Towards High-Quality Sound Synthesis of the Guitar and String Instruments

Matti Karjalainen[1,2], Vesa Välimäki[1,2], and Zoltán Jánosy[1,2,3]

[1]Helsinki University of Technology, Acoustics Laboratory
Otakaari 5A, SF-02150 Espoo, Finland

[2]CARTES (Computer Arts Center at Espoo)
Ahertajankuja 4, SF-02100 Espoo, Finland

[3]Technical University of Budapest, Department of Telecommunications
Sztoczek u. 2., H-1111 Budapest, Hungary

matti.karjalainen@hut.fi, vpv@vipunen.hut.fi, janosy@tsys.hit.bme.hu

Abstract

The sound quality of real-time synthesis based on physical models has so far been inferior to sampling techniques. In this paper we introduce new principles to make model-based sound synthesis of the guitar and other plucked string instruments more attractive from the viewpoint of sound quality. A major improvement is achieved by estimating the model parameters and the excitation signal from the sound of an acoustic instrument. It is shown that the impulse response of the body is included in this excitation. More complex string behavior, including nonlinearities in some instruments, is briefly studied. Furthermore, different aspects of controlling the real-time synthesis model are discussed. High-quality real-time synthesis is shown to be feasible by using a single digital signal processor.

1 Introduction

The term physical modeling is often used for computational models of acoustic-mechanical principles found in musical instruments. By means of physical models it has been possible to simulate quite detailed effects of sound generation [McIntyre *et al.*, 1983]. It has also been shown that remarkable reductions in computation may be gained, e.g., by digital waveguide techniques [Smith, 1987] that allow for real-time synthesis on modern signal processors [Karjalainen and Laine, 1991]. The sound quality of isolated sounds has so far remained inferior to sample-based methods. However, model-based synthesis has practical advantages such as a natural set of control parameters that allow wide variations of the synthesized sound. As an example, so-called sympathetic vibrations of strings are difficult to simulate in detail by sampling and other traditional methods.

This paper introduces new ways to greatly improve the sound quality of model-based real-time synthesis of the guitar and plucked string instruments. Such an approach is based on balancing three major sources of knowledge: understanding the acoustic principles (physical modeling), digital signal processing expertise, and taking into account human perception.

First an introduction to the modeling of a plucked string instrument is given. Models for the strings and the body as well as their interactions are discussed. Examples of complex string behavior are reported as they have been found in some instruments. In Section 3 the estimation of the model parameters is studied as a means to add reality to the sound. Finally, in Section 4 solutions to the problem of controlling the real-time synthesis model are presented.

2 String Instrument Model

The main elements and intercouplings found in most plucked string instruments are as shown in Fig. 1. Each string is a distributed subsystem that starts to vibrate when excited (plucked or picked). The strings are coupled to the body and may also interact with each other (sympathetic vibrations). The body or a soundboard is a complicated resonator that is needed for acoustic amplification, sound radiation, and coloring of the sound.

2.1 Modeling of the String

The general solution of the wave equation for a string is composed of two independent transversal waves traveling in opposite directions (see e.g. [Fletcher and Rossing, 1991]). At the string terminations the waves reflect back with inverted polarity and form standing waves. The losses in the system damp the almost periodic vibration of the string. All losses and other linear non-idealities may be lumped to the termination and excitation or pickup points. The string itself is then described as an ideal lossless waveguide [Smith, 1993]. The system may be modeled by a pair of delay lines and a pair of termination filters as illustrated in Fig. 2.

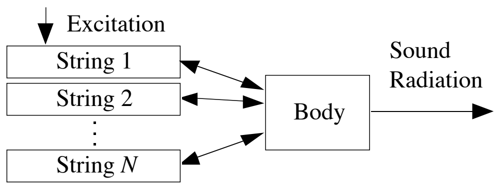

Fig. 1 *Model for a plucked string instrument.*

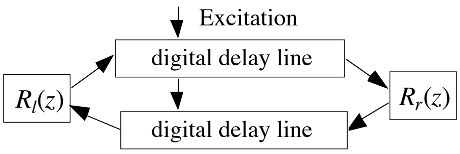

Fig. 2 *Digital waveguide model for a string.*

A practical implementation is a digital waveguide with two digital filters which may often be combined into a single one—called later the *loop filter*—and optional excitation and pickup filters. The lossless delay line in a waveguide filter can be implemented very efficiently by a circular buffer. This reduces the computational load by several orders of magnitude.

