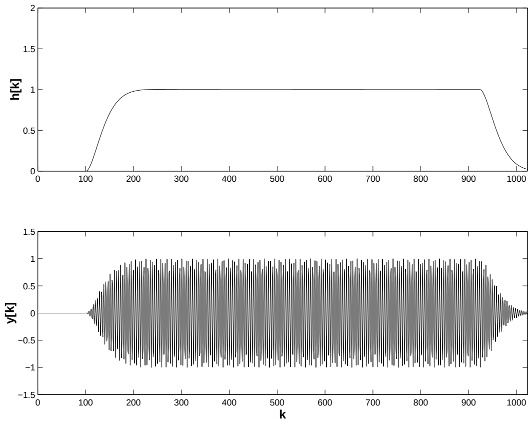

Figure 20: Signal model of a given partial realized with envelope filters. (a) step response of the envelope filter($h[k]$); (b) output of the system ($y[k] = h[k]x[k]$).

where $w_{i,k}$ is the impulse response of the filter, ε_k is the step function and $*$ denotes convolution.

The main effects of Eq. (22) in the time domain are depicted in Fig. 20.

Multiplying in the time domain with a sine wave is a simple modulation. Hence, in the frequency domain it becomes convolution of the sine wave and the step response of the signal, i.e.

$$Y(z) = (W(z)E(z)) * X(z). \tag{24}$$

Since this is a clear modulation of the sine wave with the step response of the envelope filter, the above equation can be rewritten as follows:

$$Y(z) = (W(z) * X(z))(E(z) * X(z)) = R(z)(E(z) * X(z)). \tag{25}$$

Note that $R(z) = W(z) * X(z)$ in the time domain is $r[k] = w[k]x[k]$, i.e. a sine wave multiplied with a second order system's impulse response. In the frequency domain, the convolution with the sine wave shifts up the original filter poles located at DC to the frequency of the sine wave. Thus, this expression can be realized with the same number of resonators as the number of poles of the original filter. The input to these resonators is the sine wave triggered by the trigger signal $\varepsilon[k]$. Fig. 21 shows some time-domain signals of this realization.

Thus, the signal model with envelope filters applied to the partials of the sound can be realized with a set of resonators. The required number of resonators depends on the number of partials to be generated and the order of the filters.

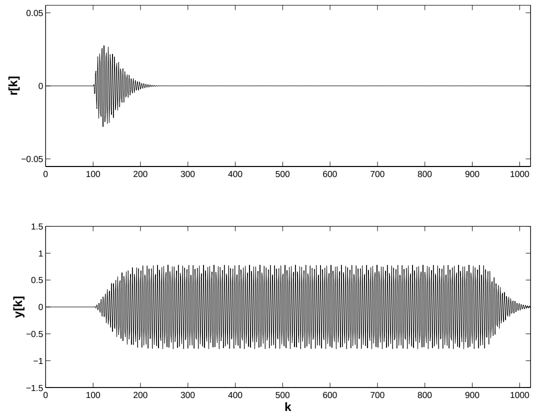

Figure 21: Signal model of a given partial realized with resonators. (a) impulse response of the two resonators; (d) output of the system.

4.1.2 The Physical Model

The transfer function of the digital waveguide model of Fig. 7, assuming that the reflection filter is constant for all the frequencies (i.e, $H_r(z) = -r$, $0 < r < 1$) is:

$$\frac{F_{out}}{F_{in}} = \frac{1}{1 - rz^{-N}} \left(1 + z^{-2M_{\mathrm{in}}}\right) z^{-(M-M_{\mathrm{in}})} \tag{26}$$

