

# MODEL-BASED SOUND SYNTHESIS OF THE HARPSICHORD<sup>†</sup>

*Vesa Välimäki<sup>1</sup>, Henri Penttinen<sup>1</sup>, Mikael Laurson<sup>2</sup>, Cumhuri Erku<sup>1</sup>, and Jonte Knif<sup>2</sup>*

<sup>1</sup>Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing,  
P.O. Box 3000, FIN-02015 HUT, Espoo, Finland

<sup>2</sup>Sibelius Academy, Centre for Music and Technology,  
P.O. Box 86, FIN-00251, Helsinki, Finland  
vesa.valimaki@hut.fi

## ABSTRACT

We have developed a sound synthesis algorithm for the harpsichord by applying the principles of digital waveguide modeling. A short overview on the acoustics of the harpsichord is given, and signal-processing techniques for synthesizing harpsichord tones are suggested. In particular, a new loop filter for the digital waveguide model is introduced and analyzed. The implementation of the synthesizer using a block-based graphical programming language is briefly described. Parameter values estimated from recorded data are given. Sound examples are available at <http://www.acoustics.hut.fi/publications/papers/jasp-harpsy/>.

## 1. INTRODUCTION

Sound synthesis is particularly interesting for acoustic keyboard instruments, since they are usually expensive and large and may require amplification during performances. Electronic versions of these instruments benefit from the fact that keyboard controllers using MIDI are commonly available. Digital pianos imitating the timbre and features of grand pianos are among the most popular electronic instruments. This work focuses on the imitation of the harpsichord, which is expensive and relatively rare, but is still commonly used in Renaissance and baroque music. Figure 1 shows the instrument used in this study. It is a two-manual harpsichord that contains three individual sets of strings, two bridges, and has a large soundboard.

Instead of wavetable and sampling techniques, that are popular in digital instruments, we apply modeling techniques to design an electronic instrument that sounds nearly identical to its acoustic counterpart and faithfully responds to the player's actions, just as an acoustic instrument. We use the modeling principle called commuted waveguide synthesis [2], [3], [4], but have modified it, because we use a digital filter to model the soundboard response. Commuted synthesis uses the basic property of linear systems, that in a cascade of transfer functions their ordering can be changed without affecting the overall transfer function. This way, the complications in the modeling of the soundboard resonances extracted from a recorded tone can be hidden in the input sequence. In the original form of commuted synthesis, the input signal contains the contribution of the excitation mechanism and that of the soundboard with all its vibrating modes [5]. In the current implementation, the input samples are short (less than half a second) and only contain the initial part of the soundboard response; the tail of the soundboard response is reproduced with a reverberation algorithm.

Digital waveguide modeling [6] appears to be an excellent tool for the synthesis of harpsichord tones. A strong argument supporting this view is that tones generated using the basic Karplus–Strong algorithm [7] are reminiscent of the harpsichord for many listeners. This synthesis technique has been shown to be a simplified version of a waveguide string model [6], [8]. However, this does not imply that realistic harpsichord synthesis is easy. A detailed imitation of the properties of a fine instrument is challenging. Careful modifications to the algorithm and proper signal analysis and calibration routines are needed for a natural-sounding synthesis.

---

<sup>†</sup> A long version of this paper will be published in the special issue on model-based sound synthesis in EURASIP Journal on Applied Signal Processing [1].



Figure 1. The harpsichord used in this study has two keyboards, three registers (string sets) and two bridges. This picture was taken during the tuning of the instrument in the large anechoic chamber of Helsinki University of Technology.

## 2. CONSTRUCTION AND OPERATING PRINCIPLE OF THE HARPSICHORD

The harpsichord is a stringed keyboard instrument with a long history dating back to at least the year 1440 [9]. It is the predecessor of the pianoforte and the modern piano. It belongs to the group of plucked string instruments due to its excitation mechanism. In this section, we describe briefly the construction and the operating principles of the harpsichord and give details of the instrument used in this study. For a more in-depth discussion and description of the harpsichord, see e.g., [10], [11], [12], and [13].

### 2.1. Construction of the instrument

The form of the instrument can be roughly described as triangular, and typically the oblique side is curved. A harpsichord has one or two manuals that control two to four sets of strings, also called registers or string choirs. Two of the string choirs are typically tuned in *unison*. These are called the 8' (8 foot) registers. Often the third string choir is tuned an octave higher, and it is called the 4' register. The manuals can be set to control different registers, usually with a limited number of combinations. This permits the player to use different registers with left and right-hand manuals and therefore vary the timbre and loudness of the instrument. The 8' registers differ from each other in the plucking point of the strings. Hence, the 8' registers are called 8' back and front registers, where 'back' refers to the plucking point away from the nut (and the player).

The keyboard of the harpsichord typically spans four or five octaves, which became a common standard in the early 18<sup>th</sup> century. One end of the strings is attached to the nut and the other to a long, curved bridge. The

portion of the string behind the bridge is attached to a hitch pin, which is on top of the soundboard. This portion of the string also tends to vibrate for a long while after a key press, and it gives the instrument a reverberant feel. The nut is set on a very rigid wrest plank. The bridge is attached to the soundboard. Therefore, the bridge is mainly responsible for transmitting string vibrations to the soundboard. The soundboard is very thin—about 2 to 4 mm—and it is supported by several ribs installed in patterns that leave trapezoidal areas of the soundboard vibrating freely. The main function of the soundboard is to amplify the weak sound of the vibrating strings, but it also filters the sound. The soundboard forms the top of a closed box, which typically has a rose opening. It causes a Helmholtz resonance, the frequency of which is usually below 100 Hz [13]. In many harpsichords, the soundbox also opens to the manual compartment.

## 2.2. Operating principle

A plectrum—also called a quill—that is anchored onto a jack, plucks the strings. The jack rests on a string, but there is a small piece of felt (called the damper) between them. One end of the wooden keyboard lever is located a small distance below the jack. As the player pushes down a key on the keyboard, the lever moves up. This action lifts the jack up and causes the quill to pluck the string. When the key is released, the jack falls back and the damper comes in contact with the string with the objective to dampen its vibrations. A spring mechanism in the jack guides the plectrum so that the string is not re-plucked when the key is released.

