

Vesa Välimäki, Jyri Huopaniemi*, Matti Karjalainen** and Zoltán Jánosy***,*

**Helsinki University of Technology, Finland, , **Stanford University, U.S.A.,*

****Technical University of Budapest, Hungary.*

**Presented at
the 98th Convention
1995 February 25 - 28
Paris**



AES

This preprint has been reproduced from the author's advance manuscript, without editing, corrections or consideration by the Review Board. The AES takes no responsibility for the contents.

Additional preprints may be obtained by sending request and remittance to the Audio Engineering Society, 60 East 42nd St., New York, New York 10165-2520, USA.

All rights reserved. Reproduction of this preprint, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

AN AUDIO ENGINEERING SOCIETY PREPRINT



Physical Modeling of Plucked String Instruments with Application to Real-Time Sound Synthesis

Vesa Välimäki^{1,2}, Jyri Huopaniemi¹, Matti Karjalainen^{1,3}, and Zoltán Jánosy⁴

¹*Helsinki University of Technology, Laboratory of Acoustics and Audio
Signal Processing, Otakaari 5A, FIN-02150 Espoo, Finland*

²*CARTES, Ahertajankuja 4, FIN-02100 Espoo, Finland*

³*Stanford University, CCRMA, Stanford, CA 94305, USA*

⁴*Technical University of Budapest, Department of Telecommunications,
Sztoczek u. 2, H-1111 Budapest, Hungary*

Vesa.Valimaki@hut.fi, Jyri.Huopaniemi@hut.fi,
Matti.Karjalainen@hut.fi, Janosy@tsys.hit.bme.hu

ABSTRACT

Physical modeling is a modern approach to musical sound synthesis. In physical models the sound production principle of a musical instrument is explicitly simulated to produce acoustic signals. We have developed efficient DSP algorithms for real-time synthesis of plucked string instruments, such as the acoustic guitar, the mandolin, and the kantele—a traditional Finnish zither. We present signal analysis results of these instruments and show that the values of the model parameters can be reliably estimated based on the analysis of acoustic signals. We describe how nonlinear effects due to string terminations can be incorporated into a discrete-time model and discuss the implementation of a string instrument model using multirate techniques. Results of model-based resynthesis are illustrated to demonstrate that high-quality synthetic sounds can be generated using the proposed modeling principles. Real-time implementation using a signal processor is described and several aspects of controlling physical models of plucked string instruments are studied. Our presentation includes audio examples.

0 INTRODUCTION

Physical modeling of musical instruments has become an increasingly active field of research in musical acoustics and computer music. The term physical modeling refers to mathematical or computational simulation of sound production principles of musical instruments.

Traditionally, musical sound synthesis, such as FM synthesis or waveshaping, has tried to achieve a desired waveform or spectrum [1], [2]. In sampling or wavetable synthesis, digital recordings of acoustic sounds are edited and processed for resynthesis. Although sampled tones may sound perfect, convincing synthesis of musical instruments has been difficult since one sample corresponds to a sound played by one instrument in a certain way. A small change in the playing technique would demand the synthetic sound to be resampled.

The approach taken in physical modeling can be divided into two classes:

- 1) mathematical modeling of physical principles
- 2) design of model-based sound synthesizers

The former approach is typical to physicists who aim at understanding the acoustics of musical instruments. The latter approach—which we have taken—is more practical, but comprehensive understanding of the physics of musical instruments to be synthesized is still needed.

Three different viewpoints are of paramount importance when designing a physically based synthesis technique. First, essential features of the physics of a musical instrument have to be studied carefully. Second, the properties of the human auditory system need to be considered because it is the ear that finally judges whether the synthetic sound is satisfactory or not. In practice a physical model can be constructed by simplifying the underlying physical principles. Here the knowledge on the hearing ability can be of help: those features that are not perceptually relevant need not be simulated precisely, which leads to simplification. The third point of view is that of a DSP engineer. The model should be computable in real time, preferably on a commercial (signal) processor. Thus the model should be easily and efficiently programmable.

In this paper, we show how synthesis models for plucked string instruments, such as guitars, the banjo, the mandolin, and the kantele, can be constructed following the above principles. In our approach, the model for a single string is based on the Karplus–Strong (KS) algorithm [3] which is a simple computational technique for synthesizing plucked string sounds. As reported by Jaffe and Smith [4], this algorithm is a simplification of a *digital waveguide model* for a vibrating lossy string, which is based on the solution of the wave equation (see also [5]–[7]). The improvements in our model over the KS algorithm are

that a continuously variable digital delay line is used for adjusting the length of the string [8] and a more flexible and precise lowpass filter—called a loop filter—is used to bring about the frequency-dependent damping of a string [9]. The model incorporates the influence of the body in an exceptionally efficient way [9]–[11]. Additions to the basic string and body model are discussed in order to include beat effects inherent in string vibration and sympathetic couplings between strings [11].

After the model structure has been designed, an estimation method is needed for calibrating the values of the model parameters. A particular subproblem is the estimation of the coefficients of the loop filter. This problem was first addressed by Smith in the beginning of 1980's [12], [13]. He studied several parametric methods for matching the loop filter and also proposed some non-parametric methods, such as the use of deconvolution. In [14], a time-frequency representation was applied to the analysis of musical sounds and the loop filter was designed based on that data. In [15], the same approach was used in the analysis, but instead of using the KS model, a pair of poles was assigned to each sinusoidal component. The technique was applied to the synthesis of the piano, but was reported to have been tested for guitar tones as well.

In this paper we show that a prototype frequency response for the loop filter can be obtained by means of short-time Fourier analysis and envelope tracking of the harmonics. A first-order all-pole filter is matched to the analysis data using a weighted least squares design. Thereafter the input signal for the model can be extracted from a recorded string instrument sound using inverse filtering. This approach has also been proposed in [14] and [15]. The residual can be truncated or windowed and the resulting short-duration sequence can be fed into the synthesis model to produce an approximation of the original signal. The parameters of the model can be modified and yet a very natural sounding signal will result.

The organization of the paper is as follows. An overview of the plucked string instruments that are studied in this paper is given in Section 1. The principles of modeling string instruments are discussed in Section 2. Two basic models for a vibrating string and signal processing techniques such as Lagrange interpolation are briefly described. The basic models are extended by introducing, e.g., a dual-polarization string model. The modeling of the body of a guitar is considered from three points of view, that of the digital filter approximation, of a principle based on commutativity of linear systems, and a hybrid where some resonances of the body are explicitly modeled and a processed input signal includes the rest of the body response. Section 3 concentrates on the analysis techniques that can be used for estimating the values of the model parameters. Examples of synthesis results are reported in Section 4.

Real-time implementation techniques are considered in Section 5. In Section 6, control aspects of the physical models of plucked string instruments are discussed. Finally, conclusions are drawn and directions for future work are given in Section 7.

1 OVERVIEW OF PLUCKED STRING INSTRUMENTS

The history of guitar-like plucked string instruments spans to antiquity. Cultural and geographical differences have led to evolution of a large family of string instruments over a long period of time. In this paper we have concentrated on the analysis and model-based sound synthesis of six example cases of the plucked string instrument family:

- 1) the modern classical nylon-string acoustic guitar
- 2) the flat-top steel-string acoustic guitar
- 3) the electric guitar
- 4) the banjo
- 5) the mandolin
- 6) the kantele

All these string instruments exhibit some special characteristics which are to be taken into account when designing model-based sound synthesizers. The goal in this research was, however, not only to create sound synthesis methods, but also to use physical modeling as a tool in the research of different string instruments. A short overview of the string instruments featured in this paper is given below. A more detailed analysis of the acoustics of some of these instruments can be found in, e.g., [16].

1.1 The Acoustic Guitar

Analysis and modeling of the classical acoustic guitar has been presented in previous papers [8], [9], [11], [17], but as an extension to guitar modeling we measured and modeled the behavior of the steel-string acoustical guitar. The main difference in the acoustics of these instruments is caused by two facts: 1) the material of the strings, and 2) the plucking method. Steel-string guitars typically have crossed bracing in the soundboard due to the higher tension of the strings [16]. The use of plectrum instead of finger as the excitation method results in a stronger pluck and a brighter tone. These features can be taken into account in a physical model.

1.2 The Electric Guitar

The main difference between acoustic and electric guitars is in the body of the guitar. The solid-body construction as in the case we measured (Fender Stratocaster) radiates very little sound from the body itself or the strings. Our model-based approach to electric guitar does not cover the effects of magnetic pick-ups nor nonlinearly behaving amplifiers. Instead we have focused on the behavior of the string vibration in order to design a physical model for the plain electric guitar.

The three-dimensional plot shown in Fig. 1 exhibits the sound behavior of an electric guitar. The third string was plucked while other strings were damped. The neck pick-up was used for the recording. It can be seen that due to the lack of body resonances the attack part is very well-behaving. The decay of the harmonics is also smooth, i.e., there is no nonlinear behavior. The signal was analyzed using the short-time Fourier transform (STFT) technique, which is covered in Section 3.2.

1.3 The Mandolin

The mandolin is a string instrument used widely in folk and bluegrass music in western countries. It features eight steel strings that are tuned in four pairs. The sound of the mandolin is brighter and decays faster than that of the acoustic guitar. The pairs of equally-tuned strings usually result in beating due to a minor difference in the tuning. Figure 27 depicts the STFT-based spectral analysis of one string of the first string pair of the mandolin plucked with a plectrum near the bridge. The remaining strings, including the other string of the string pair, were carefully damped. The quite rapid decay of harmonics as well as a linear behavior of the strings can be observed.

1.4 The Banjo

The banjo differs from the classical acoustic guitar mainly in the construction of the instrument body. The five strings have been coupled via a metal bridge to a drum-like resonating plate. The time-varying spectrum of a banjo tone is depicted in Fig. 2. The figure shows the response after the first string was plucked with a fingerpick at a distance of 14 cm from the bridge while other strings were damped. The resonances of the body can be observed in the first 200 ms of the tone. The impulse response of the soundboard has been found to be quite long. This must be taken into account in the excitation signal of the synthesis model (see Section 3.5).

1.5 The Kantele

The kantele is an ancient Finnish instrument that features 5-40 strings. Figure 3 shows the 5-string kantele that has been used in the analysis. There are two special characteristics in the kantele that are not found in other string instruments [18]:

- 1) very strong beats in harmonic envelopes and
- 2) prominent even harmonics.

Both effects were found to be due to the unusual way the strings have been terminated. The beats are generated when the horizontally and vertically polarized vibration components are superimposed since there is a clear difference in the effective string length in the two polarizations (see Section 2.6.2 below). This is due to the knot termination around the metal bar (Fig. 4a). The strong second harmonic comes from the longitudinal tension variation that bends the tuning peg (Fig. 4b) and radiates from the soundboard proportionally to the square of the transversal string displacement.

Figure 5 shows a three-dimensional STFT analysis result of a kantele tone. Note that the second harmonic is almost 10 dB louder than the first harmonic in the attack part but it attenuates at a faster rate. Strong beating can be observed in higher harmonics.

1.6 Instrument Measurements

In order to find relevant analysis data for model-based synthesis we conducted accurate and thorough measurements of the chosen string instruments. Professional musicians and high-quality instruments were used. The basic measurements consisted of recording single notes played on each string at several fret positions. Both finger-picking and plectrum-picking were used as the string excitation method for instruments such as the electric guitar and the steel-string guitar. Typical playing styles, such as strumming, vibrato, and glissando, were also recorded to obtain information for the control of the models.

All measurements were carried out in a large anechoic chamber using Brüel&Kjaer microphones and preamplifiers. The measurement data were first stored on a DAT recorder at 44.1 kHz sampling rate and 16 bit quantization giving a theoretical signal-to-noise ratio of about 96 dB. In the second phase, the data were transferred onto a hard disk using the QuickSig software [19] designed at the Laboratory of Acoustics and Audio Signal Processing of the Helsinki University of Technology.

2 GENERAL MODEL FOR A STRING INSTRUMENT

A string instrument can be divided into functional blocks as illustrated in Fig. 6. The sound sources of the instrument are the strings, which have been connected to the body via the bridge and which also interact with each other. The sound radiates mainly through the body. The vibrating strings themselves act as dipole sources and radiate sound inefficiently. First, we concentrate on the strings.

2.1 Basic Model for a Vibrating String

The general solution of the wave equation for a vibrating string is composed of two independent transversal waves traveling in opposite directions (see, e.g., [16]). At the string terminations the waves reflect back with inverted polarity and form standing waves. The losses in the system damp the quasi-periodic vibration of the string. The system is assumed to be linear, and thus all losses and other linear non-idealities may be lumped to the termination and excitation or pick-up points. This is because we are only interested in the output of the system and not in the amplitude values at arbitrary points inside the system. Thus we can combine all the linear operations, i.e., delays and filters, between the input and output point (see, e.g., [6]). The string itself can then be described as an ideal lossless waveguide, which is a *bidirectional discrete-time delay line* [20], [6]. This system may be modeled using a pair of delay lines and a pair of reflection filters $R_b(z)$ and $R_f(z)$ as depicted in Fig. 7. Each of these filters includes the reflection function of the corresponding termination of the string and the dissipative and dispersive contribution due to the string material. A lossless delay line can be implemented computationally efficiently as a circular buffer where the data does not move but only read and write pointers are updated. This may reduce the computational load by several orders of magnitude as compared to a shift register where the data is actually moved.

An efficient structure for a string model is obtained by commuting one of the delay lines with one reflection filter. Then the delay lines may be combined into a double-length line and the filters into a single one that can be expressed as

$$H_l(z) = R_b(z)R_f(z) \quad (1)$$

The transfer function $H_l(z)$ is called the *loop filter*. This formulation is in practice equivalent to the model of Fig. 7 when a comb filter $P(z)$ is cascaded with the string model to cause the filtering effect due to the plucking position [4]. The main difference is the delay between the excitation point and the output but this is insignificant and can be compensated if necessary. This modified

system that we call a *generalized KS model* is illustrated in Fig. 8.

The filter $P(z)$ is called a *plucking-point equalizer* and its transfer function is

$$P(z) = 1 + z^{-M}R_f(z) \quad (2)$$

where $R_f(z)$ is the reflection filter of one end of the string model and M is the delay (in samples) between the excitation point in the upper and lower line of the original waveguide model (see Fig. 7). In practice $R_f(z)$ in (2) need not be modeled very accurately since its magnitude response is only slightly less than unity at all frequencies. In practice the most remarkable difference between the direct and reflected excitation is the inverted phase and thus the filter can be replaced by a constant multiplier $r_f = -1 + \varepsilon$ where ε is a small nonnegative real number. The contribution of a pick-up point in an electric guitar can be incorporated into the model using a similar comb filter [21].