After the fractional expansion of the denominator of Eq. (26) we obtain the transfer function of a set of complex exponentials:

$$\begin{aligned} \frac{F_{out}}{F_{in}} &= \left\{ \frac{a_1}{1 - z^{-1} r_1 e^{j\vartheta_1}} + \ldots + \frac{a_N}{1 - z^{-1} r_N e^{j\vartheta_N}} \right\} \\ & a_k = j\frac{2}{N} \sin(2k\pi \frac{M_{in}}{N}) e^{-j\vartheta_k M} \\ & r_1 = \ldots = r_N = r^{\frac{1}{N}} \end{aligned} \tag{27}$$

where $\vartheta_k = (2k\pi)/N$ is the frequency of the k^{th} mode, $N = 2M$ is the total length of the delay line, a_k are the complex amplitudes and r_k are the pole radii. The impulse response $h(n)$ of the digital waveguide can be obtained from Eq. (27) by the inverse Z transform:

$$h(n) = \sum_{k=1}^{N} a_k \left(r_k e^{j\vartheta_k}\right)^n = \sum_{k=1}^{N/2} a_k \left(r_k e^{j\vartheta_k}\right)^n + a_{N-k} \left(r_{N-k} e^{j\vartheta_{N-k}}\right)^n \tag{28}$$

Because $\vartheta_{N-k} = 2\pi - \vartheta_k$, follows that the corresponding pole pairs will be conjugate pairs $r^{N-k} e^{j\vartheta_{N-k}} = r^k e^{-j\vartheta_k}$, and so the amplitudes $a_{N-k} = \overline{a_k}$, where the overline refers to complex conjugation. Therefore the impulse response $h(n)$ can be expressed as a sum of exponentially decaying sinusoids:

$$h(n) = \sum_{k=1}^{N/2} r_k^N \left(a_k e^{j\vartheta_k n} + \overline{a_k} e^{-j\vartheta_k n}\right) = \sum_{k=1}^{N/2} |a_k| r_k^N \sin\left(\vartheta_k n + \varphi_k\right) \qquad (29)$$

where $|a_k|$ is the magnitude, and φ_k is the phase of the complex coefficient a_k.

It can be seen from Eq. (29) that the impulse response of the digital waveguide with $H_r(z) = -r$ is the sum of exponentially decaying sinusoids, whose frequencies are equally distributed on the unit circle, and their decay rates are equal. For an arbitrary reflection filter $H_r(z)$ the modal frequencies and decay times cannot be derived in a closed form, however, they can be determined by numerical iterations. In any case, the digital waveguide can always be substituted by a set of parallel resonators. Their impulse response are exponentially decaying sinusoids with arbitrary initial amplitudes and phases, thus, they can be implemented as second order IIR filters in parallel.

Similar derivations with a different formulation were presented in [34], and it was shown that if two or three waveguides are coupled, the partials can be expressed by the sum of two or three sinusoids. Obviously, when the beating and two-stage decay of the piano is modeled by the multi-rate resonator bank of Sec. 3.2.3, the equivalent resonator structure can be obtained by adding the parallel resonators $R_1 \ldots R_k$ of Fig. 9 to the equivalent resonators of the waveguide. In this case, two resonators will correspond to some of the partials.

Note that here the digital waveguides have been treated only. However, the impulse response of other string models (e.g., based on finite differences) can also be expressed as a sum of exponentially decaying sinusoids, if there are no multiple poles in the transfer function.

So far, the digital waveguide model has been substituted by a set of resonators connected in parallel, behaving the same way as the original string model. Now the question is in which cases the signal model of Sec. 2 can produce an equivalent output compared to the digital waveguide. In the case of the piano, the hammer excitation is impulse-like, thus, its main role is to set the initial amplitudes of the partials. After the hammer has left the string, the partial envelopes decay exponentially in the string signal (here we neglect the transients introduced by the soundboard). Therefore, for a specific hammer velocity, each partial can be modeled by a sine generator connected to an envelope-filter.

4.1.3 The link

In the case of the piano, for a given hammer velocity, the signal model produces the same output as the physical model, except the initial transients. This is because our signal model is closer to the physical structure than the signal models used in general. By using digital filters as envelope generators, rather than arbitrary amplitude envelopes stored in a table, we assume that each partial can be described by a set of exponentially decaying sinusoids, which is indeed the physical reality for impulsively excited instruments, such as the piano or guitar.