An ideal pluck may most conveniently be considered as an acceleration impulse to the excitation point, one half of which travels in each direction [Smith, 1982]. Alternative variables for the wave signals (and related excitations) are velocity (unit step), displacement (triangular wave), slope (= the spatial derivative of the displacement), and force. The output may be taken from any meaningful point of the waveguide, e.g., by summing (or differentiating) the velocities from the two delay lines. The radiated sound pressure is approximately proportional to the velocity (integral of the acceleration) of the string pickup point.

As shown by Jaffe and Smith[1983], there is a relation between the model of Fig. 2 and the Karplus-Strong (KS) model [Karplus and Strong, 1983] (see Fig. 3b). Originally excited by a sequence of random numbers, the KS filter formulation corresponds to the model of Fig. 2 if a comb filter (see Fig. 3a) is cascaded to shape the spectrum due to the effect of plucking position. The output of an electric guitar pick-up can be simulated in the same manner [Sullivan, 1990].

2.2 String Model Implementation

An excellent early paper on the implementation of string models was written by Jaffe and Smith[1983]. A relatively detailed model of the guitar with a real-time implementation on a single DSP processor was presented by Karjalainen and Laine[1991]. In the following, we give a brief overview of the latter approach.

Fractional Length Approximation

The implementation of the string may be divided into the design of a waveguide and one or two loop filters. The desired pitch values to be played by an instrument cannot in general be realized by integer-sized delay lines (unless the sampling rate is varied). Fractional delay approximation is thus needed and often the length of a string should be continuously variable during sound synthesis (e.g. for vibrato and glissando effects).

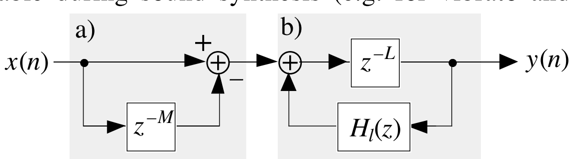

Fig. 3 *a) A plucking-position equalizer cascaded with b) the Karplus-Strong model.*

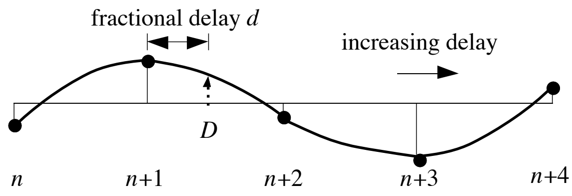

Fig. 4 *Fractional delay approximation.*

Both IIR and FIR types of interpolation filters may be utilized. A comprehensive guide to fractional delay filter design is given in [Laakso *et al.,* 1993]. The interpolation problem is illustrated in Fig. 4 where a band-limited discrete-time signal is known at integer sample points but should be known at some real-valued point D instead. The ideal interpolator is a shifted and sampled sinc function of infinite length which implies that in practice it is only possible to approximate it.

Lagrange interpolation is a good FIR type of approximation that turns out to be a maximally flat (at zero frequency) filter. Its first-order version (linear interpolation) is used, for example, by Sullivan[1990]. The filter coefficients $h(n)$ for the Lagrange interpolator are expressed as [Laine, 1988]

$$h(n) = \prod_{k=0,k\neq n}^{N} \frac{D-k}{n-k} \quad \text{for } n = 0,1,\ldots,N \qquad (1)$$

where the approximation is computed for a fractional delay $D = \text{floor}(D) + d$, $d \in \Re$, and N is the order of the filter. Advantages of this method are the simple computation of coefficients (for real-time updating) and good suppression of signal transients while the length of the line is varied. The main drawback is that the magnitude response is not flat unless $d = 0$.

An allpass filter with an ideally flat magnitude response is another good alternative. See [Laakso *et al.*, 1992] for details on a maximally flat fractional delay approximation by allpass filters. The price paid to get an ideal magnitude response is a compromised phase response and a more complex update of coefficients. Furthermore, the allpass filter is more prone to signal transients when the delay is varied.

We have used the waveguide principle of Fig. 2 with third-order Lagrange interpolation at a sampling rate of 22 kHz [Karjalainen and Laine, 1991] in order to cover carefully the frequency range up to about 5 kHz. No audible transients are generated in glide sounds. For simplicity, the pluck point as well as the pickup point have been assigned to integer points of the delay line. This has not been a limitation in practice.