The fact that the finger or plectrum exciting the string has a finite (nonzero) touching width has been ignored in this model, and it has been assumed that the excitation acts at a single point. Widening the excitation point to an interval adds more lowpass filtering to the excitation. We can bring about this effect using a low-order *excitation filter* $E(z)$ that models the dependency of the spectral tilt on the dynamic level of the pluck. Jaffe and Smith [4] suggested that a one-pole digital filter be suitable for this purpose. Furthermore, the complex interaction between the finger and the string has not been modeled. Instead, the excitation has been supposed to be an event that can be modeled using linear operations. In reality, the finger may grab the string for a short while causing nonlinear interaction.

2.2 Length of the String

The effective delay length of the feedback loop in the string model determines the fundamental frequency of the output signal. The delay length (in samples) can be computed as

$$L = \frac{f_s}{f_0} \quad (3)$$

where f_s and f_0 are the sampling rate of the synthesis system and the desired fundamental frequency, respectively. In general $L \in \mathbb{R}$ and thus the implementation of the delay line calls for the use of a *fractional delay filter* that we denote by $F(z)$. It is a phase equalizer that brings about a controllable fractional delay. A first-order allpass filter [4] and a third-order maximally flat FIR fil-

ter based on Lagrange interpolation [8] have been suggested as alternatives for the implementation of the fractional-length delay line in a string model.

We have used Lagrange interpolation for fractional delay approximation. The filter coefficients $h(n)$ of the Lagrange interpolator are computed as (see, e.g., [22])

$$h(n) = \prod_{\substack{k=0 \\ k \neq n}}^N \frac{D-k}{n-k} \quad \text{for } n = 0, 1, 2, \dots, N \quad (4)$$

where N is the order of the FIR filter and D the desired delay. Lagrange interpolation is equivalent to the *maximally flat* approximation (at $\omega = 0$) of band-limited interpolation. The approximation error (and its N derivatives) is zero at $\omega = 0$ and it is negative everywhere else (unless $d = 0$), i.e., the Lagrange interpolator is a lowpass filter. A comparison of the computational efficiency of N th-order maximally flat allpass and FIR fractional delay filters is given in [23]. An extensive review of digital filter approximations for fractional delay has been written by Laakso *et al.* [22].

Real strings are more or less stiff so that the wave propagation is dispersive, i.e., the total loop delay is frequency-dependent. In principle, a single filter may implement both the magnitude and phase in the string model loop, but in practice it is easier to use three filters to control the three parts separately: a linear-phase allpass approximation $F(z)$ to control the fundamental frequency, the loop filter $H_l(z)$ to control the magnitude response, and an allpass filter $D(z)$ to adjust the frequency-dependent overall phase delay. See [24] and [25] for a discussion on implementation of dispersive waveguide models using all-pass filters. For low strings, especially in the piano, dispersivity is important or crucial to the timbre of sound synthesis. For higher strings the dispersivity is relatively small. In plucked string instruments the effect due to dispersion can be neglected and the filter $D(z)$ may be left out of the model.

To conclude our discussion on the frequency-dependence of the string length, let us express $L(\omega)$ by means of the three filters discussed above:

$$L(\omega) = L_l + \tau_H(\omega) + \tau_F(\omega) + \tau_D(\omega) \quad (5)$$

where $L_l \in \mathbb{Z}_+$ is the number of unit delays in the delay line, and $\tau_H(\omega)$, $\tau_F(\omega)$ and $\tau_D(\omega)$ are the phase delays of the loop filter $H_l(z)$, the fractional delay filter $F(z)$, and the dispersion filter $D(z)$, respectively. Figure 9 illustrates the implementation of the string model $S(z)$.

2.3 Loop Filter

In the original KS model, the loop filter $H_l(z)$ is a two-tap averager which is easy to implement without multiplications [3]. This simple filter, however, cannot match the frequency-dependent damping of a physical string. We feel that an FIR filter is not suited to this purpose since its order should be rather high to match the desired characteristics. Instead we suggest the use of an IIR lowpass filter for simulating the damping characteristics of a physical string. From references [16] and [26] it can be concluded that a second or third-order all-pole filter could be suitable for a loop filter. It is important to use the simplest satisfying loop filter since one such filter is needed for the model of each string. Also, the loop filter is not a static filter but its coefficients should be changed as a function of the string length and other playing parameters. This is easiest to achieve using a low-order IIR filter. We have found that a reasonable approximation may be obtained using a first-order all-pole filter

$$H_l(z) = g \frac{1 + |a_1|}{1 + a_1 z^{-1}} \quad (6)$$

where g is the gain of the filter at 0 Hz and a_1 is the filter coefficient that determines the cutoff frequency of the filter. For $H_l(z)$ to be a stable lowpass transfer function, we require that $-1 < a_1 < 0$. The numerator $1 + |a_1|$ scales the frequency response (divided by g) at 0 to unity thus allowing control of the gain at $\omega = 0$ using the coefficient g . We require that $|g| \leq 1$.

Figure 10 shows the magnitude response and group delay of the loop filter $H_l(z)$ (6) with three different values of coefficient a_1 . These values have been chosen so that they represent cases met in practice. In this example $g = 0.99$. (Change of g merely shifts the magnitude response curve vertically.) Note that the group delay of the loop filter is very small in all the three cases.

Figure 11 shows the impulse response of the string model $S(z)$ with the three loop filters of Fig. 10. Here the length of the delay line is $L = 19$. It is seen that the impulse response decays more rapidly as the magnitude of a_1 increases.

2.4 Modeling of the Body

From the viewpoint of signal processing, the body of the acoustic guitar and the transfer function from the bridge to a specific direction may be considered as a high-order linear filter. A set of transfer functions would be needed to simulate the directivity pattern of a string instrument [27]. In most physical models, only one transfer function has been included. This simplified approach simulates the radiation of the sound to one point in front of the sound hole of

the string instrument. This kind of model will thus at best produce a sound reminiscent to a recording of a musical instrument in a relatively anechoic room. In Section 2.5, we discuss different methods to incorporate directivity characteristics into physical models.

The transfer function from the bridge to the listener can be approximately measured by exciting the bridge with an impulse hammer and by registering the radiated sound. Figures 12 and 13 show the impulse response and the magnitude spectrum of the body of a classical acoustic guitar, respectively. The impulse response can be seen to be long enough so that the temporal envelope can be assumed to be perceptually important. The situation is comparable to the perception of reverberation in a small room. This implies that a digital filter approximation should not be designed exclusively based on magnitude-response criteria. The spectral envelope seen in Fig. 13 is relatively flat but there is a large number of resonances starting from the lowest resonance around 100 Hz.

We have studied several digital filter approximations for modeling the body of the acoustic guitar [17]. An FIR filter model of the body response must be at least 50 to 100 ms long (more than 1000 taps) to yield good synthetic sound. Linear prediction (LPC) analysis suggests an all-pole filter model of order 500 or more. Both of these are computationally too expensive for real-time implementation using a single DSP processor. We also designed reduced-order IIR filters that approximate the frequency resolution of the human auditory system but even these did not reduce the computational load enough without audible degradation of the sound quality.

To overcome the inherently heavy computational load of the filter-based body models a novel method was invented [7], [9]–[11]. Let us consider the string instrument model of Fig. 14a. The output signal $y(n)$ of this system can be expressed as

$$y(n) = e(n) * h(n) * b(n) \quad (7)$$

where the asterisk denotes the operation of linear discrete convolution, $e(n)$ the excitation signal [i.e., the impulse response of the excitation filter $E(z)$], and $h(n)$ and $b(n)$ the impulse responses of the generalized KS string model [$H(z) = P(z)S(z)$] and of the body, respectively. Fundamental properties of linear operators, such as associativity and commutativity, are valid for discrete convolution, and thus Eq. (7) can be rewritten as

$$y(n) = x(n) * h(n) \quad (8)$$

where we have defined

$$x(n) = e(n) * b(n) \quad (9)$$

Thus, it is possible to swap the transfer functions of the generalized KS model and the body, and, in addition, $e(n)$ and $b(n)$ can be combined as shown in Fig. 14b. The signal $x(n)$ is then used as the input to the generalized KS string model. The combination of the excitation signal and the impulse response of the body is motivated by the fact that then the body need not be modeled explicitly during real-time synthesis. The signal $x(n)$ combining the excitation and the body response can be estimated by precomputing it based on some model, by measuring it somehow, or by inverse filtering a digital recording of an instrument as will be discussed in Section 3.5.

We may develop the model further by explicitly modeling the excitation signal. The first step towards this direction is to extract the most prominent resonances of the body, measure their central frequencies and Q values, and design a second-order all-pole filter to represent each resonance. These resonances must then be removed from the excitation signal, e.g., by using narrow-band linear-phase FIR filters.

We have successfully extracted the two lowest resonances of the body of an acoustic guitar. The main advantage of this approach is that the residual signal with the low-frequency high-Q resonances removed decays more rapidly than the original one. As a consequence, a shorter excitation signal can be used and the memory requirements for the model-based synthesizer are lowered. Also, the Q values and central frequencies of the lowest resonances are now parameterized and thus independently controllable.

2.5 Modeling Directional Properties of String Instruments

The directional characteristics of musical instrument sound radiation are of interest in several applications using model-based sound synthesis. From the scientific point of view the physical modeling approach serves as a flexible tool for analysis and simulation of the directivity of the instruments. Directional characteristics must be taken into account when model-based sound synthesis methods are used for sound generation in room acoustics simulation and other virtual reality applications [27].

Plucked string instruments exhibit complex sound radiation patterns due to various reasons. The resonant mode frequencies of the instrument body account for most of the sound radiation (see, e.g., [16]). In string instruments, different mode frequencies of the body have their own patterns such as monopoles, dipoles, or quadrupoles, and their combinations. The sound radiated from the vibrating strings, however, is weak and can be neglected in the simulation.

Another noticeable factor in the modeling of directivity is masking caused by (and reflection from) the player of the instrument. Masking plays an important role in virtual environments where the listener and sound sources are freely moving in a space.

It is clear that computational modeling of the detailed directivity patterns is out of the capacity of real-time DSP sound synthesis. It is therefore important to find simplified models that are efficient from the signal processing point of view and as good as possible from the perceptual point of view. In an earlier study we have considered three strategies [27]:

- 1) directional filtering
- 2) set of elementary sources
- 3) direction-dependent excitation

A direction-dependent digital filter may be attached to each path from the source to the listener. Moving and rotating sources can be modeled by changing the filter parameters of the paths in a proper way (e.g., the Leslie effect of a rotating loudspeaker can be simulated). The directional filtering method was studied for the acoustic guitar. We came to the conclusion that even first or second-order directivity filters give useful results thus leading to an efficient implementation.

Figure 15 depicts the modeling of direction-dependent radiation of the acoustic guitar (in the horizontal plane) relative to the main axis radiation. Shown in the figure are magnitude spectra for second-order IIR filters at azimuth angles 90° , 135° , and 180° . The reference magnitude spectrum at 0° is assumed to be flat. The lowpass characteristic is noticeably increased as the relative angle is greater. The measurement was carried out by exciting the bridge of the instrument by an impulse hammer and by registering the reference response at 0° and the related response in various directions. The measured reference and the directional response were fitted separately with first or second-order AR models. A simple division of the models was performed to obtain the pole-zero directivity filter.

Figure 16a shows the transfer function measured at azimuth angles 0° and 180° . Figure 16b shows the response at 0° filtered with a first-order IIR directivity filter and the actual measured response at 180° azimuth. Note that the spectral slopes are nearly the same, as was expected. In this example, the transfer function of the directional filter is

$$R(z) = \frac{0.2041 - 0.1431z^{-1}}{1 - 0.4389z^{-1}} \quad (10)$$

When zooming to the details of the lowest resonance modes we notice, as described in [16] that the individual modes behave differently. To model such details, a relatively high-order directional filter is needed. It is important to notice that due to the critical band frequency resolution, an *auditory smoothing* may be applied to the responses before directional filter estimation. This helps to reduce the order of the filter.

The use of elementary sources is based on the idea that the radiation pattern of an instrument is approximated by a small number of monopoles or dipoles. In general the method is computationally expensive if a large number of paths to the receiver is needed since each new elementary source adds a new set of path filters.

In the case of commutative excitation (e.g., a plucked string instrument model shown in Fig. 14b) the directivity filtering may be included in the composite excitation in a way similar to the inclusion of the (early) room response [7]. The problem with this method is that the same number of string model instances are to be run in parallel as there are directions to be included. This limits the number of simulated directions, e.g., to the six main directions of the Cartesian coordinate system. Even then this approach is computationally very inefficient.

The considerations above as well as our experiments have shown that the directional filtering technique is normally the most practical one. A first or second-order filter approximation is often a satisfactory solution in a real-time implementation.

2.6 Extensions of the String Model

The string model can be extended in several ways to more carefully simulate a physical vibrating string. In the following we present three of them.

2.6.1 Sympathetic Vibrations

Sympathetic vibration is a phenomenon where some modes of a string are excited by the vibration of other strings primarily due to couplings through the bridge. It can be simulated by feeding a small fraction of the output of each string model to the other strings [4], [11]. This simplified technique does not take into account the frequency-dependency of sympathetic couplings but it still adds realism to the synthetic tones. Since there is feedback via all strings, the values of the coupling coefficients must be carefully adjusted small enough not to make the system unstable. Another and theoretically more correct way to incorporate the sympathetic vibrations is to represent the bridge by a separate filter that is common to all strings in a waveguide model of an instrument [7]. This approach adds, however, computational complexity to the overall model.

2.6.2 Dual-Polarization String Model

Physical strings vibrate in both vertical and horizontal direction. If the effective length of the string is not the same in the two polarization planes, the mixing of the two subsignals of slightly different frequency causes beats to the sound. This has been found to be an important feature in the sound quality of the kantele [18], where the explanation for this effect is the peculiar way in which the strings have been terminated. In other string instruments, however, the effective length in the two directions can differ due to changes in the driving-point impedance of the bridge. This can be incorporated in the model by using two basic string models with different L for each string [11]. Figure 17 illustrates a complete model including dual-polarization string, wavetable excitation, and sympathetic couplings.

In the case of the mandolin, beating is caused by the fact that there are four pairs of equally-tuned strings. This can be modeled using two separate string models for each pair.

2.6.3 Nonlinearities

In strings, the polarization of vibration may change continuously in a complex manner since there are, e.g., weak nonlinearities to transfer energy between the modes and polarizations. In the kantele, the bending of the tuning peg at one end of the strings has been shown to induce nonlinearity due to longitudinal forces [18]. This effect has been successfully simulated in the following way: square the delay line signal, filter it using a leaky integrator, and add the result to the output of the string model. This procedure boosts the even harmonics of the signal.

In [28], a passive nonlinear filter was proposed for producing effects similar to mode-coupling in string instruments. A first-order digital allpass filter is attached to the end of the delay line of the string model. The coefficient of the allpass filter depends on the sign of the delay line signal. This technique was found to be suitable for producing synthetic signals that exhibit time-varying behavior in their decay part.