For the violin and for the organ the link between the physics of the instruments and the envelope-filter based signal model is not as clear as for the piano. As these two instruments are continuously excited, and their excitations are of nonlinear nature, the partials cannot be synthesized by a set of exponentially decaying sinusoids. Accordingly, the partial envelopes cannot be precisely described by linear filters. From a physical point of view, the organ pipe can be modeled by a single digital waveguide connected to a nonlinear exciter. In our approach this nonlinear system is modeled with a linear system of a higher order. Third order envelope-filters have been found to be adequate for modeling the organ sound, this is equivalent to three digital waveguides coupled to each other. In other words, three linearly excited and coupled acoustic tubes produce similar sound to one tube connected to a nonlinear exciter. The same holds for violin synthesis: although the excitation is highly nonlinear, for a given playing style the partial envelopes could be modeled by linear envelope-filters. Obviously, different envelope-filter parameters would be required for all the different playing styles.

4.2 Practical Considerations

In this section, the signal-based and the physics-based approach is compared, from the point of view of their applicability. The main features of the methods are listed in Table 1. Then the most important properties of the different instruments covered in this paper are described, serving as a base for the choice among the synthesis methods. We note that an exhaustive evaluation of many different sound synthesis methods can be found in [2].

4.2.1 The Methods

The signal-based approach models the sound of the instrument itself. Accordingly, it does not make any assumptions on the structure of the musical instrument, only that the generated sound is periodic. Therefore, it can model a wide range of instrument sounds, since they differ only in their parameters, not in the model structure, which is always a set of sinusoids. As it is a general representation, its parameter estimation is simple, basically reduces to tracking partial envelopes, which can be easily automated. In general, a large amount of data is required to describe a given tone, but this specific tone, from which the parameters originate, is almost perfectly reproduced. As the structure of the instrument is not modeled, the interaction of the musician cannot be easily taken into account, meaning that, e.g., for different bow forces or velocities in the case of the violin different parameter sets are required for resynthesis. In practice, this means that for a single note the analysis procedure has to be run for all the different playing styles that a player can produce, and a large amount of data has to be stored or transmitted. As it treats the notes separately, the interaction of the different notes, e.g., the coupled vibration of strings, cannot be modeled. Changing the parameters of the synthesis program directly is not user-friendly: dozens of parameters can be changed, which all influence the sound in a different way compared to musicians got used to it in the case of real instruments. The quality and the computational load of the synthesis is usually varied by changing the number of simulated partials, which is probably not the best way from a perceptual point of view.

The physics-based approach models the functioning of the instrument, rather than the produced sound itself. It makes assumptions about the instrument it models, therefore, it looses generality. A piano model, e.g., cannot be used for violin modeling by just changing its parameters, since the excitation model is completely different for the two instruments. Consequently, the parameter estimation cannot be completely automated, at least the model structure has to be determined by the user. As the model structure already describes the main features of the instrument, only small number of parameters are needed, and modifications to these parameters produce perceptually meaningful results. For example, the user now controls the bow force, rather than the loudness of a single partial, and the instrument reacts in a way as a real violin would do. Therefore, only one parameter set is required for one note, since the different playing styles according to the interaction of the musician are automatically modeled. As it describes the physical structure, the interaction of the different model parts are also taken into account, e.g., the string coupling on the piano is easily modeled. A drawback that none of the tones will be perfectly modeled: the model may sound as a piano, but will be always different from that piano where its parameters come from. The quality and the computational load is varied by, e.g., changing the accuracy of modeling losses and dispersion, rather than changing the number of simulated partials, which is less noticeable for the

Method	Signal modeling	Physical modeling
Assumptions on the structure	Poor	Yes
Generality	Yes	No
Parameter estimation	Simple	Complicated
Nature of parameters	Abstract	Meaningful
Modeling a specific sound	Precisely	Approximately
Interaction of the musician	Hard to model	Modeled
Interaction of instrument parts	Hard to model	Modeled

Table 1: Main features of the synthesis methods described in the paper.

listener.

4.2.2 The Instruments

The choice between the two approaches strongly depends on which instrument should be modeled. The features which are relevant from this viewpoint for the instruments covered in this paper are listed in Table 2. Naturally, other factors also influence the choice of the user, e.g., if automatic parameter estimation is required, the signal modeling approach should be chosen.

The sound of a specific organ pipe cannot be influenced by the player. Moreover, the coupling between the different pipes is negligible, therefore the different tones can be synthesized independently. As signal modeling models a specific sound almost perfectly, it is the best choice for organ synthesis. Its computational load is acceptable, since the number of partials is low in the case of the organ flue pipes.