Implementation Issues

According to the theory of string vibration (see, e.g., [Fletcher and Rossing, 1991]), the damping due to internal losses, air friction, and end support movement might be approximated by a second-order low-pass filter in the string model loop. We have found that even a single first-order IIR low-pass filter gives relatively good results when the dc-gain and the cutoff frequency are adjusted properly. The values of these parameters depend on the string properties, the fin-

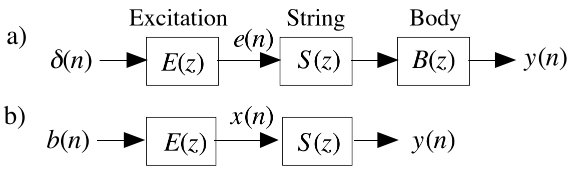

Fig. 5 *a) The string instrument model. b) The modified model that is functionally equivalent to (a).*

gering position, and possible extra attenuation due to a specific playing style.

To control the sharpness vs. softness of the plucking, we have filtered the excitation (an impulse in the simplest case) by a first or second-order low-pass filter with controllable cutoff frequency.

2.3 Body Modeling

The acoustics of the guitar body has been studied both qualitatively and quantitatively (see [Fletcher and Rossing, 1991] for references) but detailed computational models seem to be difficult to construct. From a signal processing point of view the body and the sound radiation to a specific direction may be considered as a high-order filter. In general we need many responses to simulate the directivity pattern, actually a matrix of responses from the main vibrational directions of the bridge to various radiation angles.

Body Model as a Digital Filter

The transfer function from the bridge to the listener can be measured approximately by exciting the bridge with a mechanical impulse and by registering the radiated sound. An analysis shows that the spectral envelope of the body response of a good acoustic guitar is relatively flat but there is a large number of resonances starting from the lowest mode frequency (around 100 Hz). However, not only the frequency-domain magnitude response but also the temporal envelope of the impulse response (“reverberation”) is perceptually important.

We have tried several filter-based principles of body modeling for sound synthesis [Karjalainen *et al.*, 1991]. An FIR filter model of the body response must be 50 to 100 ms (more than 1000 taps) long to yield satisfying 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 on a modern 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.

Body Response as String Excitation

In order to overcome the inherently heavy computational load of filter-based body models, a novel method was invented. Let us consider the string instrument model of Fig. 5a as a chain of linear subsystems

$$y(n) = e(n) * s(n) * b(n) \tag{2}$$

where $*$ is the convolution operator, $e(n)$ is the excitation source (pluck in the case of the guitar), $s(n)$ is the impulse response of the string (from plucking point to the bridge) and $b(n)$ is the impulse response of the body model (from the bridge to the radiated sound). The input $\delta(n)$ in Fig. 5a is a unit impulse.

The system of Fig. 5a can be transformed into the form of Fig. 5b by reordering. This modification is valid due to the mathematical fact that the convolution operation is commutative, i.e.,

$$e(n) * s(n) * b(n) \equiv b(n) * e(n) * s(n) \tag{3}$$

Now, if the body response $b(n)$ is time-invariant or its various forms can be represented (approximated) by wavetable(s) it is possible to avoid one convolution of Fig. 5a. This is achieved by precomputing, measuring, or estimating the impulse response $b(n)$ of the body and storing it as a wavetable that can be read out sequentially after each excitation event (plucking). This reduces the computation by several orders of magnitude. The original body model convolution requires multiply-add operations on the order of N or $\log_2 N$ (using fast convolution) per output sample, where the length of the impulse response N is more than 1000. In contrast, wavetable synthesis requires only one read operation per sample. This makes it possible to generate high-quality synthetic sounds of the acoustic guitar in real time on a signal processor such as the TMS320C30.

2.4 Sympathetic Vibrations

The importance of sympathetic string vibrations (i.e. the excitation of some harmonics of a string by the vibration of other strings) to the quality of model-based synthesis has been discussed, e.g., by Jaffe and Smith[1983]. In our guitar model sympathetic couplings were implemented by simply feeding a small fraction of a string output to other strings at the bridge position. Although this is an oversimplification that does not model the complicated frequency-dependent couplings of a real instrument the result is quite satisfactory and makes the guitar synthesis sound more realistic. This scheme is somewhat critical to the coupling coefficients since there is feedback via strings and excessive coupling can make the system unstable.

2.5 Complex String Behavior

Many simplifications were inherent in the string models presented above. In the following, three special cases of more complex behavior and the related modeling solutions are discussed.