3 CALIBRATION OF MODEL PARAMETERS

The synthesis system includes three parts that completely determine the character of the synthetic sound: the string model, the plucking point equalizer, and the input sequence. This implies that to calibrate the model to some particular instrument it is needed to estimate the values for the length of the delay line L , the coefficients of the loop filter $H_l(z)$, the delay M and the parameter r_n of the plucking-point equalizer, and the input signal $x(n)$. In this section, the parameter estimation procedures are described that will extract these values.

3.1 Estimation of the Delay Line Length L

The delay length L (in samples) determines the fundamental frequency f_0 of the synthetic signal according to Eq. (3) so that

$$f_0 = f_s / L \quad (11)$$

where f_s is the sample rate. For pitch detection we have used a well-known method based on the autocorrelation function. The short-term or windowed autocorrelation function is defined as [29]

$$\phi_k(m) = \frac{1}{N} \sum_{n=0}^{N-1} [y(n+k)w(n)y(n+k+m)w(n+m)], \quad 0 \leq m \leq M \quad (12)$$

where $y(n)$ is the signal to be analyzed and $w(n)$ is a window function (e.g., the Hamming window). An estimate for the pitch is obtained by searching for the maximum of the $\phi_k(m)$ for each k .

Typical pitch contours of a guitar and a kantele tone are shown in Fig. 18. The contours have been smoothed using a 3-point running median filter. In both cases the pitch decreases with time and approaches a constant value which it reaches within about 0.5 s. In the kantele, the decrease of the fundamental frequency is quite substantial. The problem now is to determine the best estimate \hat{f}_0 for the fundamental frequency in a perceptual sense. In practice a good solution is to use the average pitch value after some 500 ms as the nominal value, since it is important to have a reliable pitch estimate towards the end of the note. This improves the quality of synthesized tones.

When the estimate \hat{f}_0 for the fundamental frequency has been chosen, the effective length L of the delay line can be computed as

$$L = f_s / \hat{f}_0 \quad (13)$$

When also the loop filter has been designed, we can subtract its phase delay from L and define the nominal fractional delay L_F as

$$L_F = L - \tau_H(\hat{\omega}_0) - \text{floor}[L - \tau_H(\hat{\omega}_0)] \quad (14)$$

where $\hat{\omega}_0 = 2\pi\hat{f}_0$. The delay L_F is used as the desired delay in the design of the fractional delay filter $F(z)$. In practice the phase delay of this filter can be expressed as $\tau_F(\hat{\omega}_0) = M_F + L_F$, where $M_F \in \mathbb{Z}_+$ is the integral part of the phase delay that depends on the order of the fractional delay filter and L_F is the fractional part (see, e.g., [18]).

3.2 Measuring the Frequency-Dependent Damping

Next we discuss how to measure the damping of the string as a function of frequency. As discussed in Section 2.1, all losses including the reflection at the two ends of the string are incorporated in the loop filter $H_l(z)$. In order to design a loop filter we need to estimate the damping factors at the harmonic frequencies of the sound signal. This is achieved using the *short-time Fourier transform* (STFT) and tracking the amplitude of each harmonic.

The STFT of $y(n)$ is a sequence of discrete Fourier transforms (DFT) defined as

$$Y_m(k) = \sum_{n=0}^{N-1} w(n)y(n+mH)e^{-j\omega_k n} \quad \text{for } m = 0, 1, 2, \dots \quad (15)$$

where N is the length of the DFT, $w(n)$ is a window function, and H is the hop size or time advance (in samples) per frame. In practice we compute each DFT using the FFT algorithm. To obtain a suitable compromise between time and frequency resolution, we use a window length of four times the period length of the signal. The overlap of the windows is 50% implying that H is 0.5 times the window length. We apply excessive *zero-padding* by filling the signal buffer with zeros to reach $N = 4096$ in order to increase the resolution in the frequency domain. The spectral peaks corresponding to harmonics can be found from the magnitude spectrum by first finding the nearest local minimum at both sides of an assumed maximum. The largest magnitude between these two local minima is assumed to be the harmonic peak. The estimates for the frequency and magnitude of the peak are fine-tuned by applying *parabolic interpolation* around the local maximum and solving for the value and location of the maximum of this interpolating function (see [30] or [31] for details). The number of harmonics to be detected is typically $N_h < 20$ (for the acoustic guitar). The STFT analysis is applied to the portion of the signal that starts

before the attack and ends after some 0.5-1 s after the attack.

The spectral peak detection results in a sequence of magnitude and frequency pairs for each harmonic. The sequence of magnitude values for one harmonic is called the *envelope curve* of that harmonic. A straight line is matched to each envelope curve on a dB scale, since ideally the magnitude of every harmonic should decay exponentially, i.e., linearly on a logarithmic scale. Measurements show that this idealized case is rarely met in practice, and many different kinds of deviations are common. It is possible to decrease the error in the fit by starting it at the maximum of the envelope curve [32] and by terminating it before the data gets mixed with the noise floor, e.g., after the decay of about 40 dB. As a result, a collection of slopes β_k for $k = 1, 2, \dots, N_h$ is obtained (See Fig. 19). The corresponding *loop gain* of the string model at the harmonic frequencies is computed as

$$G_k = 10^{\frac{\beta_k L}{20H}} \quad \text{for } k = 1, 2, \dots, N_h \quad (16)$$

where β_k are the slopes of the envelopes, and H the hop size. The sequence G_k determines the prototype magnitude response of the loop filter $H_l(\omega)$ at the harmonic frequencies ω_k , $k = 1, 2, \dots, N_h$, as illustrated in Fig. 20.

The desired phase response for $H_l(\omega)$ can be determined based on the frequency estimates of the harmonics. We do not, however, try to match the phase response of the filter. There are two principal reasons for this. First, we believe that it is much more important to match the time constants of the partials of the synthetic sound than the frequencies of the partials, since quite small deviations from harmonic frequencies occur in the case of plucked string instruments. In case we wanted to resynthesize strongly inharmonic tones, the phase response of the loop filter should be considered. The second reason is that as we want to use a low-order loop filter, the complex approximation of the frequency response would not be very successful. Thus we restrict ourselves to magnitude approximation only in the loop filter design.

3.3 Loop Filter Design

It is known that the hearing is sensitive to the change of decay rate of a sinusoid, and in practice we measure the time constant of some 20 lowest harmonics with the object of matching the frequency response of the loop filter $H_l(z)$ to that data. Since we use a first-order loop filter, it is clear that there are more restrictions than unknowns in this problem and only an approximate solution is possible.

In principle, the loop filter should be designed based on an auditory criterion, but since that would be complicated, we have decided to use weighted

least squares design. The error function to be minimized in the magnitude only approximation can be defined as

$$E = \sum_{k=0}^{N_h-1} W(G_k) \left[|H_l(\omega_k)| - G_k \right]^2 \quad (17)$$

where N_h is the number of frequency points where the loop gain G_k is approximated, $\omega_k = k\hat{\omega}_0$ are the central frequencies of the N_h lowest harmonics, and $W(G_k)$ is a nonnegative error weighting function. It is reasonable to choose a weighting function $W(G_k)$ that gives a larger weight to the errors in the time constants of the slowly decaying harmonics since the hearing tends to focus on their decay. A candidate for such a weighting function is

$$W(G_k) = \frac{1}{1 - G_k} \quad (18)$$

We require that $0 \leq G_k < 1$ for all k , which is a physically reasonable assumption since the system to be modeled is passive and stable.

Let us denote the numerator of Eq. (6) by

$$A = g(1 + |a_1|) \quad (19)$$

and $H_l(\omega)$ with the numerator removed by

$$\tilde{H}_l(\omega) = \frac{H_l(\omega)}{A} \quad (20)$$

Then Eq. (17) can be rewritten as

$$E = \sum_{k=0}^{N_h-1} W(G_k) \left[|A\tilde{H}_l(\omega_k)| - G_k \right]^2 \quad (21)$$

The gain of the loop filter at low frequencies, i.e. g , can be chosen based on the loop gain values of the lowest harmonics. In many cases it is good enough to set $g = G_0$ whereas sometimes the average of two or three lowest loop gain values gives a better result.

The value for the coefficient a_1 that minimizes E can be found by differentiating (21) with respect to a_1 . This yields

$$\frac{\partial E}{\partial a_1} = 2A_0 \sum_{k=0}^{N_k-1} W(G_k) \frac{\partial |\tilde{H}_l(\omega_k)|}{\partial a_1} \left[|A_0 \tilde{H}_l(\omega_k)| - G_k \right] \quad (22)$$

By substituting Eqs. (6) and (18), we can write

$$\frac{\partial E}{\partial a_1} = 2A_0 \sum_{k=0}^{N_k-1} \frac{A_0 \cos \omega_k (1 + a_1 \cos \omega_k)^{-3} - G_k \cos \omega_k (1 + a_1 \cos \omega_k)^{-1}}{1 - G_k} \quad (23)$$

The aim is now to find the zero of this function. In practice we find a near-optimal solution in the following way: the value of the derivative is evaluated and depending on the sign of the result, a_1 is changed by a small increment, the derivative is evaluated again, and so on. After the derivative has reached a very small value, the iteration is terminated and the final value for a_1 is used in the synthesis model. We have verified the convergence of this design procedure in practice by analyzing signals generated using the synthesis model. The loop filter designed based on the analysis data yields the same filter parameters—within numerical accuracy—as used in the synthesis. Also match with natural tones has been found to be satisfactory in many cases. Figure 20 illustrates the loop filter design for a typical kantele tone.

3.4 On the Estimation of the Plucking Point

It is well understood that when a string is put to vibration by plucking it, the sound signal will lack those harmonics that have a node at the plucking point (see, e.g., [16]). However, in general the string is not plucked exactly at the node of any of the lowest harmonics and since the amplitude of the higher harmonics is considerably small anyway, it is not possible to accurately detect the plucking point by simply searching for the lacking harmonics in the magnitude spectrum. Another practical problem may occur because of nonlinear behavior of the string. Namely, the amplitude of vibration of a weak harmonic can gain energy from other modes so that its amplitude begins to rise reaching a maximum about 100 ms after the attack and then begin to decay [33]. This can often be seen in the analysis of harmonic envelopes of the guitar. For these reasons we believe that a more comprehensive understanding of the effect of plucking point can be achieved by studying the time-domain behavior of the string in terms of the short-time autocorrelation function.

Estimation of the plucking position is an inherently difficult problem since a recorded tone can include contributions of several delays of approximately the same magnitude, e.g., early reflections from objects near the player, such as the floor, the ceiling, or a wall. To minimize the effect of these additional

factors, we suggest analysis of tones recorded in an anechoic chamber.

It is, however, not absolutely obligatory to estimate and model the effect of the plucking position. Its contribution can be left in the excitation signal obtained using inverse filtering, which is described in the following section.

3.5 Inverse Filtering

The input signal $x(n)$ can be estimated using inverse filtering, i.e., filtering of the signal $y(n)$ —which is now assumed to be the output of the model—with the inverted transfer function $S^{-1}(z)$ of the string model. The transfer function of the string model shown in Fig. 9 can be expressed as

$$S(z) = \frac{1}{1 - z^{-L_l} F(z) H_l(z)} \quad (24)$$

If $S(z)$ had zeros outside the unit circle, the inverse filter $S^{-1}(z)$ would be unstable. In that case, inverse filtering would not yield an acceptable result. Fortunately, this is not a problem in practice as can be seen by substituting (6) into (24). Inverting this equation yields

$$S^{-1}(z) = \frac{1 + a_1 z^{-1} - g(1 + a_1 z^{-1}) z^{-L_l} F(z)}{1 + a_1 z^{-1}} \quad (25)$$

This technique is simple to apply but since the order of the loop filter is low, the harmonics are not canceled very accurately. The resulting $x(n)$ may suffer from high-frequency noise. The low-order loop filter was chosen primarily since it is then efficient to implement the synthesis model. However, inverse filtering is an off-line procedure where accuracy is much more important than efficiency. We could thus design a higher order filter to be used in the inverse filtering (see, e.g., [15]).

Another deficiency of the above inverse filtering technique is that it does not take into account the nonlinear effects of the physical strings. This can be accounted for by using a time-varying inverse filter which cancels the impulse response of the string. This filter can be designed based on the STFT analysis that was discussed above.

However, satisfactory results are obtained when using carefully chosen tones that behave well, i.e., there are no strong nonlinear effects nor beating present. Figure 21 shows the spectrum of a mandolin tone and the magnitude response of the inverse filter. The magnitude spectrum of the residual (the result of inverse filtering) is illustrated in Fig. 22. Note that in this case the harmonics have been canceled quite accurately.

4 SYNTHESIS OF PLUCKED STRINGS

The residual signals resulting from the analysis and inverse filtering of an original string instrument tone can be directly applied as the excitation to the string model, as described in Sections 2 and 3. The synthesis procedure is carried out in the following way:

- 1) obtain the residual signal by inverse filtering,
- 2) window the first 100 ms of the residual signal using, e.g., the right half of a Hamming window,
- 3) use the truncated signal as the excitation to the string model,
- 4) run the string model using the parameters derived from the analysis.

4.1 The Guitars

In the analysis and synthesis results of the electric guitar shown in Figs. 23 and 24 it can be seen that the length of the residual signal used in resynthesis can be reduced to about 50 ms without any significant loss of information. This is due to the lack of body resonances. The resulting sound is quite similar to that of the basic Karplus–Strong model excited with an impulse. It is clear that in this case physical modeling principles should be extended to various magnetic pick-up combinations and amplifiers.

The analysis and synthesis results of the steel-string acoustic guitar were similar to those of the nylon-string acoustic guitar. The use of the plectrum instead of finger in the string excitation results in a different-sounding residual signal, but the string model can be the same for both instruments.

4.2 The Mandolin

The analysis and resynthesis of the mandolin are illustrated in Figs. 25 and 26. We notice that the windowing and truncation of the residual signal has only a minor effect shortening the decay of the lowest body resonances. The resynthesized tone in Fig. 26b has been calculated using the first 100 ms of the residual and a single-polarization string model. The original mandolin signal shows a slight beating effect after the transient part, whereas the resynthesized tone has an exponentially decaying behavior. It is, however, almost impossible to distinguish the synthetic tone from the original one by ear. Using a dual-polarization string model with a very small beating effect would result in an even more natural tone. Figures 27 and 28 depict the temporal characteristics of the six lowest harmonics of the original and synthetic mandolin tones,

respectively. It can be clearly seen that the transient part is identical in both plots. The decay of harmonics is very similar but not identical.

4.3 The Kantele

The use of the dual-polarization string model is essential in the case of the kantele. As shown in Fig. 5, there is significant beating in the signal that clearly effects the sound quality. Modeling this dual-polarization behavior is quite straightforward with two separate string models as was discussed in Section 2.6.2. Figures 29 and 30 depict the analysis and synthesis results for a kantele tone. A dual-polarization string model was used for the synthesis. The nonlinear characteristics of the kantele have been resynthesized according to the principles presented in Section 2.6.3.