## 2.3. The harpsichord used in this study

The harpsichord used in this study (see Figure 1) was built in 2000 by Jonte Knif and Arno Pelto. It has the characteristics of harpsichords built in Italy and Southern Germany. This harpsichord has two manuals and three sets of string choirs, namely an 8' back, an 8' front, and a 4' register. The instrument was tuned to the Vallotti tuning [14] with the fundamental frequency of  $A_4$  of 415 Hz. The tuning is considerably lower than the current standard (440 Hz or higher). This is typical of old musical instruments. There are 56 keys from  $G_1$  to  $D_6$ , which correspond to fundamental frequencies 46 Hz and 1100 Hz, respectively, in the 8' registers; the 4' register is an octave higher, so the corresponding lowest and highest fundamental frequency are about 93 Hz and 2200 Hz. The instrument is 240 cm long and 85 cm wide, and its strings are all made of brass. The plucking point changes from 12% to about 50% of the string length in the bass and in the treble range, respectively. This produces a round timbre (i.e., weak even harmonics) in the treble range. In addition, the dampers have been left out in the last octave of the 4' register to increase the reverberant feel during playing. The wood material used in the instrument has been heat-treated to artificially accelerate the aging process of the wood.

## 2.4. Inharmonicity

Dispersion is always present in real strings. It is caused by the stiffness of the string material. This property of strings gives rise to inharmonicity in the sound. An offspring of the harpsichord, the piano, is famous for its strongly inharmonic tones, especially in the bass range [15], [10]. The inharmonicity of the recorded harpsichord tones were investigated in order to find out whether it is relevant to model this property. The partials of recorded harpsichord tones were picked semi-automatically from the magnitude spectrum, and with a least-square fit we estimated the inharmonicity coefficient  $B$  [15] for each recorded tone. The measured  $B$  values were compared with the threshold of audibility and its 90% confidence intervals taken from listening test results [16]. It was noticed that the  $B$  coefficient is above the mean threshold of audibility in all cases, but above the frequency 140 Hz the measured values are within the confidence interval. Thus, it is not guaranteed that these cases actually correspond to audible inharmonicity (see Fig. 5 in [1]). At low frequencies, in the case of the 19 lowest keys of the harpsichord, the inharmonicity coefficients are about  $10^{-5}$ , and the inharmonicity is audible according to this comparison. It is thus important to implement the inharmonicity for the lowest two octaves or so, but it may also be necessary to implement the inharmonicity for the rest of the notes.

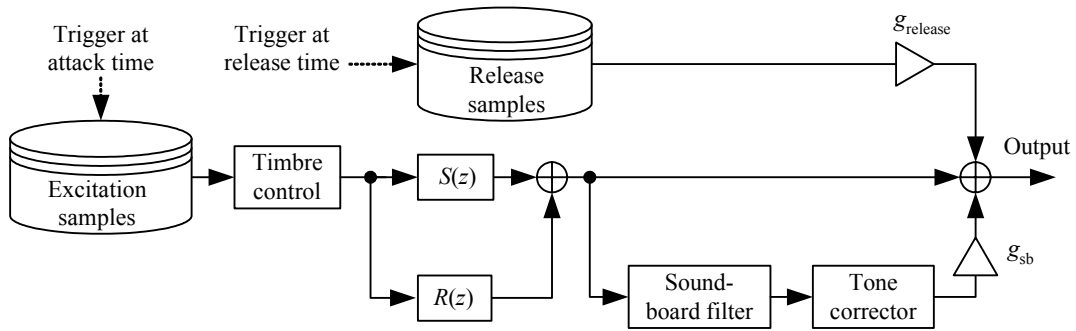


Figure 2. Overall structure of the harpsichord synthesis model for a single string.

### 3. SOUND SYNTHESIS ALGORITHM

We have developed a sound synthesis algorithm for the harpsichord by applying the principles of digital waveguide modeling. Figure 2 shows the block diagram of the synthesis model. The algorithm structure has been modified and extended from previous ones that we have used for model-based synthesis of the acoustic guitar [17] and the clavichord [18]. A version of the commuted waveguide synthesis approach is used, where each tone is generated with a parallel combination of the string model  $S(z)$  and a second-order resonator  $R(z)$  that are excited with a common excitation signal. The second-order resonator  $R(z)$  is tuned to reproduce the beating of one of the partials, as proposed earlier by Bank et al. [19], [20].

We use a version of the vibrating string filter model proposed by Jaffe and Smith [21]. It consists of a feedback loop, where a delay line, a fractional delay filter, a high-order allpass filter, and a loss filter are cascaded, as illustrated in Figure 3. The delay line and the fractional delay filter determine the fundamental frequency of the tone. The high-order allpass filter [21] simulates dispersion which is a typical characteristic of vibrating strings and which introduces inharmonicity in the sound. For the fractional delay filter we use a first-order allpass filter, as originally suggested by Smith and Jaffe [22], [21]. This choice was made because it allows a simple and sufficient approximation of delay when a high sampling rate is used (the sampling rate used in this work is 44100 Hz). Furthermore, there is no need to implement fundamental frequency variations (pitch bend) in harpsichord tones. Thus, the recursive nature of the allpass fractional delay filter, which can cause transients during pitch bends, is not harmful.

#### 3.1. New loop filter structure

The loss filter of waveguide string models is usually implemented as a one-pole filter [23], but now we use an extended version. The transfer function of the new loss filter is

$$H(z) = b \frac{r + z^{-R}}{1 + az^{-1}}, \quad (1)$$

where the scaling parameter  $b$  is defined as

$$b = g(1 + a), \quad (2)$$

$R = \text{round}(r_{\text{rate}}L)$  is the delay-line length of the ripple filter,  $r$  is the ripple depth, and  $a$  is the feedback gain. Parameter  $r_{\text{rate}}$  is the ripple rate parameter that adjusts the ripple density in the frequency domain. The total delay-line length  $L$  in the feedback loop is  $1 + R + L_1$  plus the phase delay caused by the fractional delay filter  $F(z)$  and the allpass filter  $A_d(z)$ . Figure 3 shows the block diagram of the string model with details of the new loss filter, which is seen to be composed of the conventional one-pole filter and a ripple filter in cascade.