In the case of the piano, the player can vary only one parameter for a given note, by changing the impact velocity of the hammer, thus, the timbre space of one note is one-dimensional. For a signal model, this would mean storing different parameter sets for a few hammer velocities, and interpolation could be used between sets. Although it is also possible with the signal model, the effect of the player is much easier modeled by the physics-based approach. Moreover, the strings of the piano are coupled when the damper pedal is depressed which is also controlled by the player: this can be modeled by the physics-based approach only. As the low piano tones may contain about hundred partials, the signal based model would be computationally more demanding than the physical model based on digital waveguides.

For the violin, the freedom of the player is enormous: he can vary the bow force, velocity, position, and angle, the finger position and pressure, and decide on which string he plays the given note. Therefore, the timbre

Instrument	Organ	Piano	Violin
Number of partials	< 20	5-100	10–50
Number of playing parameters	0	Few	Many
Coupling between the instrument parts	Negligible	Present	Significant

Table 2: Main features of the different instruments, serving a base for choosing the proper synthesis approach.

space of the violin is multi-dimensional: for signal-based synthesis many sounds along all these dimensions should be recorded and analyzed, which is an impossible task. Since the goal is not only to render the sound of a specific violin note, but to create a playable instrument, the only choice which remains is physical modeling. The inputs of the physical model are the real physical parameters (e.g., bow force and velocity), therefore the effect of the player is automatically taken into account.

5 Conclusion

In this paper signal-model and physical-model based sound synthesis methods have been described, namely additive synthesis with envelope-filters and digital waveguide modeling. Three case studies (applications of the methods to the sound synthesis of the organ, the piano and the violin) have been introduced, and detailed analysis of the effectiveness of the different synthesis methods have been discussed.

The proposed additive synthesis method is capable of the accurate reproduction of a specific sound of an instrument, but primarily the sound from which its parameters are derived from. The model can be made more realistic by analyzing more then one waveforms and including the systematic variations of the parameters (e.g. parameter values as functions of the fundamental frequency) and their random variations (such as change of the parameters of the attack transient filter, or the random variation of the noise spectrum of an organ flue pipe). With the analysis of these variations, the signal model is able to behave as a real instrument. However, as the parameters of the model are not correlated directly with those of the instrument, the control of the instrument is not an easy task.

As the physical model is based on the physics of real instruments, its transient and random behavior is close to those of the instrument to be modeled. In addition, its parameters are derived directly from those of the instrument (such as string length, bow velocity), thus, controlling a physics-based instrument is a much easier task. In this paper, computationally efficient physical modeling based methods were presented. It was shown that the models need to be evaluated also from a perceptual point of view and this way the trade-off between efficiency and high fidelity can be controlled.

As a theoretical result, it was shown that the signal model and the physical model can be equivalent under specific circumstances. Finally, it was proven that all methods can be used for realistic instrument modeling, but their computational efficiency varies as the function of the instrument to be modeled.

Acknowledgement

This work was supported by the Hungarian National Scientific Research Fund OTKA under contract number F035060.

References

[1] Julius O. Smith, "Viewpoints on the history of digital synthesis," in *Proc. Int. Computer Music Conf.*, Montreal, Canada, September 1991, pp. 1–10.

[2] Tero Tolonen, Vesa Välimäki, and Matti Karjalainen, "Evaluation of modern sound synthesis methods," Tech. Rep. 48, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland, March 1998, URL: `http://www.acoustics.hut.fi/~ttolonen/sound_synth_report.html`.

[3] John M. Chowning, "The synthesis of complex audio spectra by means of frequency modulation," *J. Aud. Eng. Soc.*, vol. 21, no. 7, pp. 526–534, 1973.

[4] Marc Le Brun, "Digital waveshaping synthesis," *J. Aud. Eng. Soc.*, vol. 27, no. 4, pp. 250–266, April 1979.

[5] Daniel Arfib, "Digital synthesis of complex spectra by means of multiplication of nonlinear distorted sine waves," *J. Aud. Eng. Soc.*, vol. 27, no. 10, pp. 757–768, 1979.