4.4 The Banjo

In the analysis of the banjo we experienced that most of the characteristics of a banjo tone are included in the attack part of the signal. The distinctive sound of the banjo is retained in the residual signal after inverse filtering. Figures 31 and 32 depict the results of the analysis/synthesis.

In this example, some of the harmonics of the original signal decay much faster than the first harmonic. The synthesis model with a first-order loop filter is not able to catch this phenomenon. For this reason, the decay rate of some harmonics is too slow in the resynthesized signal. This can be seen by comparing Figs. 31 and 32. Apart from this slight difference, the overall sound quality of the synthetic signal is excellent.

5 REAL-TIME IMPLEMENTATION ON A SIGNAL PROCESSOR

The real-time synthesis models were implemented using a Texas Instruments TMS320C30 floating-point signal processor. This processor is capable of executing a maximum of 30 million floating-point operations per second (30 MFLOPS, 15 MIPS). Our experience was, however, that this limit cannot be reached in practice, except in special cases when register, pipeline and memory conflicts can be totally avoided. Thus our hand-optimized assembly code taking up about 110 instructions per output sample was able to run six strings in real time only with a sampling rate of 22 kHz, including host communication and parameter calculation. However, this was found to be adequate for producing excellent plucked string sounds since there is not much energy in the spectrum of these signals above 10 kHz.

An overview of the software and hardware environment for real-time implementation of physical models is illustrated in Fig. 33. Programs written

in the QuickC30 environment [34] can be run on different hardware platforms without modification. All hardware-dependent details are hidden from the application by using specialized macros and functions. So far the QuickC30 system has been implemented on the Macintosh and the PC platforms. We use commercially available DSP boards.

Our current developments include using a multiprocessor environment for running several instruments in parallel or for a more detailed simulation of the instruments including nonlinear effects. We are experimenting with a system containing two TMS320C31 processors, which are less expensive and slightly reduced versions of the TMS320C30, and we are building a more advanced system using TMS320C40 floating-point processors that support multiprocessing in hardware.

The implementation of plucked string synthesis on a signal processor follows the principles of Fig. 9. Each substring (single polarization) consists of a ring buffer delay line with third-order Lagrange interpolation (for fine tuning) and a loop filter. Table lookup is used for computing the interpolator coefficients from pitch information. Also the loop filter parameters have been table-coded. String excitations are read from wavetables that have been constructed using inverse filtering techniques.

Most control parameters are MIDI-like, i.e., they cover a number range of 0...127. The computation of the model parameters, such as the delay and loop filter parameters, excitation filtering, is done as much as possible by table lookup since computationally expensive divisions, logarithms, and exponents are otherwise needed. (For control parameters, see also Section 6).

The updating of model parameters is executed every 1 ms that is found fast enough in most transients and transitions. The reduced parameter update rate is necessary to keep the computational cost low enough, e.g., in the case of guitar synthesis.

5.1 Multirate String Model

A string instrument tone, like most natural signals, is a lowpass signal the spectrum of which varies through time so that high-frequency components are damped faster than low-frequency components. Thus it would be advantageous to design a multirate synthesis model where the input sequence is fed in at a high sampling rate but where the delay line and the loop filter $H_l(z)$ would run at a rate considerably lower. This idea has also been mentioned by Smith [35, p. 50]. As a result, the attack part of the synthetic sound will still have energy at high frequencies thus preventing the sound to be too lowpass filtered.

Figure 34a illustrates the idea of the multirate string model. In the feedback loop, the signal is first decimated by a factor K , which is a small integer num-

ber, say 2 or 3. In decimation, the signal is lowpass filtered using a linear-phase FIR filter $H_K(z)$ and every K th sample is retained. The delay line length L has to be divided by K to keep the fundamental frequency constant. Also the loop filter $H_l(z)$ has to be redesigned. The output of the loop filter is upsampled by the factor K and added to the input sequence.

A more efficient version of the multirate model is presented in Fig. 34b. Now two versions of the excitation signal are needed: the original one $x_p(n)$ (i.e., the output of the plucking point equalizer) and a decimated one $\tilde{x}_p(n)$. In this model, the excitation signal $x_p(n)$ is not processed at all since the feedback loop is computed separately. The output of the feedback loop, which runs at a lower sample rate, is upsampled using a linear-phase interpolating FIR filter $H_K(z)$.

The advantages of the multirate approach are reduced memory requirements (i.e., shorter delay line) and computational savings due to the lower sample rate in the feedback loop. However, this method also requires a more sophisticated fractional delay filter $F(z)$. The Lagrange interpolator does not work well in this case since its magnitude response error is unacceptable at high frequencies. An allpass fractional delay filter is better suited to this case.

6 CONTROL OF THE STRING INSTRUMENT MODELS

In this section we examine the necessary environment for using the models as musical instruments. Though the ideas described herein are not restricted to physical modeling, we believe that model-based synthesis is superior to conventional methods when it comes to the control of the synthesis. Most model parameters have a direct physical meaning, and thus adjusting them changes the sound in a predictable way. One disadvantage of model-based methods is that each instrument family has to be modeled and controlled differently. In order to get a convincing sound, one has to study the performance techniques used with the given instrument and, in a way, the performer has to be modeled as well.

6.1 Problems Related to String Instruments

Most plucked string instruments have several strings, the length of which can be fixed (e.g., the kantele, or the harp) or variable (as in the lute family). There are several problems of sound synthesis control that appear only with certain types of instruments [36]. Here we consider two of them: *note transitions* and *string allocation*.

Note transitions appear when the next note has to be played on the same string that was used for the previous note. Transitions are perceptually very important. The lack of transitions makes the sound artificial in a melodic con-

text even if individual notes sound realistic. Correct simulation of note transitions depends on two conditions: the ability of controlling the synthesizer in a proper way and the ability of recognizing when transitions are needed. Physical modeling synthesis produces convincing note transitions without any special effort when the parameters of a model are adjusted for a new note. However, special care must be taken for eliminating extra noises caused by parameter updating.

The string allocation mechanism is responsible for finding the proper sound generator (i.e., string model) for a new note. This is straightforward for instruments with fixed strings, since each pitch can be played only on a single string. If the number of generators is less than the number of available pitches on the real instrument (as with certain types of the kantele, or, e.g., in the case of the piano), a voice allocation mechanism is needed [36]. Otherwise each string can have a separate sound generator assigned to it. In this case the algorithm is straightforward: select the generator with the given (fixed) pitch. There are no note transitions in the sound of these instruments.

The problem appears with variable-length strings. To allow polyphony, several strings are used on a single instrument. The pitch ranges of individual strings usually overlap. Thus the pitch does not uniquely determine the string. On the real instrument the performer has to select the appropriate string for a given note. Since individual strings typically have different physical properties (thickness, mass, and tension), choosing a different one will result a slightly different timbre. When the computer has to perform a piece of music, a string allocation mechanism is needed to simulate this decision. This mechanism is entirely different from the voice allocation. It is needless to say that the string allocation determines note transitions as well.

The simplest string allocation mechanism tries to assign the lowest string that can play the given pitch to the note. A more elaborate method would be allocating the strings so that the movement of the left hand would be minimized. This can be accomplished easily if it is possible to preprocess the score. However this latter method cannot be used with real-time control. As a compromise, *nearly-real-time control* can be used [36].

6.2 MIDI and Intelligent Synthesis Control

MIDI has several caveats when used with complex sound synthesis such as physical modeling. In principle, one could use many different controller messages for detailed control over the synthesis. This approach, however, has severe drawbacks: it is rather difficult to control many parameters manually and there are many parameter combinations that simply do not make a sound.

Another problem is that MIDI glues together the pitch set and note trigger-

ing into a single ‘Note On’ message. These events are clearly separated in time on the variable source instruments. Therefore, when such sound is synthesized, these control events should be separated to create natural note transitions.

In spite of these problems it would be convenient to use existing MIDI equipment with new synthesizers. We suggest a new approach for controlling complex synthesis with MIDI: let the control interface—which maps the performance events into control events—be more *intelligent*. By “intelligent” we mean that it should work *reasonably* even if not all details are given explicitly. The missing information can be deducted from a knowledge base created by studying the performance techniques of the given instrument and analyzing the impact of different playing techniques on the sound. This approach has an additional advantage: it is much easier to play an intelligent instrument than a complex but dumb synthesizer.

6.3 Simplifying the Control Problem

The usual timbre space has three independent parameters: pitch, level and timbre. These parameters, however, are too general for controlling specialized synthesizers such as those based on physical models. However, during the performance of a musical piece the timbre will probably not exit a bounded subspace stretched by the possible sounds of, e.g., a given acoustic guitar within a given performance style. Naturally, there will be different subspaces for different instruments and/or performance styles, but the main point is that the performer can choose the adequate subspace *a priori*. It may also be necessary to allow shifting from one subspace into another during performance if the conditions change substantially.

For moving around inside one of these subspaces we can use fewer parameters: pitch and something that we call “expression”. Choosing pitch as the primary parameter is motivated by our interest in traditional Western music, which is entirely based on the twelve-tone scale and where the most important property of a note is the pitch. In our approach all internal parameter calculations are based on the desired pitch.

While pitch is a well-defined property—in practice it corresponds roughly to the fundamental frequency of a note—expression cannot be uniquely defined. Still, everyone “feels” what it means. Although these feelings are not necessarily the same, it can be agreed that a low value of expression means something like “colorless”, “passing”, or “soft”, while a high value can be interpreted as “emotional”, “bright”, “emphasized”, or “loud”. Moreover, the interpretation of the expression seems to be closely related to the performance style. Realizing this led to the idea of combining the timbre *and* the mapping of pitch and expression to the actual synthesis parameters into *performance styles*,

like “classical guitar”, “blues guitar” or “flamenco guitar”.

Reducing the control parameters to only two drastically reduces the complexity of the control interface and it still allows human interaction. For more detailed control, the attack part of the notes, which carries the most important indication of the playing technique, could be controlled separately. On the keyboard these expression controls could be assigned to the aftertouch and key velocity, respectively.

6.4 Overview of QuickMusic

QuickMusic is the musical sound synthesis control package of the QuickSig system (see Fig. 33). It is written entirely in Common Lisp/CLOS. It contains a platform-independent sequencer that uses a Lisp-syntax musical notation and a real-time MIDI interface. Originally the package has been developed under Macintosh Common Lisp (MCL). Recently we have ported it (together with other parts of the QuickSig environment) to the PC platform under Allegro Common Lisp for Windows.

6.5 The Lisp Sequencer

The Lisp sequencer can be used to quickly create test note sequences or even demo pieces. For describing a score a Lisp-like musical notation is used allowing easy parameterizing of the individual notes by keyword parameters. The sequencer is object-oriented and thus easily expandable.

The score is processed in four phases: parsing, performing, exploding and playing. During the “parse” phase a high-level object description of the note sequence is generated. The “perform” phase was included to allow automatic performance of the piece using performance rules. This part has not been exploited so far. The “explode” phase generates (usually several) low-level control events from the notes. Finally the built-in scheduler plays the piece on the DSP hardware in real time.

6.6 MIDI Control Interface

The most important elements of QuickMusic are the different control interfaces that map the input MIDI events (key on/off, wheel movement, etc.) to the actual synthesis parameters. They were added to allow interactive performance and easy experimenting with real-time synthesis algorithms. As an additional advantage of using MIDI an external MIDI sequencer can be used to record and replay performances.

A platform-independent MIDI message dispatch system has been built on top of the low-level interface. Message dispatching is accomplished automatically by the CLOS method dispatching mechanism, since the generic method

process-midi-message is specialized on both the instrument class and the type of the MIDI message. The MIDI control interfaces are implemented as different instrument classes.

6.7 Operating Modes

The interactivity of Lisp makes it easy to experiment with different control strategies. We have developed several different modes partly to overcome the limitations posed by MIDI itself or making easier to play “guitarlike” on a normal keyboard. The keyboard, of course, by no means can replace a guitar controller (just as much as the synthesizer does not replace the acoustic guitar), but these specialized modes help in getting the most out of the synthesizer in terms of fidelity and ease of playing.

6.7.1 Standard MIDI Modes

For being able to use existing MIDI files sequenced by others we have implemented the standard MIDI control modes as an option.

Mono Mode

The MIDI mono mode is often used with MIDI guitar controllers to send the performance data on separate MIDI channels for each string. This mode simplifies the MIDI implementation in the receiver. There is one sound generator per channel and there is no need for voice allocation. Ambiguous pitches, which could be played on different strings, do not cause a problem. Pitch bend can be used individually on different strings without affecting other notes.

While mono mode is perfectly suited for the MIDI guitar, it is difficult to use it with a keyboard. Although keyboard split could be used to send data on different channels, playing chords is quite difficult if not impossible this way.

Poly Mode

In the poly mode, the data for all strings is received on a single MIDI channel and the control interface has to assign the notes to available sound generators. There is a potential problem with instruments of the variable source type: assigning a different generator to a new note will not result in a correct transition. String allocation is necessary for most string instruments.

6.7.2 Smart MIDI Modes

Smart MIDI modes are enhanced MIDI modes. While they accept MIDI events intended for normal use, they try to add more details to the sound, e.g., by splitting the ‘Note On’ to separate pitch set and note start events. Since setting the pitch happens before starting the note, this can be only achieved by

delaying the start of a note by a small amount. Other enhancements, like *automatic fret noise generation*, need even more delay. Allowing a relatively large (0.5–1 s) constant processing delay helps in both cases. When the synthesizer is driven by a sequencer the delay can be compensated by advancing the given track by the amount of delay. Unfortunately, this technique cannot be used in real-time performance.

6.7.3 Special Control Modes

We have experimented with advanced control modes for the guitar synthesizer. They help in creating a natural sounding guitar performance using a standard MIDI keyboard. Two basic styles have been investigated: solo playing and strumming.

Solo Guitar

Playing a guitar solo is slightly different from using the MIDI mono mode. On the guitar a solo is played usually on several strings. Therefore automatic string allocation is used. In solo mode all special playing techniques that will be discussed later are available.

Rhythm Guitar

One difficulty of playing voiced guitar chords correctly on the keyboard is the relatively large inter-note separation of the chord members. To reduce this problem we have implemented an intelligent chord recognizer allowing simplified chord-fingering. The algorithm tries to fit different chords to the depressed keys and the most probable chord is selected (e.g., A, Am, A7, etc.). It is then revoiced for the guitar using lookup tables. When alternative voicings are possible (as in most cases) the position of the previously played chord can be used to select the chord in the closest position. The player can influence the selection of chord position by using different octaves of the keyboard.