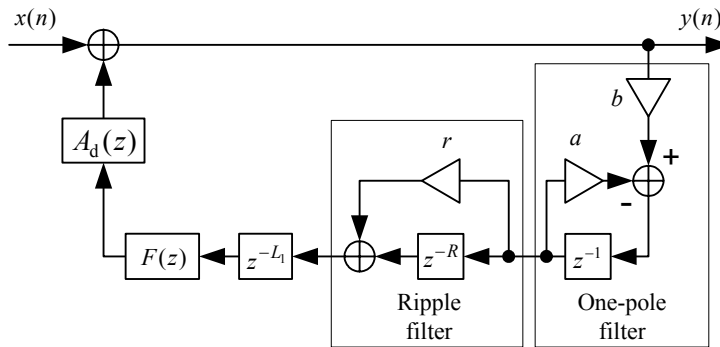


Figure 3. Structure of the proposed string model. The feedback loop contains a one-pole filter, a feed-forward comb filter called 'ripple filter', the rest of the delay line, a fractional delay filter  $F(z)$ , and an allpass filter  $A_d(z)$  simulating dispersion.

### 3.2. Sample databases

The excitation signals of the string models are stored in a database from where they can be retrieved at the onset time. The excitation sequences contain 20,000 samples (0.45 s), and they have been extracted from recorded tones by canceling the partials. The analysis and calibration procedure is discussed further in Section 4 of [1]. The idea is to include in these samples the sound of the quill scraping the string plus the beginning of the attack of the sound so that a natural attack is obtained during synthesis, and the initial levels of partials are set properly. Note that this approach is slightly different from the standard commuted synthesis technique, where the full inverse filtered recorded signal is used to excite the string model [23]. In the latter case, all modes of the soundboard (or soundbox) are contained within the input sequence, and virtually perfect resynthesis is accomplished if the same parameters are used for inverse filtering and synthesis. In the current model, however, we have truncated the excitation signals by windowing them with the right half of a Hanning window. The soundboard response is much longer than that (several seconds), but imitating its ringing tail is taken care of by the soundboard filter (see Section 4).

In addition to the excitation samples, we have extracted short release sounds from recorded tones. One of these is retrieved and played each time a note-off command occurs (see Figure 2). Extracting these samples is easy: once a note is played, the player can wait until the string sound has completely decayed, and then release the key. This way a clean recording of noises related to the release event is obtained without extra processing.

### 3.3. Synthesis parameters

In this section we discuss the parameter values used for synthesis. The parameter values are related to the string model shown in Figure 3, and they have been extracted from recorded tones. A detailed discussion on parameter estimation methods is included in the long version of this paper [1]. Figure 4 shows the  $g$  and  $a$  parameters for all 56 keys of the 8' back register. These values were estimated from recorded tones. The other string models use currently the same parameters, but slightly different delay-lines lengths for detuning. In the near future, we will replace those values with new estimates for each register. The fundamental frequencies of the lowest (#1) and highest (#56) key are 46.9 Hz and 1110 Hz, respectively.

We have previously found that the loop gain parameter has an approximately linear trend as a function of key index (i.e., logarithm of frequency) with some variations and a few outliers [18]. We thus experimented with replacing the estimated  $g$  parameter values with values obtained by linear regression:

$$g \approx 0.9953 + 0.00005559k \quad (3)$$

where  $k = 1 \dots 56$  is the key number starting from  $G_1$ , when  $k = 1$ , and continuing up to  $D_6$  when  $k = 56$ . This straight-line fit is illustrated in Figure 4 (top) with a solid line.

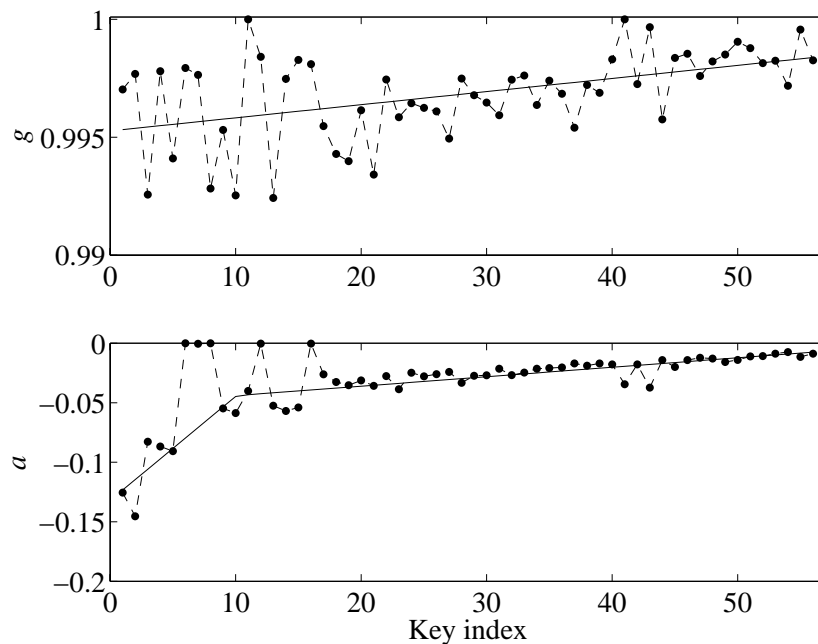


Figure 4. Estimated values of string model parameters  $g$  (top) and  $a$  (bottom) (dots connected with dashed line) together with the straight-line approximations (solid lines) on the key index scale.