[6] Curtis Roads, *The Computer Music Tutorial*, The MIT Press, Cambridge, Massachusetts, USA, 1995.

[7] Xavier Serra and Julius O. Smith, "Spectral modeling synthesis: a sound analysis/synthesis system based on deterministic plus stochastic decomposition," *Computer Music J.*, vol. 14, no. 4, pp. 12–24, Winter 1990.

[8] Julius O. Smith, "Physical modeling using digital waveguides," *Computer Music J.*, vol. 16, no. 4, pp. 74–91, Winter 1992.

[9] Neville H. Fletcher and Thomas D. Rossing, *The Physics of Musical Instruments*, Springer-Verlag, New York, USA, 1998.

[10] Judit Angster, *Modern methods and their results in measurement of the sound and vibration of the organ flue pipes*, Ph.D. thesis, MTA MMSZ Acoustic Research Lab, Budapest, 1990, In Hungarian.

[11] M. Piszczalski et al., "Performed music: Analysis, synthesis and display by computer," *Journal of the Audio Engineering Society of America*, vol. 29, no. 1/2, pp. 38–55, February 1981.

[12] J. Meyer, "Temporal fine structure of organ sounds in churches," in *138th Meeting of the Acoustical Society of America*, Columbus, Ohio, USA, 1–5 November 1999, pp. —.

[13] J. Angster, J. Angster, and A. Miklós, "Coupling between simultaneously sounded organ pipes," in *94th AES Convention (Preprint 3534)*, Berlin, Germany, 16–19 March 1993, pp. 1–8.

[14] H. Klotz, *The Organ*, Zeneműkiadó, Budapest, Hungary, 1972.

[15] G. Péceli, "A common structure for recursive discrete transforms," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 10, pp. 1035–36, October 1986.

[16] János Márkus, "Signal model based synthesis of the sound of organ pipes," M.S. thesis, BUTE, DMIS, Budapest, Hungary, June 1999, 120 p., in Hungarian.

[17] S. K. Mitra and J. F. Kaiser, Eds., *Handbook for Digital Signal Processing*, John Wiley & Sons, Inc., New York, 1993.

[18] C. S. Burrus and T. W. Parks, Eds., *Digital Filter Design*, John Wiley & Sons, Inc., New York, 1988.

[19] J. Márkus, "Signal model based synthesis of the sound of organ pipes," Tech. Rep., Budapest University of Technology and Economics, Budapest, Hungary, 2000, URL: http://www.mit.bme.hu/˜markus/projects/organ.

[20] P. M. Morse, *Vibration and Sound*, McGraw-Hill, 1948, Reprint, (1st ed. 1936).

[21] Lejaren Hiller and Pierre Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part 1," *J. Aud. Eng. Soc.*, vol. 19, no. 6, pp. 462–470, June 1971.

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

[23] Julius O. Smith, *Techniques for Digital Filter Design and System Identification with Application to the Violin*, Ph.D. thesis, Stanford University, California, USA, June 1983.

[24] Timo I. Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K. Laine, "Splitting the unit delay—tools for fractional delay filter design," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30–60, January 1996.

[25] Vesa Välimäki, Jyri Huopaniemi, Matti Karjalainen, and Zoltán Jánosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *J. Aud. Eng. Soc.*, vol. 44, no. 5, pp. 331–353, May 1996.

[26] Vesa Välimäki and Tero Tolonen, "Development and calibration of a guitar synthesizer," *J. Aud. Eng. Soc.*, vol. 46, no. 9, pp. 766–778, September 1998.

[27] Balázs Bank, "Physics-based sound synthesis of the piano," M.S. thesis, Budapest University of Techology and Economics, Hungary, May 2000, Published as Report 54 of HUT Laboratory of Acoustics and Audio Signal Processing, URL: `http://www.mit.bme.hu/∼bank`.

[28] C. Erkut, "Loop filter design techniques for virtual string instruments," in *Int. Symp. on Musical Acoustics (ISMA'01)*, Perugia, Sept. 2001, pp. 259–262.

[29] Balázs Bank and Vesa Välimäki, "Robust loss filter design for digital waveguide synthesis of string tones," *IEEE Signal Processing Letters*, vol. 10, no. 1, pp. 18–20, Jan. 2003.