Each performance style can have its own set of chords that is based on the most often used chords of the given style. Furthermore, styles can contain chords that use only the upper strings leaving the lower strings free for playing bass notes as in, e.g., folk or Latin-American styles. In this case the bass notes can be played on another part of the keyboard using the keyboard split mode, or they can be played by using a special note triggering mode. The notes are *not* triggered immediately by fingering a chord but separate keys—one key per string—are used to pluck the strings afterwards. This technique can be used for playing nice arpeggios as well.

With these techniques many common guitar styles can be played from the keyboard with little effort.

6.8 Mixing Control Modes

It is possible to switch between control modes during performance. This feature helps to play, e.g., chordal solos interleaved with short runs. It is also possible to split the keyboard and define different control modes over the two ranges.

6.9 Special Techniques

In the following section we discuss some special playing techniques not available on keyboard instruments. These techniques can be simulated with our intelligent control interfaces.

Hammer-On and Pull-Off

Hammer-on and *pull-off* are left-hand techniques that a guitar player sometimes uses to pluck a note. By striking a finger against a fret, one can set the string into motion without plucking it with the right hand. This technique can be used if the pitch of the hammered note is higher than the previous one on the same string. It is convenient for playing grace notes or fast passages. A similar technique is fret-tapping, used by many electric guitarists, which allows playing very fast solos with both hands on the fretboard.

Pull-off is a complementary technique, which results in a lower pitch. The left hand finger releases the fret used for the previous note, and at the same time it plucks the string by pulling it a little sideways. Advanced players can play complete melodies without using their right hand at all.

When hammer-on recognition is on, the control interface assumes that a new ‘note on’ that is played on an already sounding string should be a ‘hammer-on’ or a ‘pull-off’, depending on the direction of the pitch change. In the solo mode this can be accomplished by hitting the next key before the previous one has been released (similarly to the fingered portamento in mono mode on many synthesizers). In poly mode the string allocation algorithm suggests a hammer-on or pull-off, when it cannot allocate a new string with the given configuration. In every case, releasing a key before the next one is pressed ensures, that the new note will be plucked.

Tremolo Picking

Tremolo picking is a fast up-down alternate stroke plucking technique, typically used with the mandolin, the bouzouki, or the balalaika. The string is not damped manually between the individual strokes but a new pluck will automatically damp the previous note creating a characteristic sound. This effect can be somehow imitated on a normal keyboard by fast repetition of a key. However, this way it is not possible to tell the stroke direction, which is important

in double-stringed instruments. Moreover, the standard interpretation of 'Note On' and 'Note Off' would cut away the note in between the repetitions what sounds very unnatural. Though the sustain pedal could be used to avoid this artifact, as a result the new notes would be triggered without damping the old ones thus making a ringing effect instead of the tremolo picking.

The use of the special string excitation keys (discussed in the section Rhythm Guitar above) automatically results in correct transition to the new note. Furthermore, there is a separate 'tremolo key', which, when pressed, plucks last note again with a different pluck sequence corresponding to the opposite stroke direction.

Fingered Vibrato

Automatic vibrato sounds usually rather unnatural. It can be replaced by mapping the keyboard aftertouch to 'pitch bend up' with a small pitch bend range. This way expressive vibratos can be played easily by hand.

Muting the Strings

Unlike on the piano, on the guitar the strings are not muted automatically. The guitar player can mute them either by using his/her right hand or palm, or by decreasing the pressure on the frets with the left hand (a technique often used, e.g., in country or beat strumming styles). Both techniques result in a quick damping of the string. Right hand muting often causes an extra sound when the palm hits the guitar body and the strings. In contrast, left hand muting usually does not introduce extra noise and it is often only partial, damping the higher harmonics more than the fundamental.

In our system muting can be accomplished by stepping on the sustain pedal. The same effect can be assigned to a key. Partial muting—an often used effect with the electric guitar—can be controlled with the expression pedal.

Changing Pluck Position

Plucking a string close to the bridge causes a softer, brighter and sharper tone, while plucking towards the neck makes a louder, mellower sound. Because of the position of the right hand fingers, low strings are usually plucked further away from the bridge than the higher ones. Moreover, varying the pluck position has an artistic effect by changing the overall timbre. On the keyboard the position can optionally be mapped to the key velocity (softer touch will move it closer to the bridge) or to the modulation wheel.

6.10 Parameter Calculation

The calculation of the actual synthesis parameters—delay line length, loop filter coefficients, and fractional delay filter coefficients—is based on two-dimensional lookup-tables indexed by the desired pitch and expression values. We are experimenting with an alternative method that employs a neural network for implementing the nonlinear mapping functions. Both methods use parameters derived from actual performances recorded previously and analyzed by a computer.

7 CONCLUSIONS AND FURTHER WORK

A physically based modeling technique for plucked string instruments has been developed aiming at high-quality sound synthesis. The model for a vibrating string used in this method is based on a digital waveguide model which is constructed of digital delay lines and linear filters. Many approaches to the modeling of the body of string instruments have been studied and particularly successful results in terms of efficiency and sound quality have been obtained by using an inverse filtered input signal.

The parameter values of the synthesis model can be calibrated based on the analysis of acoustic signals and thus the model can imitate the sound of a real instrument. The analysis consists of pitch detection and short-time Fourier analysis. Estimates for the length of the delay line of the string model and the coefficients of the loop filter can be obtained based on these analysis data.

The synthesis model accurately reproduces the attack part (the first 100 ms or less) of the signal and approximates the average decay rate of the harmonics. Since these two aspects are of great importance in the recognition of a musical instrument, the individual synthetic signals are often indistinguishable by ear from the original ones.

Principles presented in this paper can be applied to many plucked string instruments. We have investigated model-based synthesis of the acoustic guitar, the steel-string guitar, the electric guitar, the banjo, the mandolin, and the kantele. Nonlinear extensions that are required for imitating beating of harmonics were also discussed. Real-time implementation of the synthesis model using a signal processor was studied and a novel multirate implementation technique was developed. Several aspects of controlling the physical models of plucked string instruments were studied.

Future work in modeling plucked strings instruments includes design of a more sophisticated inverse filtering technique that would take into account the time-varying character of each harmonic. Furthermore, a method to estimate the plucking point is needed. The plucking point of the string has to be determined to cancel its effect from the analyzed signal.

8 ACKNOWLEDGMENTS

The authors wish to thank Dr. Timo I. Laakso for his helpful comments on an earlier version of this paper and Dr. Julius O. Smith for inspiring discussions on model-based synthesis. The various plucked string instruments were played by Jukka Savijoki, Lassi Logrén, Tuomas Logrén, and Tuomo Laakso. Their contribution is acknowledged. This research was financially supported by the Academy of Finland.

9 REFERENCES

- [1] C. Roads and J. Strawn, eds., *Foundations of Computer Music*, (MIT Press, Cambridge, Massachusetts, 1985).
- [2] F. R. Moore, *Elements of Computer Music*, (Prentice-Hall, Englewood Cliffs, New Jersey, 1990).
- [3] K. Karplus and A. Strong, "Digital synthesis of plucked string and drum timbres," *Computer Music J.*, vol. 7, no. 2, pp. 43–55 (1983).
- [4] D. Jaffe and J. O. Smith, "Extensions of the Karplus-Strong plucked string algorithm," *Computer Music J.*, vol. 7, no. 2, pp. 56–69 (1983).
- [5] J. O. Smith, *Efficient Yet Accurate Models for String and Air Columns Using Sparse Lumping of Distributed Losses and Dispersion*, Report No. STAN-M-67, CCRMA, Dept. of Music, Stanford University (Stanford, CA, 1990 Dec.).
- [6] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music J.*, vol. 16, no. 4, pp. 75–87 (1992).
- [7] J. O. Smith, "Efficient synthesis of stringed musical instruments," in *Proc. 1993 Int. Computer Music Conf.*, pp. 64–71 (Tokyo, Japan, Sept. 10–15, 1993).
- [8] M. Karjalainen and U. K. Laine, "A model for real-time synthesis of guitar using a floating-point signal processor," in *Proc. 1991 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 5, pp. 3653–3656 (Toronto, Canada, 1991 May).
- [9] 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, July 29–Aug. 1, 1993). Sound examples included on the SMAC'93 CD.
- [10] J. O. Smith, M. Karjalainen, and V. Välimäki, personal communication (New Paltz, New York, 1991 Oct.).
- [11] M. Karjalainen, V. Välimäki, and Z. Jánosy, "Towards high-quality synthesis of the guitar and string instruments," in *Proc. 1993 Int. Computer Music Conf.*, pp. 56–63 (Tokyo, Japan, Sept. 10–15, 1993).
- [12] J. O. Smith, "Synthesis of bowed strings," in *Proc. 1982 Int. Computer*

Music Conf., pp. 308–340 (Venice, Italy, 1982).

[13] J. O. Smith, *Techniques for Digital Filter Design and System Identification with Applications to the Violin*, Ph.D. thesis. Report No. STAN-M-14, CCRMA, Dept. of Music, Stanford University, (Stanford, CA, 1983 June).

[14] J. Laroche and J. M. Jot, “Analysis/synthesis of quasi-harmonic sound by the use of the Karplus–Strong algorithm,” in *Proc. Second French Congress on Acoustics*, (Arcachon, France, 1992 April).

[15] J. Laroche and J.-L. Meillier, “Multi-channel excitation/filter modeling of percussive sounds with application to the piano,” *IEEE Trans. Speech and Audio Processing*, vol. 2, no. 2, pp. 329–344 (1994 April).

[16] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*, (Springer Verlag, New York, 1991).

[17] M. Karjalainen, U. K. Laine, and V. Välimäki, “Aspects in modeling and real-time synthesis of the acoustic guitar,” in *Proc. 1991 IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics* (New Paltz, New York, 1991 Oct.).

[18] M. Karjalainen, J. Backman, and J. Pölkki, “Analysis, modeling, and real-time sound synthesis of the kantele, a traditional Finnish string instrument,” in *Proc. 1993 IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, vol. 1, pp. 229–232 (Minneapolis, MN, April 27–30, 1993).

[19] M. Karjalainen, “DSP software integration by object-oriented programming: a case study of QuickSig,” *IEEE ASSP Magazine*, vol. 7, no. 2, pp. 21–31 (1990 April).

[20] J. O. Smith, *Music Applications of Digital Waveguides*. Report No. STAN-M-39, CCRMA, Dept. of Music, Stanford University (Stanford, CA, May 1987).

[21] C. R. Sullivan, “Extending the Karplus–Strong algorithm to synthesize electric guitar timbres with distortion and feedback,” *Computer Music J.*, vol. 14, no. 3, pp. 26–37 (1990).

[22] T. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, *Crushing the Delay—Tools for Fractional Delay Filter Design*, Report no. 35. Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing (Espoo, Finland, 1994 Oct.). Submitted to the *IEEE Signal Processing Magazine*.

[23] T. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, “Real-time implementation techniques for a continuously variable digital delay in modeling musical instruments,” in *Proc. 1992 Int. Computer Music Conf.*, pp. 140–141 (San Jose, CA, 1992 Oct.).

[24] A. Paladin and D. Rocchesso, “A dispersive resonator in real-time on MARS workstation,” in *Proc. 1992 Int. Computer Music Conf.*, pp. 146–149 (San Jose, CA, 1992 Oct.).

[25] S. A. Van Duyne and J. O. Smith, "A simplified approach to modeling dispersion caused by stiffness in string and plates," in *Proc. 1994 Int. Computer Music Conf.*, pp. 407–410 (Aarhus, Denmark, Sept. 12–17, 1994).

[26] A. Chaigne, A. Askenfelt, and E. Jansson, "Time domain simulations of string instruments. A link between physical modeling and musical perception," *J. de Physique IV, Colloque C1, supplément au Journal de Physique III*, vol. 2, pp. C1-51–C1-54 (1992 April).

[27] J. Huopaniemi, M. Karjalainen, V. Välimäki, and T. Huottilainen, "Virtual instruments in virtual rooms—a real-time binaural room simulation environment for physical models of musical instruments," in *Proc. 1994 Int. Computer Music Conf.*, pp. 455–462 (Aarhus, Denmark, Sept. 12–17, 1994).

[28] S. A. Van Duyne, J. R. Pierce, and J. O. Smith, "Traveling wave implementation of a lossless mode-coupling filter and the wave digital hammer," in *Proc. 1994 Int. Computer Music Conf.*, pp. 411–418 (Aarhus, Denmark, Sept. 12–17, 1994).

[29] L. R. Rabiner, "On the use of autocorrelation analysis for pitch detection," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 25, no. 1, pp. 24–33 (1977 Feb.).

[30] J. O. Smith and X. Serra, "PARSHL: An analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in *Proc. 1987 Int. Computer Music Conf.*, pp. 290–297 (Urbana, Illinois, 1987 Aug.).

[31] X. Serra, *A System for Sound Analysis/Transform/Synthesis Based on a Deterministic plus Stochastic Decomposition*, Ph.D. thesis. Report No. STAN-M-58, CCRMA, Dept. of Music, Stanford University (Stanford, CA, 1989 Oct.).

[32] J. O. Smith, personal communication (Espoo, Finland, 1993 Aug.).

[33] K. A. Legge and N. H. Fletcher, "Nonlinear generation of missing modes on a vibrating string," *J. Acoust. Soc. Am.*, vol. 76, pp. 5–12 (1984 July).

[34] M. Karjalainen, "Object-oriented programming of DSP processors: a case study of QuickC30," in *Proc. 1992 IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP'92)*, vol. V, pp. 601–604 (San Francisco, CA, March 23–26, 1992).

[35] J. O. Smith, *Digital Waveguide Modeling of Musical Instruments*, unpublished manuscript, CCRMA, Stanford University (Stanford, CA, July 24, 1993).

[36] Z. Jánosy, M. Karjalainen, and V. Välimäki, "Intelligent synthesis control with applications to a physical model of the acoustic guitar," in *Proc. 1994 Int. Computer Music Conf.*, pp. 402–406 (Aarhus, Denmark, Sept. 12–17, 1994).