The values of parameter  $a$  may be approximated by a low-order polynomial as well. However, a linear regression to the data shown in the lower part of Figure 4 yields intolerably large errors. To reduce the error, we use piecewise linear regression with a knee point between key indices 10 and 11:

$$a \approx \begin{cases} -0.1323 + 0.008276k, & \text{for } k \leq 10 \\ -0.04713 + 0.0006830k, & \text{for } k \geq 11. \end{cases} \quad (4)$$

This two-part straight-line fit is shown in Figure 4 (bottom) with a solid line. These linear approximations of the parameter values introduce minor perceivable changes in timbre, but the overall tone color behaves more uniformly with the linear approximations than with the unprocessed parameter data. We have previously noticed the same phenomenon during the calibration of the clavichord synthesizer [18].

Figure 5 shows the estimated parameters for the ripple filter. The two parameters,  $r$  and  $r_{\text{rate}}$ , have been adjusted so that the decay time of the longest ringing harmonic among the 15 lowest ones is matched exactly. The estimation algorithm is described in detail in [1]. It is seen in Figure 5 (top) that the ripple parameter  $r$  gets small values around zero. Some values are practically zero, such as those for key indices 53 and 54, and it is probable that listeners would not notice the difference even if these values were rounded to zero (i.e., the ripple filter would be replaced with a trivial transfer function with unity gain). Nevertheless, in many other cases the seemingly small parameter values bring about a subtle audible effect.

The estimated  $r_{\text{rate}}$  parameter values vary between a very small positive value and 0.5, as can be seen in Figure 5 (bottom). The latter extreme case causes the even partials of the synthetic tone to decay slower or faster than the odd partials, depending on the value of  $r$ . Values close to zero contribute to a slow oscillation of the magnitude response in the frequency domain. This implies that the magnitude responses and, consequently, the decay rates of neighboring partials will not be much different.

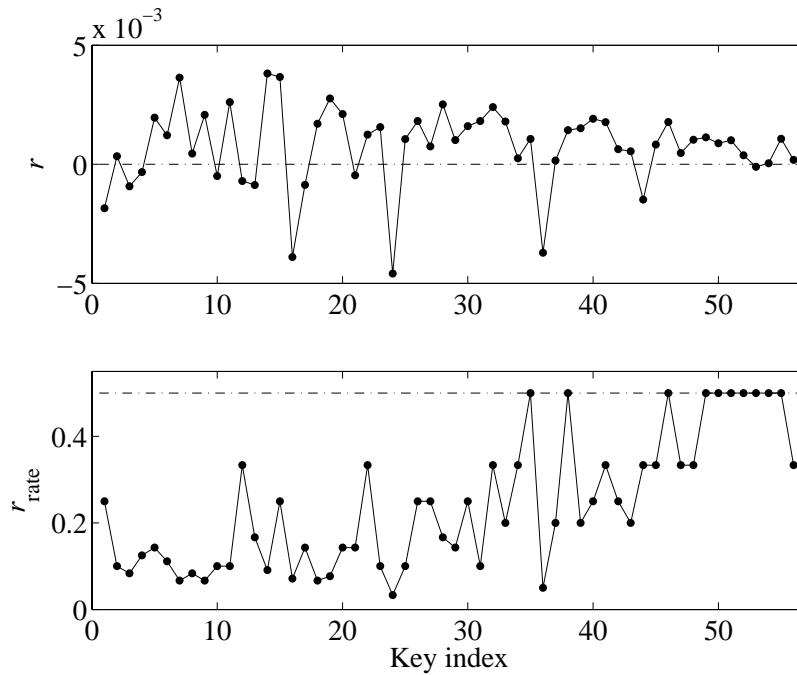


Figure 5. Estimated values of ripple filter parameters  $r$  (top) and  $r_{\text{rate}}$  (bottom).

It is obvious that dramatic timbral effects can only occur when the absolute value of  $r$  and the value of  $r_{\text{rate}}$  are large (i.e., not close to zero) at the same time. Interestingly, it is seen that this rarely happens in the data presented in Figure 5. In the cases where  $r$  is exceptionally large, the corresponding  $r_{\text{rate}}$  value is usually small. For example, at key index 24, we find the largest value for  $r$ ,  $-0.0046$ , but at the same time the  $r_{\text{rate}}$  value is  $0.0333$ , which is the smallest value in this data set. Large negative values of  $r$  are also obtained for key indices 16 and 36, and in these cases we again see that the  $r_{\text{rate}}$  curve has a notch.

#### 4. SOUNDBOARD MODELING

When a note is plucked on the harpsichord, the string vibrations excite the bridge and, consequently, the soundboard. The soundboard has its own modes depending on the size and the materials used. The radiated acoustic response of the harpsichord is reasonably flat over a frequency range from 50 to 2000 Hz [12]. In addition to exciting the air and structural modes of the instrument body, the pluck excites the part of the string that lies behind the bridge, the high modes of the low strings that the dampers cannot perfectly attenuate, and the highest octave of the 4' register strings, which do not have dampers. The resonance strings behind the bridge are about 6 to 20 cm long and have a very inharmonic spectral structure. The soundboard filter used in our harpsichord synthesizer (see Figure 2) is responsible for imitating all these features. However, the lowest body modes can be ignored since they decay fast and are present in the excitation samples. In other words, the modeling is divided into two parts, so that the soundboard filter models the reverberant tail while the attack part is included in the excitation signal, which is fed to the string model. Reference [12] discusses the resonance modes of the harpsichord soundboard in detail.

The radiated acoustic response of the harpsichord was recorded in an anechoic chamber by exciting the bridges (8' and 4') with an impulse hammer at multiple positions. Figure 6 displays a time-frequency response of the 8' bridge when excited between the C<sub>3</sub> strings, i.e., approximately at the middle point of the bridge. The decay times at frequencies below 350 Hz are considerably shorter than in the frequency range from 350 to 1000 Hz. The  $T_{60}$  values at the respective bands are about 0.5 s and 4.5 s. This can be explained by the fact that the short string portions behind the bridge and the undamped strings resonate and decay slowly.