[30] Alan V. Oppenheim and Ronald W. Schafer, *Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, USA, 1975.

[31] Davide Rocchesso and Francesco Scalcon, "Accurate dispersion simulation for piano strings," in *Proc. Nordic Acoust. Meeting*, Helsinki, Finland, 1996, pp. 407–414.

[32] M. Lang and T. I. Laakso, "Simple and robust method for the design of allpass filters using least-squares phase error criterion," *IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing*, vol. 41, no. 1, pp. 40–48, January 1994.

[33] Gabriel Weinreich, "Coupled piano strings," *J. Acoust. Soc. Am.*, vol. 62, no. 6, pp. 1474–1484, December 1977.

[34] Mitsuko Aramaki, Julien Bensa, Laurent Daudet, Philippe Guillemain, and Richard Kronland-Martinet, "Resynthesis of coupled piano string vibrations based on physical modeling," *Journal of New Music Research*, vol. 30, no. 3, pp. 213–226, 2001.

[35] Balázs Bank, "Accurate and efficient modeling of beating and two-stage decay for string instrument synthesis," in *Proc. MOSART Workshop on Curr. Res. Dir. in Computer Music*, Barcelona, Spain, November 2001, pp. 134–137.

[36] Vesa Välimäki, Matti Karjalainen, and T. Laakso, "Modeling of woodwind bores with finger holes," in *Proc. of the International Computer Music Conference (ICMC'93)*, Tokyo, Japan, 10–15 September 1993, pp. 32–39.

[37] Matti Karjalainen, Vesa Välimäki, Heikki Räisänen, and Harri Saastamoinen, "DSP equalization of electret film pickup for acoustic guitar," in *Proc. 106th AES Conv., Preprint No. 4907*, Munich, Germany, 1999.

[38] Julius O. Smith, "Efficient synthesis of stringed musical instruments," in *Proc. Int. Computer Music Conf.*, Tokyo, Japan, September 1993, pp. 64–71.

[39] Jean-Marc Jot, "An analysis/synthesis approach to real-time artificial reverberation," in *Proc. IEEE Int. Conf. on Acoust., Speech, and Sign. Proc.*, San Francisco, California, USA, 1992, vol. 2, pp. 221–224.

[40] Balázs Bank, Giovanni De Poli, and László Sujbert, “A multi-rate approach to instrument body modeling for real time sound synthesis applications,” in *Proc. 112*th *AES Conv.*, Munich, Germany, May 2002, Preprint No. 5526.

[41] Xavier Boutillon, “Model for piano hammers: Experimental determination and digital simulation,” *J. Acoust. Soc. Am.*, vol. 83, no. 2, pp. 746–754, February 1988.

[42] Antoine Chaigne and Anders Askenfelt, “Numerical simulations of piano strings. i. a physical model for a struck string using finite difference methods,” *J. Acoust. Soc. Am.*, vol. 95, no. 2, pp. 1112–1118, February 1994.

[43] Gianpaolo Borin, Giovanni De Poli, and Davide Rocchesso, “Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems,” *IEEE Trans. on Speech and Aud. Proc.*, vol. 8, no. 5, pp. 597–606, September 2000.

[44] Balázs Bank, “Nonlinear interaction in the digital waveguide with the application to piano sound synthesis,” in *Proc. Int. Computer Music Conf.*, Berlin, Germany, 2000, pp. 54–58.

[45] Julius O. Smith, “Efficient simulation of the reed-bore and bow-string mechanisms,” in *Proc. of the International Computer Music Conference (ICMC’86)*, Den Haag, Netherlands, October 1986, pp. 275–280.

[46] Stefania Serafin, Christophe Vergez, and Xavier Rodet, “Friction and application to real-time physical modeling of a violin,” in *Proc. of the International Computer Music Conference (ICMC’99)*, Bejing, China, 22–28 October 1999, pp. 216–219.

[47] Tapio Takala, Jarmo Hiipakka, Mikael Laurson, and Vesa Välimäki, “An expressie synthesis model for bowed string instruments,” in *Proc. of the International Computer Music Conference (ICMC’2000)*, Berlin, Germany, 27 August – 1 September 2000, pp. 70–73.

Contents