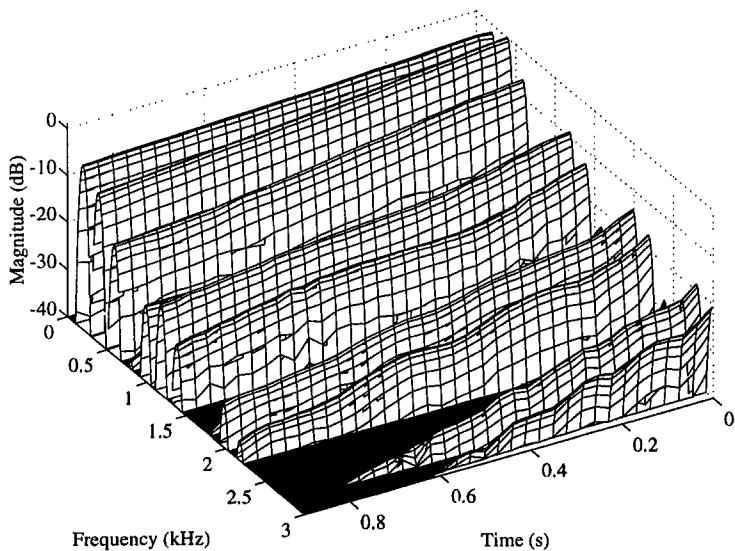


Fig. 1. *STFT analysis of an electric guitar tone. (Fender Stratocaster, third open string, fundamental frequency 195 Hz).*

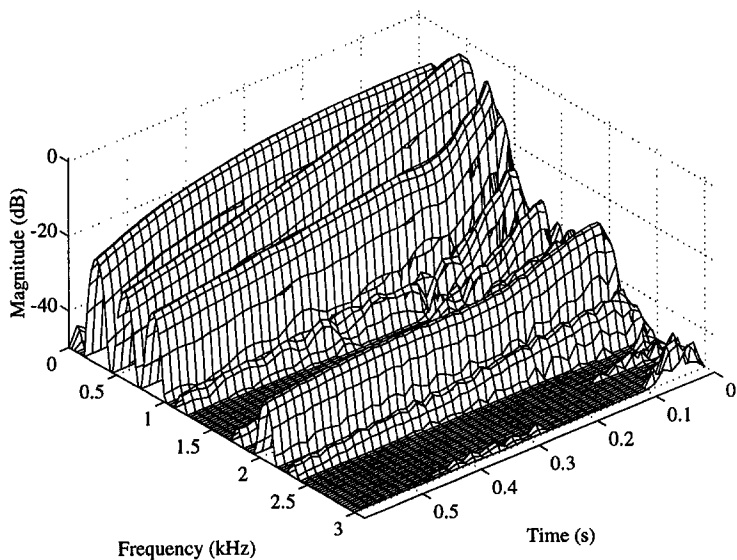


Fig. 2. *STFT analysis of a banjo tone. (Gibson Epiphone, first open string, fundamental frequency 289 Hz).*

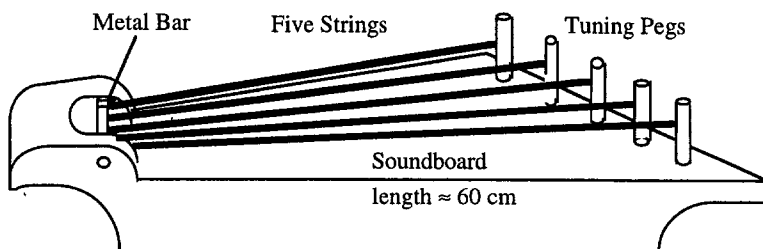


Fig. 3. *The kantele, a traditional Finnish string instrument.*

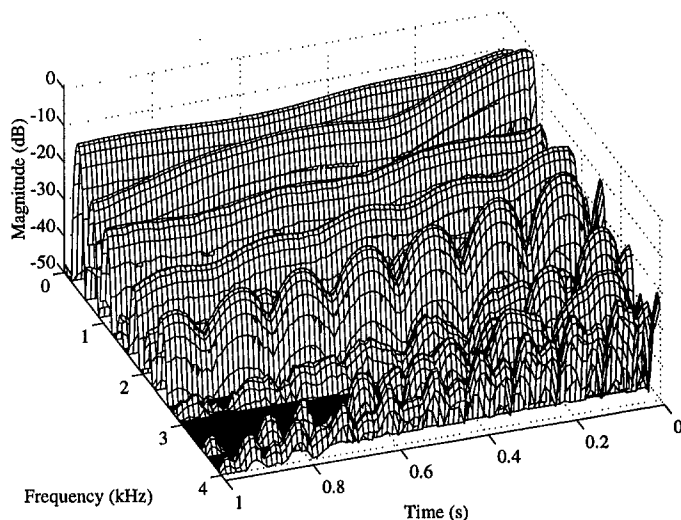
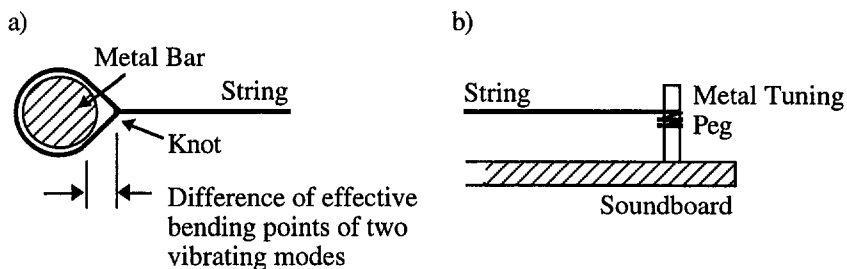


Fig. 5. *STFT analysis of a kantele tone. (fourth open string, fundamental frequency 349 Hz).*

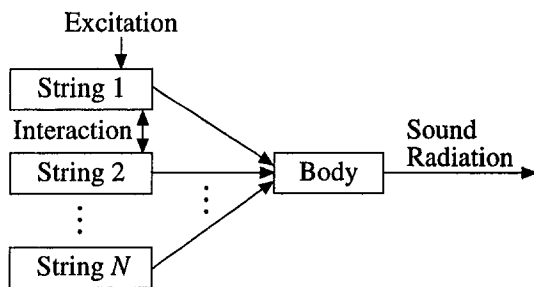


Fig. 6. Block diagram of a general model for a plucked string instrument.

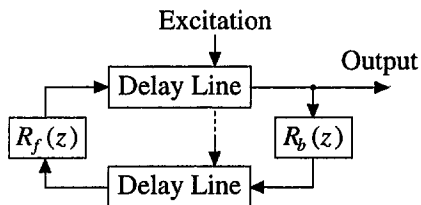


Fig. 7. Waveguide model for a vibrating lossy string.

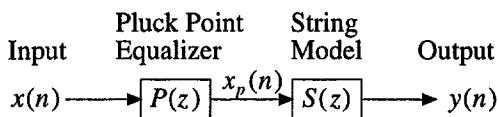


Fig. 8. The generalized Karplus–Strong model consists of a string model $S(z)$ cascaded with a plucking point equalizer $P(z)$.

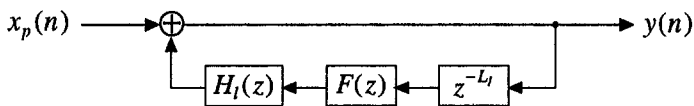


Fig. 9. The block diagram of the string model $S(z)$. The filter $F(z)$ is an FIR or allpass-type fractional delay filter and $H_f(z)$ is the loop filter.

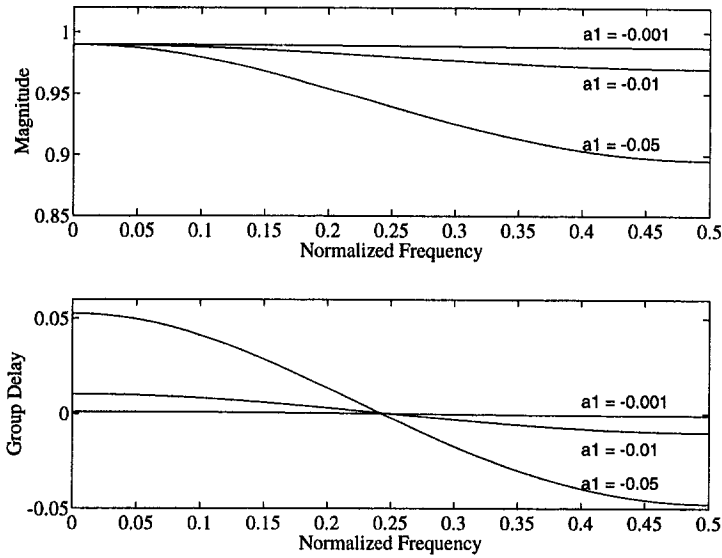


Fig. 10. *Magnitude response and group delay of the one-pole loop filter with three different values of a_1 . The coefficient g is equal to 0.99 in all cases.*

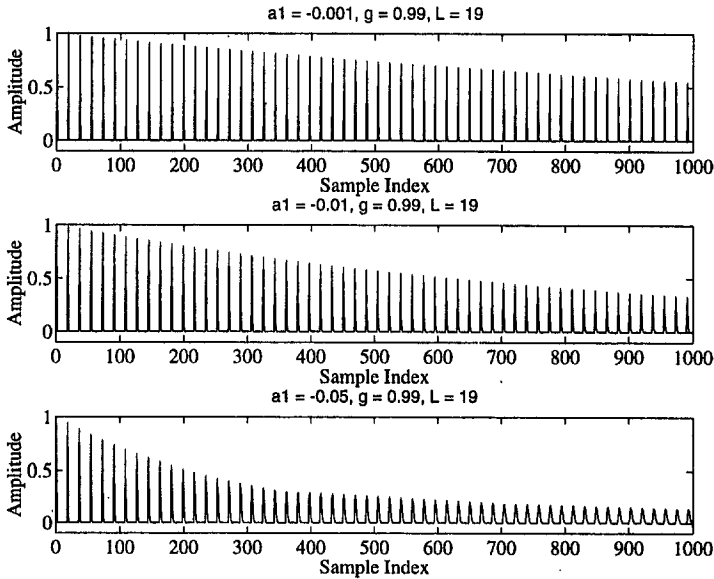


Fig. 11. *Impulse response of the string model obtained using the loop filters of Fig. 10.*

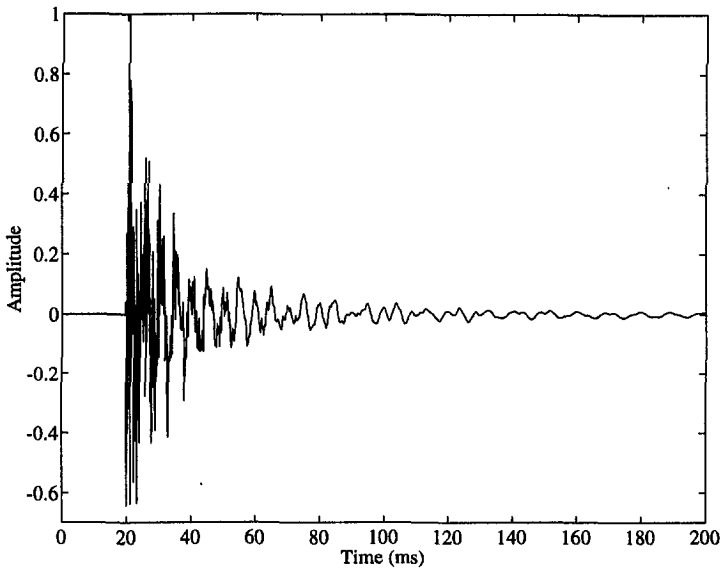


Fig. 12. *Impulse response of the body of a classical acoustic guitar.*

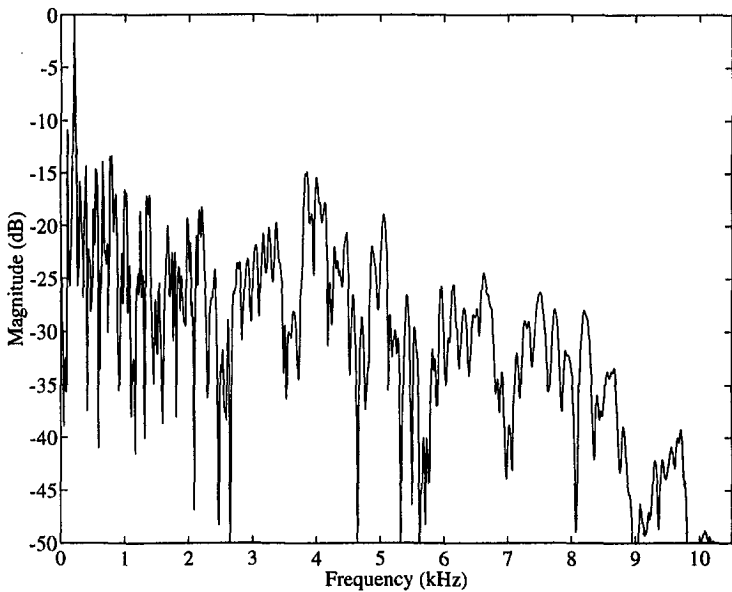


Fig. 13. *Frequency response of the body of a classical acoustic guitar.*

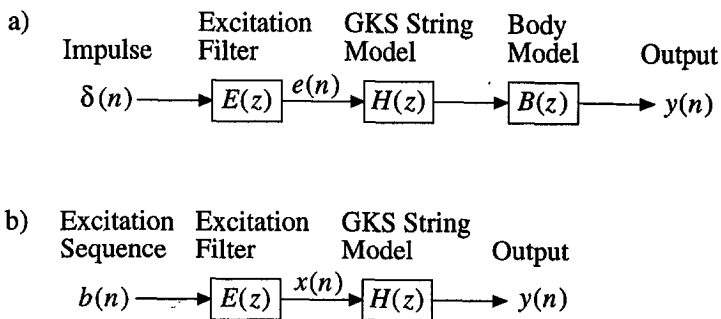


Fig. 14. a) Block diagram of a string instrument model. b) An equivalent model that has been obtained by reordering the systems.