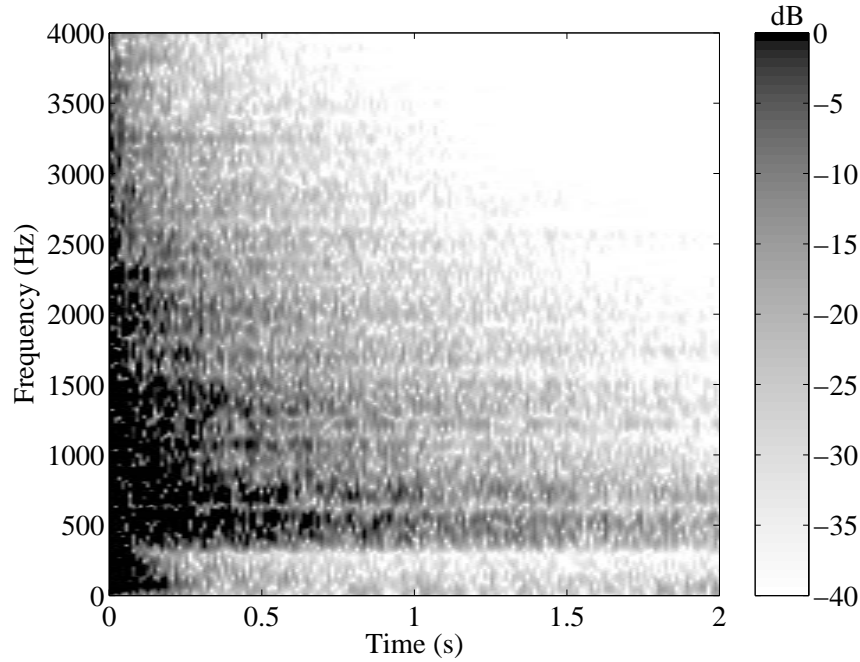


Figure 6. *Time-frequency plot of the harpsichord air radiation when the 8' bridge is excited. To exemplify the fast decay of the low-frequency modes only the first 2 seconds and frequencies up to 4 kHz are displayed.*

As suggested by several authors, see e.g., [19], and [24], the impulse response of a musical instrument body can be modeled with a reverberation algorithm. Such algorithms have been originally devised for imitating the impulse response of concert halls. In a previous work, we triggered a static sample of the body response with every note [18]. In contrast to this sample-based solution, which produces the same response every time, the reverberation algorithm produces additional variation in the sound: as the input signal of the reverberation algorithm is changed, or in this case as the key or register is changed, the temporal and frequency content of the output changes accordingly.

The soundboard response of the harpsichord in this work is modeled with an algorithm presented in [25]. It is a modification of the feedback delay network [26], where the feedback matrix is replaced with a single coefficient and comb allpass filters have been inserted in the delay-line loops. A schematic view of the reverberation algorithm is shown in Figure 7. This structure is used because of its computational efficiency. The  $H_k(z)$  blocks represent the loss filters,  $A_k(z)$  blocks are the comb allpass filters, and the delay lines are of length  $P_k$ . In this work, eight ( $N=8$ ) delay lines are implemented.

One-pole lowpass filters are used as loss filters  $H_k(z)$ , which implement the frequency-dependent decay. The transfer functions of these filters can be written as

$$H_k(z) = g_{h,k} \frac{1 - a_{h,k}}{1 - a_{h,k} z^{-1}}, \quad (5)$$

where  $g_{h,k}$  are the dc gain factors and  $a_{h,k}$  are the feedback coefficients of the loss filters ( $k = 1 \dots 8$ ). The values of these coefficients are listed in Table 1. The comb allpass filters increase the diffusion effect, and they all have the transfer function

$$A_k(z) = \frac{a_{ap,k} + z^{-M_k}}{1 + a_{ap,k} z^{-M_k}}, \quad (6)$$



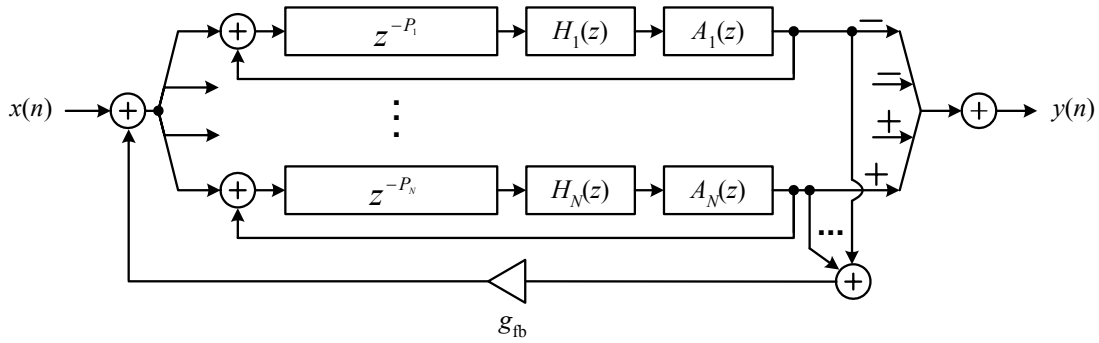


Figure 7. A schematic view of the reverberation algorithm used for soundboard modeling. The feedback gain value of  $g_{fb} = -0.25$  is used in this work. The other parameter values can be found in Table 1 and in Section 4 of this paper.

where  $M_k$  are the delay-line lengths and  $a_{ap,k}$  are the allpass filter coefficients. To ensure stability, it is required that  $a_{ap,k} \in [-1, 1]$ . In this work we use the value  $a_{ap} = 0.5$  for all allpass filters of the reverberation algorithm. We have chosen each of the delay-line lengths  $M_k$  to be 8% of the corresponding delay-line length  $P_k$ . The delay-line lengths used in this work are presented in Table 1.