Modeling the Plucking Contact

An assumption in the string models above was that the string vibrates autonomously and the excitation is superposed to the waves traveling along it. In reality the plucking contact involves more complex even nonlinear interactions.

A new idea to model this efficiently is to use a fractional delay three-port [Välimäki *et al.*, 1993*a*] at the plucking position of the string waveguide (see Fig. 2). Such a port is composed of a fractional delay interpolation out of the delay lines, reflection and interaction calculations, and deinterpolation back to the delay lines. This method has been successfully ap-

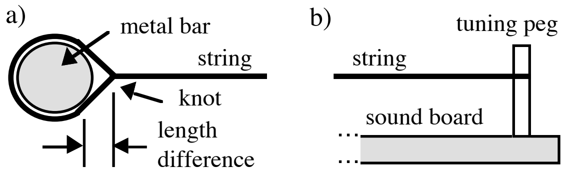

Fig. 6 *Kantele string terminations: a) knot termination and b) support around tuning peg.*

plied to the implementation of finger holes in woodwind instrument models [Välimäki *et al.*, 1993*b*].

Beats due to Double-Length Behavior

The model-based approach described above may be applied to the synthesis of many other plucked string instruments such as the lute, the banjo, or the mandolin. Each string instrument, however, exhibits its unique features. Here we discuss two specific effects that have been found in an old Finnish instrument, the kantele [Karjalainen *et al.*, 1993]. Although of minor importance in the guitar they should be included in a detailed model of any string instrument.

The traditional kantele is equipped with five strings and a body or soundboard. The termination of the strings is special: at one end the string is wound around a metal bar and is fixed by a knot (see Fig. 6a). Thus the effective length of the string is different (by 0.1-0.2%) in the two main planes of vibration. Due to this a strong beat is introduced when the vibrations in the two planes are summed up in the body. Differences of the driving-point impedance of the end supports (e.g. the bridge) in the different planes may cause similar effects also in the guitar. It is also known, for example, that the decay rate of vibration depends on the plane of vibration [Fletcher and Rossing, 1991].

A simple solution to the modeling of these effects is to use two digital waveguides, one for each vibration plane.

Nonlinearity due to Longitudinal Forces

The tension of a string changes along with transverse displacements in a nonlinear way so that new partials may be generated if this longitudinal force can pass to the body. In the kantele this effect is very prominent since there is no bridge and the string is directly supported by the tuning peg (Fig. 6b) so that the tension variation is transferred to the soundboard by peg bending. The following relation has been derived in [Karjalainen *et al.*, 1993] for the time-varying longitudinal force f

$$f(t) \sim \int_0^l \left[s_y^2(x,t) + s_z^2(x,t) \right] dx \qquad (4)$$

where $s_y(x)$ and $s_z(x)$ are the slopes in the two main planes (y,z) of the string vibration in position x, and l is the length of the string.

This case is taken as an example where substantial difficulties arise in keeping the string waveguide ideal since the force is proportional to the integral of the string slope squared. The distributed nonlinearity is computationally very expensive so that a simple localized approximation is needed for real-time synthesis [Karjalainen *et al.*, 1993].

3 Estimation of Model Parameters

The problems in sound quality of former physical models of plucked string instruments were caused by too simple excitation signals, loop filters, and body models. In most waveguide synthesis models the body has not been considered at all. The input to the model has been either white noise [Karplus and Strong, 1983] or an impulse filtered with a low-order filter, and the coefficients of the loop filter have been adjusted by hand [Adrien and Rodet, 1985], [Karjalainen and Laine, 1991]. More complicated excitation signals and loop filters are difficult to devise without help of measurements. Thus the attempt to estimate the string model from the sound of a real instrument is well motivated.

3.1 Subproblems in Estimation

The system identification of the waveguide string instrument model shown in Fig. 5a can be divided into the following subproblems: (1) Estimation of the body model $B(z)$, (2) estimation of the string model $S(z)$ which leads to the estimation of the loop filter $H_l(z)$ and the delay length L when the Karplus-Strong model is employed, and (3) estimation of an excitation sequence $e(n)$ or filter $E(z)$.