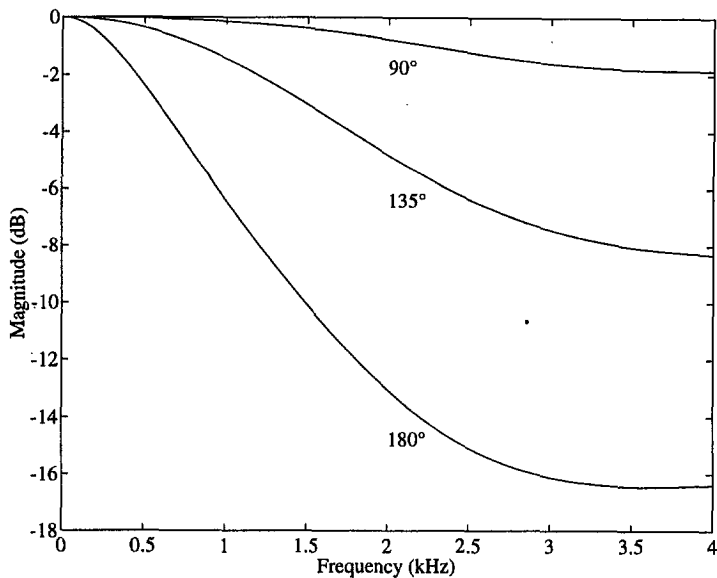
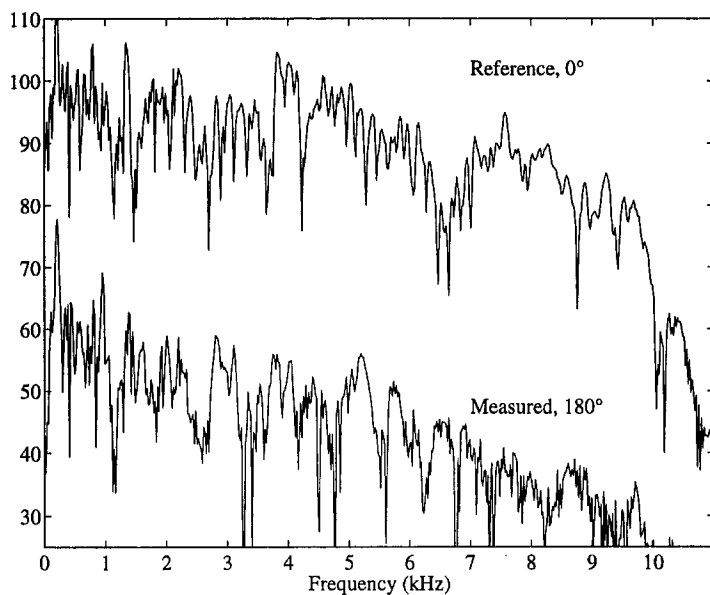


Fig. 15. Modeling the direction-dependent radiation of the acoustic guitar relative to the main axis radiation with a second-order IIR filter.

a)



b)

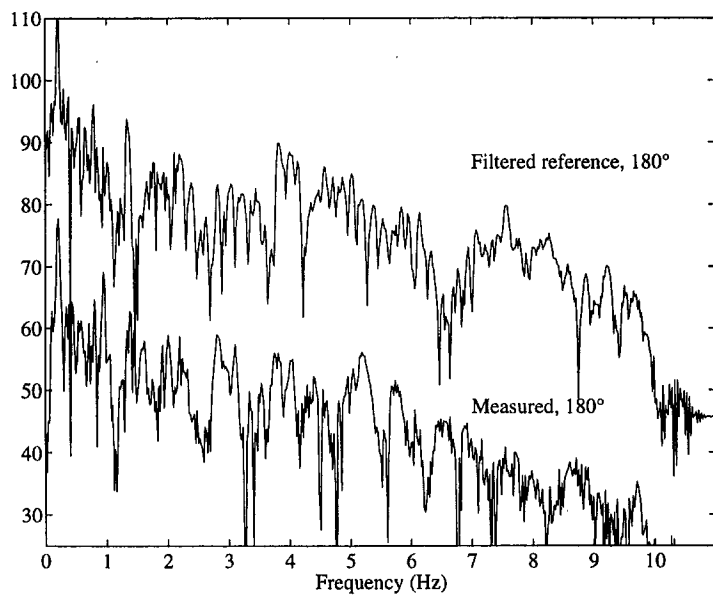


Fig. 16. Modeling of guitar sound radiation with a first-order IIR filter. a) Transfer functions measured at azimuth angles 0° and 180° . b) Response at 0° filtered with a directivity filter and the measured transfer function at 180° .

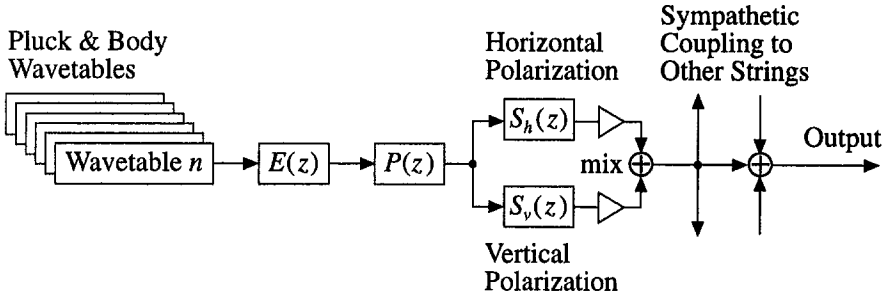


Fig. 17. Overview of the plucked string synthesis model including dual-polarization strings and sympathetic coupling between strings.

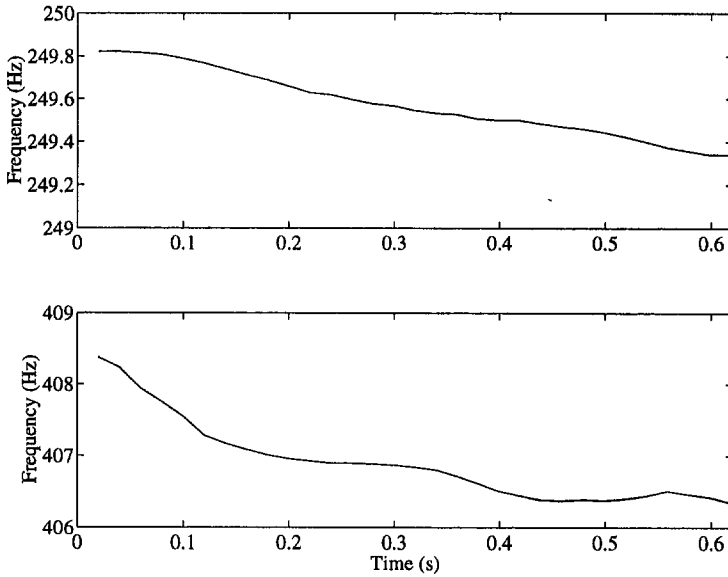


Fig. 18. Fluctuation of the fundamental frequency of a) a steel-string guitar tone, b) a kantele tone.

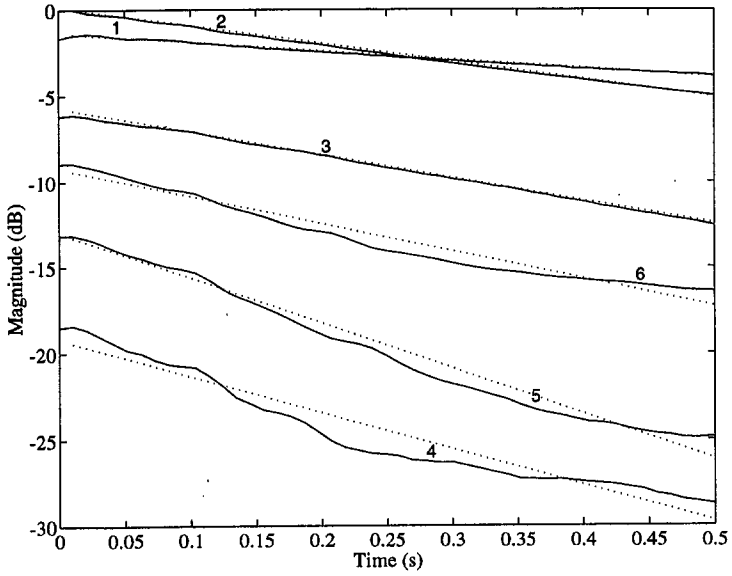


Fig. 19. Straight line fits (dotted lines) of the envelope curves (solid lines) of the six lowest harmonics of an electric guitar tone.

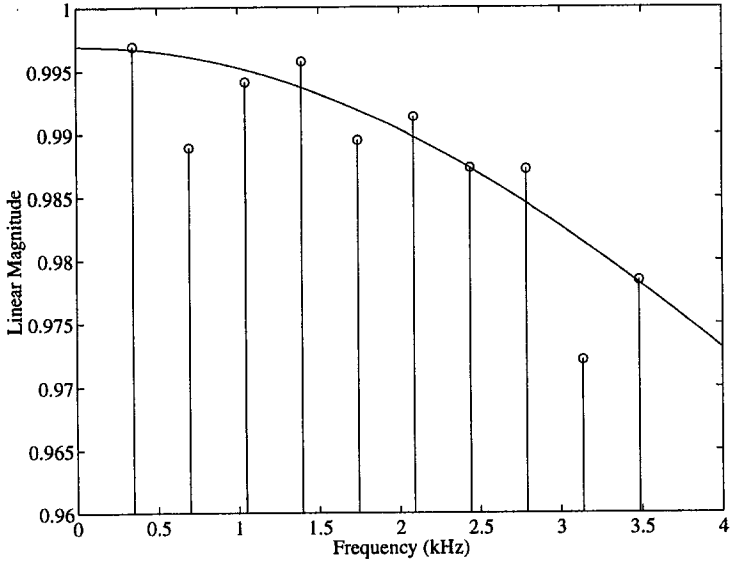


Fig. 20. Loop filter design for the kantele. The circles show the loop gains G_k^2 at harmonic frequencies. The solid line is the magnitude response $|H_1(z)|^2$.

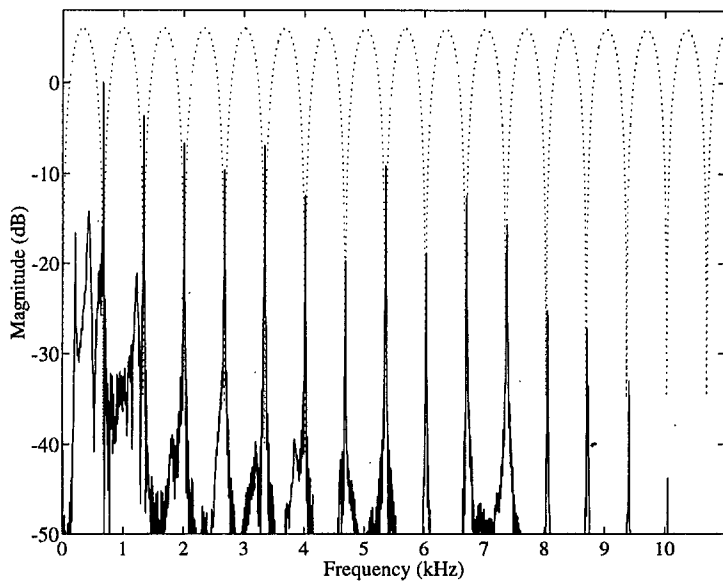


Fig. 21. *Magnitude spectrum of a mandolin tone (solid line) and the magnitude response of the inverse filter (dotted line).*