In addition to the reverberation algorithm, a tone corrector filter, as shown in Figure 2, is used to match the spectral envelope of the target response, i.e., to suppress the low frequencies below 350 Hz and give some additional lowpass characteristics at high frequencies. The tone correction section is divided into two parts: a highpass filter that suppresses frequencies below 350 Hz and another filter that imitates the spectral envelope at the middle and high frequencies. The highpass filter is a 5<sup>th</sup> order Chebyshev type I design with a 5 dB pass-band ripple, the 6 dB point at 350 Hz, and a roll-off rate of about 50 dB per octave below the cutoff frequency. The magnitude response of this IIR highpass filter is shown in Figure 8 for frequencies below 1 kHz.

A 10<sup>th</sup> order spectral envelope filter for the soundboard model was also designed using linear prediction from a 0.2 second long windowed segment of the measured target response (see Figure 6 from 0.3 s to 0.5). However, during the testing of the synthesizer it was noticed that this filter did not improve the sound quality of synthetic tones. Finally, it was decided that the spectral envelope correction filter is not included in the current implementation at all, and thus the tone corrector consists of the highpass filter alone.

Table 1. Parameter values used in the eight branches of the reverberation algorithm of Figure 7. Delays  $P$  and  $M$  are given in samples. Filter parameters  $g_n$  and  $a_n$  refer to the parameters in Eq. (5).

|                  | Delay ( $P$ ) | Loss Filter Gain ( $g_n$ ) | Loss Filter Feedback ( $a_n$ ) | Allpass Filter Delay ( $M$ ) |
|------------------|---------------|----------------------------|--------------------------------|------------------------------|
| <b>Branch #1</b> | 1009          | 0.971945                   | 0.094931                       | 81                           |
| <b>Branch #2</b> | 1109          | 0.969209                   | 0.104272                       | 89                           |
| <b>Branch #3</b> | 1237          | 0.965723                   | 0.116181                       | 99                           |
| <b>Branch #4</b> | 1373          | 0.962024                   | 0.128829                       | 110                          |
| <b>Branch #5</b> | 1489          | 0.958890                   | 0.139552                       | 119                          |
| <b>Branch #6</b> | 1609          | 0.955641                   | 0.150669                       | 129                          |
| <b>Branch #7</b> | 1741          | 0.952105                   | 0.162769                       | 139                          |
| <b>Branch #8</b> | 1999          | 0.945195                   | 0.186397                       | 160                          |

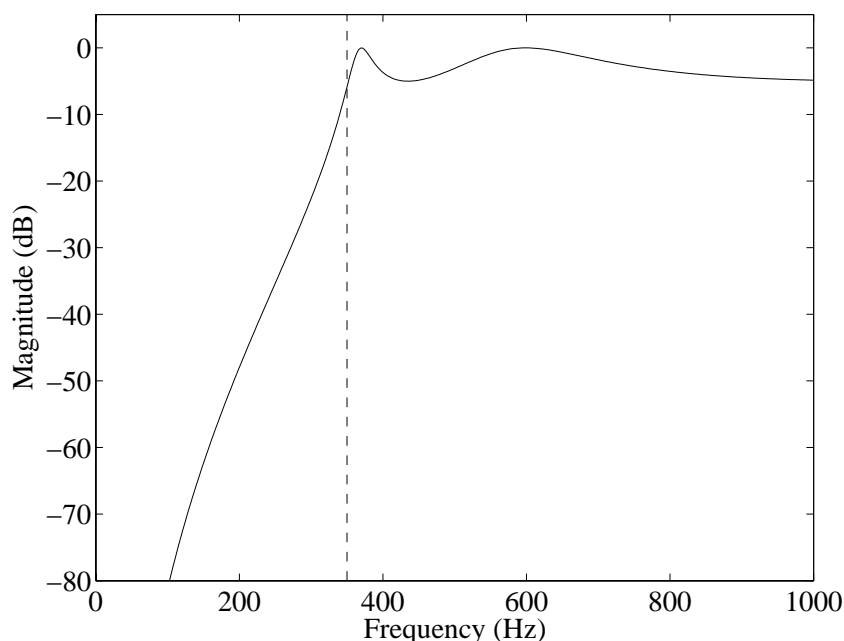


Figure 8. *Magnitude response of the IIR highpass filter used as the tone corrector for the soundboard model. The cutoff frequency ( $-6$  dB point) of 350 Hz is indicated with a vertical dashed line.*

## 5. SOFTWARE IMPLEMENTATION

The synthesis and control part of the harpsichord synthesizer is realized using a visual software synthesis package called PWSynth [27]. PWSynth, in turn, is part of a larger visual programming environment called PWGL [28]. The control information is generated using our music notation package ENP (Expressive Notation Package) [29]. Our previous work in designing computer simulations of musical instruments has resulted in several applications, such as the classical guitar [17], the Renaissance lute, the Turkish ud [30], and the clavichord [18]. The two-manual harpsichord tackled in the current study is the most challenging and complex instrument that we have yet investigated. As this kind of work is experimental, and the synthesis model must be refined by interactive listening, a system is needed that is capable of making fast and efficient prototypes of the basic components of the system. Another non-trivial problem is the parameterization of the harpsichord synthesizer. These problems are solved by using a special loop constructor box, called ‘copy-synth-patch’, that allows the duplication of arbitrary subpatches. The system generates automatically symbolic pathnames to the required control entry points in order to parameterize the synthesizer. These implementation problems are discussed in more detail in [1].

Figure 9 gives an overview of our system. The left part of the figure gives the top-level definition of a harpsichord synthesizer prototype. It consists of a string model abstraction (the box with the label ‘String-model’) that is duplicated  $N$  times where  $N$  is the maximum number of voices needed for the current musical score (‘get-max-voice-count’). The instrument body is simulated with a reverberation algorithm (see Section 4), which consists of a reverb unit abstraction box (‘Reverb-unit’) that is duplicated 8 times. The patch contains also a separate signal path for the excitation signals that allows controlling the attack sharpness [18] separately (see the output of the left-most ‘copy-synth-patch’ box with the label ‘excit’). The amplitudes of these signal paths (i.e., the strings, the body, and the excitation signals) can be controlled in real time with sliders. The right part of Figure 9, in turn, shows a musical score (‘Score-editor’), which has been prepared in advance by the user. In addition to basic musical information, such as pitches and rhythms, it contains special instrument specific expressions that allow the indication of various playing styles and performance practices that are needed to produce musically acceptable performances.