When using the modified string instrument model of Fig. 5b the estimation problem is simplified. Now only the loop filter $H_l(z)$ and the delay L have to be estimated. Once they are available, the excitation signal $x(n)$ can be extracted by *inverse filtering* the recorded guitar sound. The inverse filter $A(z)$ can be solved by inverting the transfer function of the string model $S(z)$:

$$A(z) = \frac{1}{S(z)} = 1 - H_l(z)z^{-L} \qquad (5)$$

where the delay z^{-L} in the denominator of the last form has been omitted. Extraction of the excitation signal for the KS model using inverse filtering has also been proposed by Laroche and Jot[1992].

The residual produced by the inverse filter is a short burst that corresponds to the combination of the pluck sound, the impulse response of the body, and a prediction error due to defects in the string model. Typically, the residual dies away with a time constant of about 50 ms. In practice the excitation signal $x(n)$ is formed by truncating about 100 ms from the very beginning of the residual so that the transient part of the guitar sound is included.

3.2 Estimation of the Loop Filter

The problem of calibrating the loop filter $H_l(z)$ according to a recorded sound of a vibrating string was carefully studied by Smith in the beginning of 1980's (see [Smith, 1982] and [Smith, 1983]). He reported results attained by using a system identification approach and modified linear prediction. The former method results in an IIR loop filter and the latter in an FIR filter. The methods were shown to perform well at low frequencies but the variance of the estimates was very large at the high end. Our experiments have shown that these methods often lead to unstable loop filters which cannot be used for synthesis unless made stable.

Deconvolution

In [Smith, 1982] it was also proposed that the frequency response of the loop filter could in principle be computed using deconvolution. In the frequency-domain deconvolution means division of spectra. This is equivalent to multiplication where the magnitude values of the other spectrum have been inverted. In this operation large values become small and small values enormously large, and as a result the noise is amplified. Smith[1982] reports that this technique can yield extremely noisy estimates for the magnitude response of the loop filter.

Instead of using the deconvolution of two spectra as the estimate for the frequency response, we have tried averaging over several of them. The estimate for the spectrum of the loop filter is then expressed as

$$\hat{H}_l(e^{j\omega}) = \frac{1}{M}\sum_{m=1}^{M-1}\frac{Y(e^{j\omega},t_m+P)}{Y(e^{j\omega},t_m)} \qquad (6)$$

where the terms $Y(e^{j\omega},t_m)$ are the windowed Fourier spectra computed at the instant t_m, M is the number of these spectra, and P is the period-length of the guitar sound. This method gives estimates with reasonably small variance at low frequencies. However, the results at high frequencies are fully unreliable.

A Robust Algorithm

We introduce a new, more robust technique for the estimation of the magnitude response of the loop filter. This straightforward algorithm consists of the following steps:

1) Compute the short-time Fourier transform (STFT) of the guitar sound to be resynthesized;
2) Measure the magnitude of each detectable harmonic in the STFT frames and form a sampled envelope curve for every harmonic;
3) Fit a straight line on a logarithmic (dB) amplitude scale to each sampled envelope curve;
4) Compute the corresponding loop gain for each slope;
5) Design a digital filter to match the magnitude spectrum that is formed by the collection of loop gain estimates at different frequencies.

In Step 1 we have computed the STFT (using FFT) using a Blackman window with 50% or 75% overlap with the adjoining windows. Pre-emphasis (e.g., differentiation) should be applied to the original signal before the analysis to flatten its spectrum. It is wise to use zero-padding in the FFT to increase the resolution of the magnitude response.

The amplitude of the overtones can be measured by finding the highest amplitude value near the assumed frequency of the harmonic and by using *parabolic interpolation* to find the peak value [Serra, 1989]. The analysis should be started only after the attack transient of the guitar sound has died, i.e., after the envelope of the recorded signal shows exponential decay. The higher harmonics of a guitar sound attenuate quickly and after a while they can not be detected because of the background noise. For this reason, the measurement of harmonics can be terminated some 200-500 ms after the attack.

In Step 4 the gain g of the loop filter (at the frequency of the overtone in question) can be computed as

$$g(f_k) = 10^{\frac{\beta_k L}{20H}} \text{ for } k = 1, 2, \ldots, K \qquad (7)$$

where β_k is the estimated slope of the kth harmonic, f_k the frequency at which it occurs, H the hop size used in the STFT analysis, K the number of harmonics to be extracted, and L the length of the delay line of the string model. The length L is determined as the ratio of the sampling frequency to the fundamental frequency of the analyzed sound and it is assumed to be real-valued.