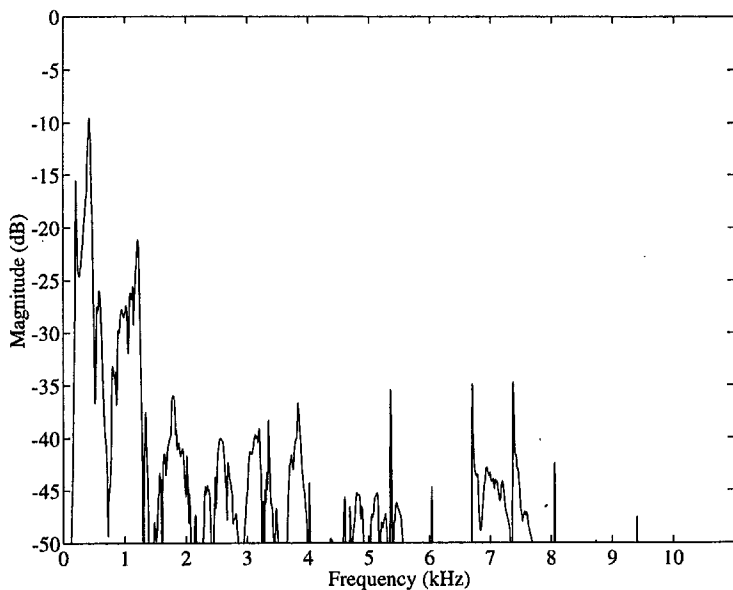


Fig. 22. *The magnitude spectrum of the inverse filtered mandolin tone.*

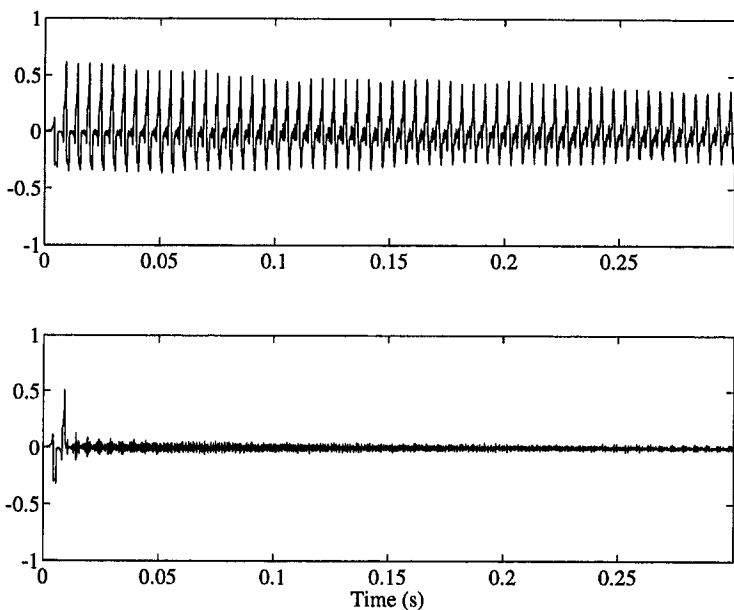


Fig. 23. *a) An electric guitar tone. b) The residual after inverse filtering.*

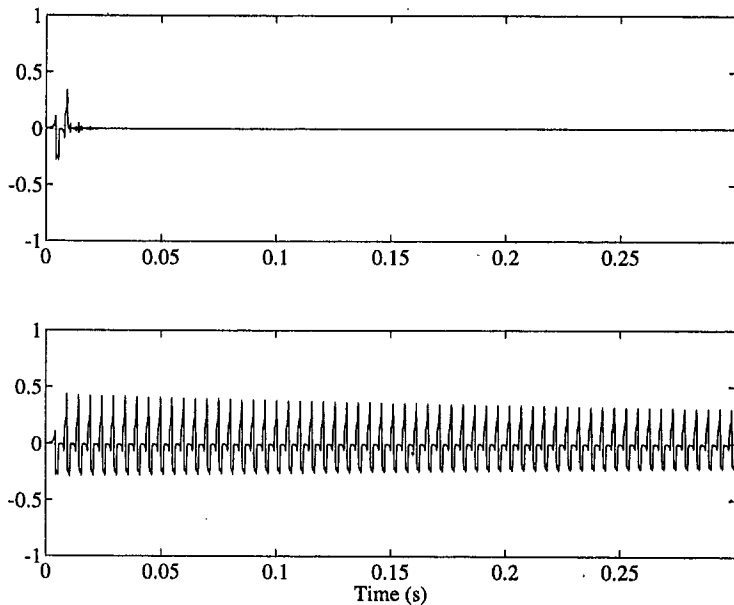


Fig. 24. *a) The truncated residual, b) The resynthesized electric guitar tone.*

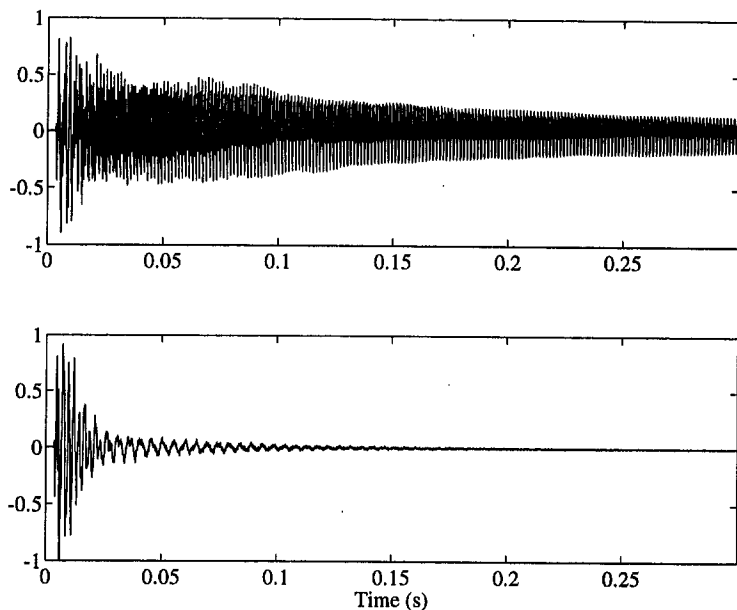


Fig. 25. *a) A mandolin tone, b) the residual signal after inverse filtering.*

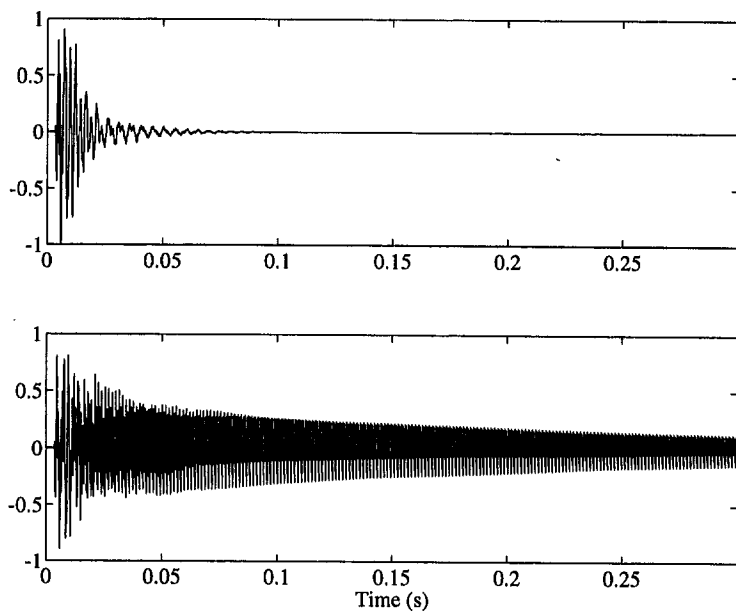


Fig. 26. *a) The truncated residual signal, b) the resynthesized mandolin tone.*

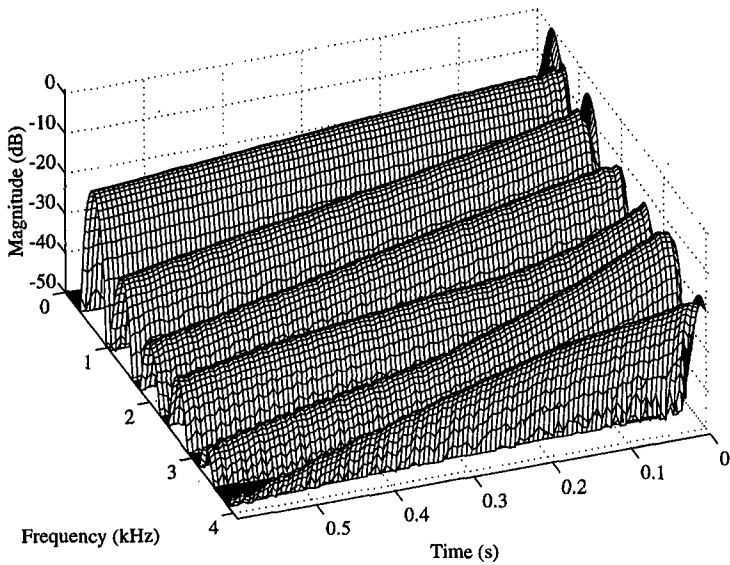


Fig. 27. *STFT analysis of a mandolin tone. (Gibson model A, first open single string, fundamental frequency 649 Hz).*

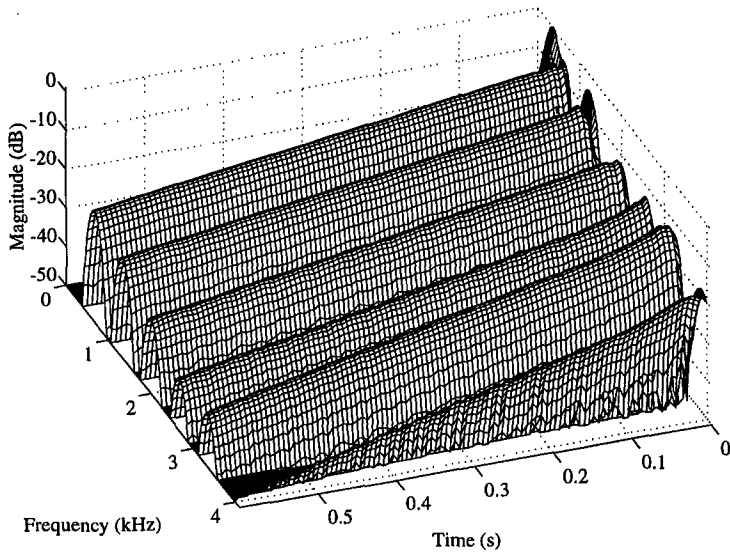


Fig. 28. *STFT analysis of a resynthesized mandolin tone.*

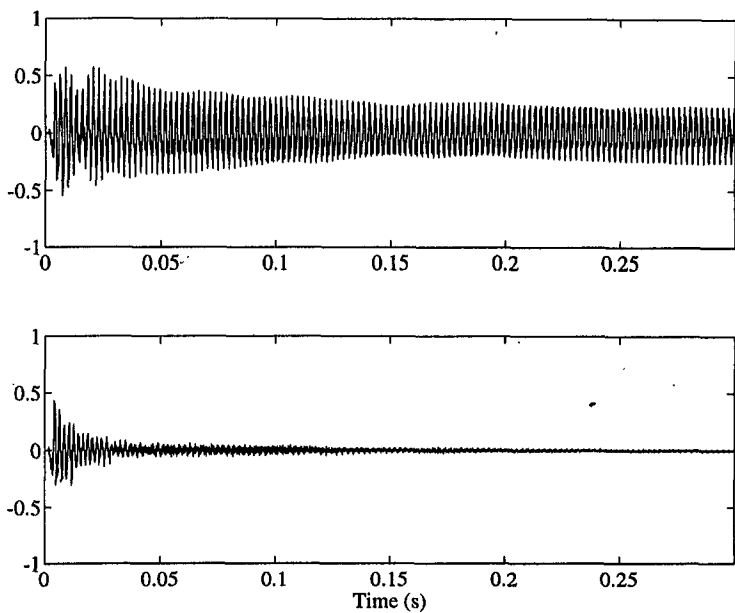


Fig. 29. *a) An original kantele tone. b) The result after inverse filtering.*

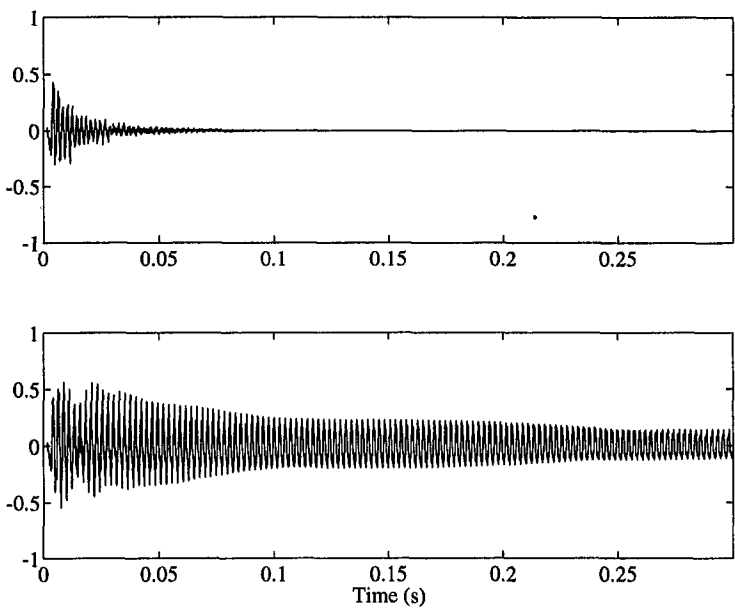


Fig. 30. *a) The truncated residual signal. b) The resynthesized kantele tone.*

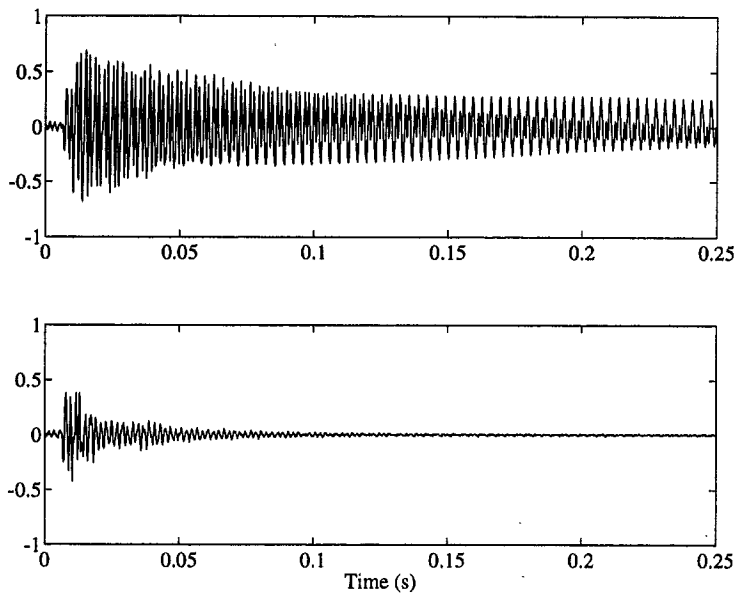


Fig. 31. *a) An original banjo tone. b) The result after inverse filtering.*

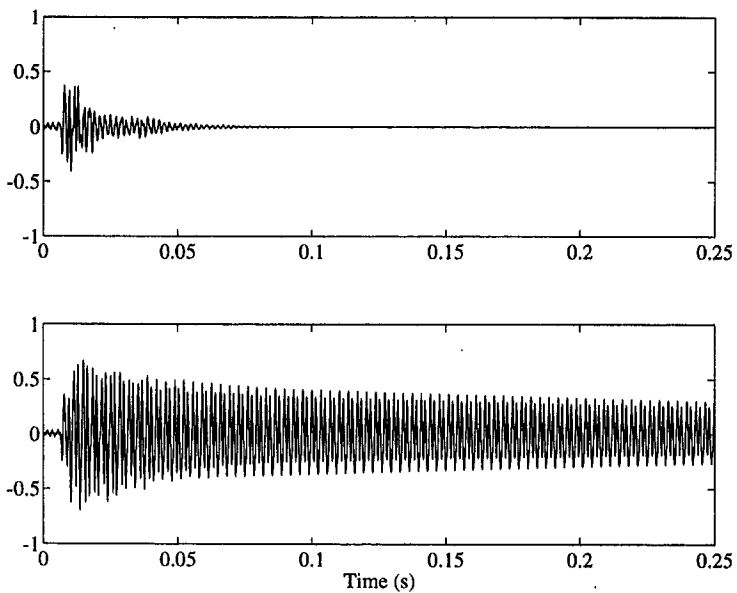


Fig. 32. *a) The truncated residual signal. b) The resynthesized banjo tone.*

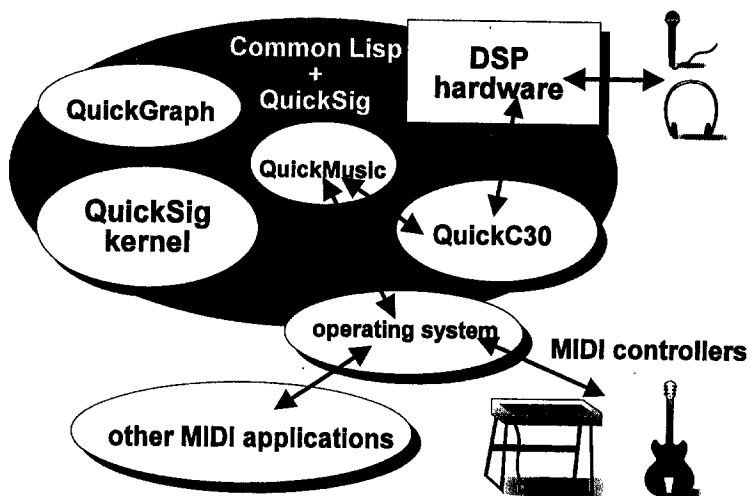


Fig. 33. The software and hardware environment for real-time implementation of physical models.

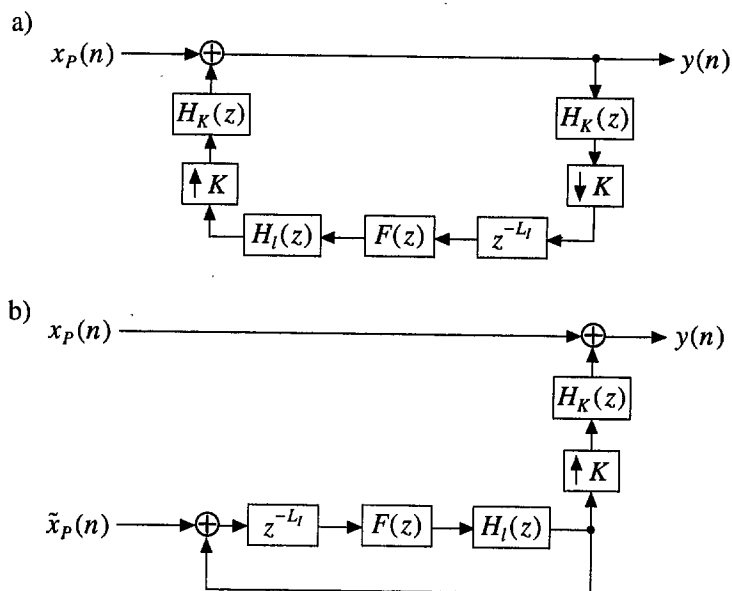


Fig. 34. a) Multirate synthesis model where the feedback loop is computed at a lower sample rate f_s / K where K is the decimation factor. b) An equivalent system where the original input signal $x_p(n)$ is fed directly to the output and a decimated input signal $\tilde{x}_p(n)$ is used in the feedback loop.