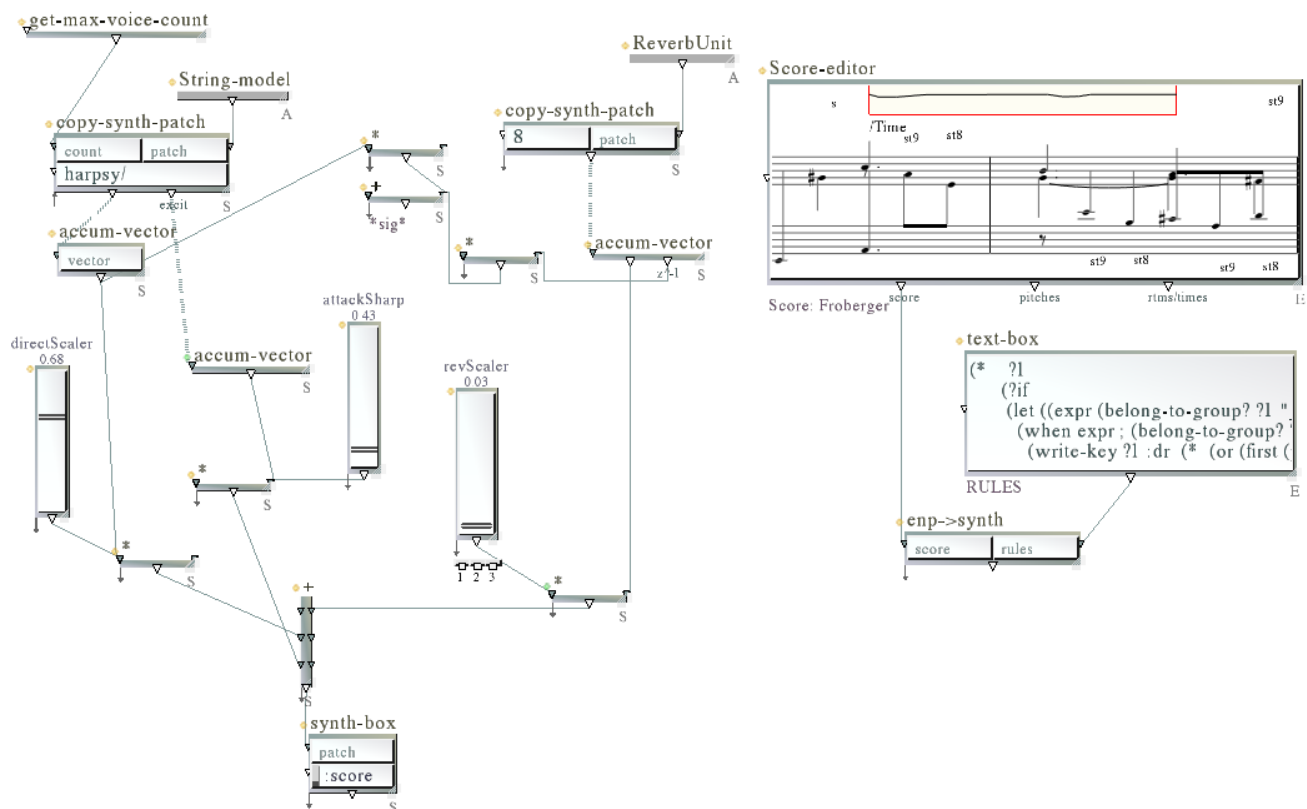


Figure 9. Screenshot of the software implementation of the harpsichord synthesizer in PWSynth.

## 6. CONCLUSIONS

Signal processing techniques for synthesizing harpsichord tones have been discussed in this paper. The new contributions to stringed instrument models include a sparse high-order loop filter and a soundboard model that uses a reverberation algorithm. The sparse loop filter consists of a conventional one-pole filter and a feed-forward comb filter inserted in the feedback loop of a basic string model. The fast-decaying modes of the soundboard are incorporated in the excitation samples of the synthesizer, while the long-ringing modes at the middle and high frequencies are imitated using a reverberation algorithm. The parameterization and use of simple filters also allow manual adjustment of the timbre. The main parameter values that have been estimated from recorded harpsichord tones are given.

A physics-based synthesizer, such as the one described here, has several musical applications, the most obvious one being the usage as a computer-controlled musical instrument. Examples of single tones and musical pieces synthesized with the synthesizer are available at <http://www.acoustics.hut.fi/publications/papers/jasp-harpsy/>.

## 7. ACKNOWLEDGMENTS

The work of Henri Penttinen has been supported by the Pythagoras Graduate School of Sound and Music Research (see <http://www2.siba.fi/pythagoras/>). The work of Cumhur Erkut is part of the EU project ALMA (IST-2001-33059, see <http://www-dsp.elet.polimi.it/alma/>).

## 8. REFERENCES

- [1] V. Välimäki, H. Penttinen, J. Knif, M. Laurson, and C. Erkut, "Sound synthesis of the harpsichord using a computationally efficient physical model," to be published in *EURASIP J. Applied Signal Processing* (special issue on Model-Based Sound Synthesis), vol. 4, no. 7, 2004.
- [2] J. O. Smith, "Efficient synthesis of stringed musical instruments," in *Proc. Int. Computer Music Conf.*, pp. 64–71, Tokyo, Japan, Sept. 1993.
- [3] M. Karjalainen and V. Välimäki, "Model-based analysis/synthesis of the acoustic guitar," in *Proc. 1993 Stockholm Music Acoustics Conf.*, pp. 443–447, Stockholm, Sweden, Aug. 1993.
- [4] M. Karjalainen, V. Välimäki, and Z. Jánosy, "Towards high-quality sound synthesis of the guitar and string instruments," in *Proc. Int. Computer Music Conf.*, pp. 56–63, Tokyo, Japan, Sept. 1993.
- [5] J. O. Smith and S. A. Van Duyne, "Commutated piano synthesis," in *Proc. Int. Computer Music Conf.*, pp. 319–326, Banff, Canada, Sept. 1995. Available at <http://www-ccrma.stanford.edu/~jos/pdf/svd95.pdf>.
- [6] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music J.*, vol. 16, no. 4, pp. 74–91, 1992. Available at <http://www-ccrma.stanford.edu/~jos/pmudw/>.
- [7] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music J.*, vol. 7, no. 2, pp. 43–55, 1983.
- [8] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: From the Karplus–Strong algorithm to digital waveguides and beyond," *Computer Music J.*, vol. 22, no. 3, pp. 17–32, 1998. Available at <http://www.acoustics.hut.fi/publications/>.
- [9] F. Hubbard, *Three Centuries of Harpsichord Making*. Harvard University Press, Cambridge, MA, 1965.
- [10] N. H. Fletcher and T. D. Rossing, *Physics of Musical Instruments*. Springer-Verlag, 1991.
- [11] E. L. Kottick, K. D. Marshall, and T. J. Hendrickson, "The acoustics of the harpsichord," *Scientific American*, vol. 264, pp. 94–99, Feb. 1991.
- [12] W. R. Savage, E. L. Kottick, T. J. Hendrickson, and K. D. Marshall, "Air and structural modes of a harpsichord," *J. Acoust. Soc. Am.*, vol. 91, pp. 2180–2189, Apr. 1992.
- [13] N. H. Fletcher, "Analysis of the design and performance of harpsichords," *Acustica*, vol. 37, pp. 139–147, 1977.
- [14] J. Sankey and W. A. Sethares, "A consonance-based approach to the harpsichord tuning of Domenico Scarlatti," *J. Acoust. Soc. Am.*, vol. 101, pp. 2332–2337, Apr. 1997.
- [15] H. Fletcher, E. D. Blackham, and R. Stratton, "Quality of piano tones," *J. Acoust. Soc. Am.*, vol. 34, no. 6, pp. 749–761, 1962.
- [16] H. Järveläinen, V. Välimäki, and M. Karjalainen, "Audibility of the timbral effects of inharmonicity in stringed instrument tones," *Acoustics Research Letters Online*, vol. 2, no. 3, pp. 79–84, 2001. Published on-line at <http://ojps.aip.org/ARLO/>.
- [17] M. Laurson, C. Erkut, V. Välimäki, and M. Kuuskankare, "Methods for modeling realistic playing in acoustic guitar synthesis," *Computer Music J.*, vol. 25, no. 3, pp. 38–49, 2001. Available at <http://lib.hut.fi/Diss/2002/isbn9512261901/>.
- [18] V. Välimäki, M. Laurson, and C. Erkut, "Commutated waveguide synthesis of the clavichord," *Computer Music J.*, vol. 27, no. 1, pp. 71–82, 2003. Available at <http://mitpress.mit.edu/CMJ> (see Sample Articles).
- [19] B. Bank, *Physics-Based Sound Synthesis of the Piano*. MSc thesis, Report no. 54, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Espoo, Finland, 2000. Available at <http://www.acoustics.hut.fi/publications/>.
- [20] B. Bank, V. Välimäki, L. Sujbert, and M. Karjalainen, "Efficient physics-based sound synthesis of the piano using DSP methods," in *Proc. European Signal Processing Conf. (EUSIPCO 2000)*, vol. 4, pp. 2225–2228, Tampere, Finland, Sept. 2000.
- [21] D. A. Jaffe and J. O. Smith, "Extensions of the Karplus–Strong plucked-string algorithm," *Computer Music J.*, vol. 7, no. 2, pp. 76–87, 1983.
- [22] J. O. Smith, *Techniques for Digital Filter Design and System Identification with Application to the Violin*. PhD thesis, Stanford Univ., Stanford, CA, 1983.

- [23] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy, “Physical modeling of plucked string instruments with application to real-time sound synthesis,” *J. Audio Eng. Soc.*, vol. 44, no. 5, pp. 331–353, 1996.
- [24] H. Penttinen, M. Karjalainen, T. Paatero, and H. Järveläinen, “New techniques to model reverberant instrument body responses,” in *Proc. Int. Computer Music Conf.*, pp. 182–185, Havana, Cuba, Sept. 2001.
- [25] R. Väänänen, V. Välimäki, J. Huopaniemi, and M. Karjalainen, “Efficient and parametric reverberator for room acoustics modeling,” in *Proc. Int. Computer Music Conf.*, pp. 200–203, Thessaloniki, Greece, Sept. 1997.
- [26] J. M. Jot and A. Chaigne, “Digital delay networks for designing artificial reverberators,” presented at the AES 90th Convention, Preprint no. 3030, Paris, France, 1991.
- [27] M. Laurson and M. Kuuskankare, “PWSynth: A Lisp-based bridge between computer assisted composition and sound synthesis,” in *Proc. Int. Computer Music Conf.*, pp. 127–130, Havana, Cuba, Sept. 2001.
- [28] M. Laurson and M. Kuuskankare, “PWGL: A novel visual language based on Common Lisp, CLOS and OpenGL,” in *Proc. Int. Computer Music Conf.*, pp. 142–145, Gothenburg, Sweden, Sept. 2002.
- [29] M. Kuuskankare and M. Laurson, “ENP2.0: A music notation program implemented in Common Lisp and OpenGL,” in *Proc. Int. Computer Music Conf.*, pp. 463–466, Gothenburg, Sweden, Sept. 2002.
- [30] C. Erkut, M. Laurson, V. Välimäki, and M. Kuuskankare, “Model-based synthesis of the ud and the Renaissance lute,” in *Proc. Int. Computer Music Conf.*, pp. 119–122, Havana, Cuba, Sept. 2001. Available at <http://lib.hut.fi/Diss/2002/isbn9512261901/>.