Finally, the loop filter can be designed using the resulting magnitude response as a prototype. Here we assume that the slightly inharmonic nature of the guitar sound is not perceptually as relevant a feature as the attenuation rate of the harmonics. If the dispersion in the string were tried to be modeled as well the frequency of each harmonic should be accurately measured and the phase should be accounted for in the filter design.

Analysis of Decay Rates of Harmonics

It appears that, disregarding the attack, the sound of a classical acoustic guitar (plucked by finger) includes an insignificant amount of energy at the frequencies above 3 or 4 kHz. Due to this observation, it is not important to try to estimate the magnitude response of the loop filter at much higher frequencies. The brightness will be provided to the synthesized guitar sounds by the excitation signal which includes the attack transient of a real pluck. At high frequencies the loop filter is only designed to have sufficient amount of damping so that the high-frequency components included in the excitation will die out rapidly.

In Fig. 7 an example of the time envelopes of the four lowest harmonics of a guitar tone are displayed. Straight lines are fitted to these curves in a least squares sense. It is seen that in this case the fitting succeeds fairly well for most harmonics. In certain tones the string and body can interact in a complex way causing some envelopes to be bumpy. The strangely oscillating envelope of the third harmonic in Fig. 7 is an example of this behavior.

We want to point out that it is our aim to design a low-order loop filter for our real-time synthesis model. For this reason the magnitude spectrum should preferably be relatively smooth, i.e., adjoining harmonics should not have very different slopes since the magnitude response of a low-order filter can not fit the corresponding spectrum. It is thus recommended to choose well-behaving tones for calibration of the model.

The values of the loop gain at the frequencies of the 11 lowest harmonics of a guitar tone (B string, 7th fret) are illustrated as circles in Fig. 8. Typically these gains show the low-pass nature of the string damping, i.e., the loop gain decreases as the

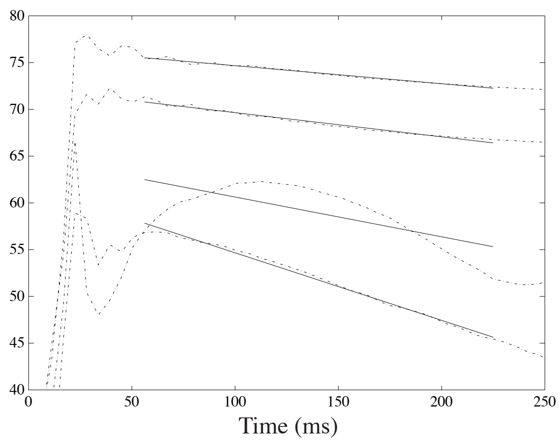

Fig. 7 *Temporal envelopes of the four lowest harmonics of a guitar tone and straight lines fits. The amplitude scale is in dB.*

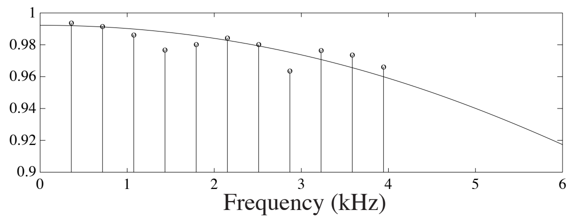

Fig. 8 *Estimated magnitude spectrum (circles) and magnitude response of a 1st-order IIR filter.*

frequency increases. As can be seen in Fig. 8 the practical situation deviates slightly from this simple assumption.

In this example the envelopes of the higher harmonics were extremely noisy because the relative level of those harmonics was very low. Thus the corresponding loop gains could not be estimated.

3.3 Loop Filter Design

The final step in the model estimation procedure is to design a loop filter to match the estimated magnitude response (see Fig. 8). It would be easiest to design an FIR loop filter. Then the filter could also have a linear phase response, if necessary. In practice it has been noticed that a fairly high-order FIR filter is required to accurately model the frequency-dependent damping of the string. Therefore we decided to use a recursive filter instead.

All-Pole Modeling using LPC

As a first method we matched an all-pole model using linear prediction (LPC) to the power spectrum of the loop gain. A practical problem in this procedure is that the target spectrum is known only in a small set of points at the low end of the frequency band and the LPC analysis is seriously disturbed if the magnitude response is set to zero in the rest of the band. A solution to this inconvenience is to assume exponential attenuation for the power spectrum at the high end. The exponent should be proportional to the order of the all-pole filter to be fitted. Otherwise this kind of trick will increase the approximation error at the important frequency band. It can be concluded that the LPC method is very critical and successful

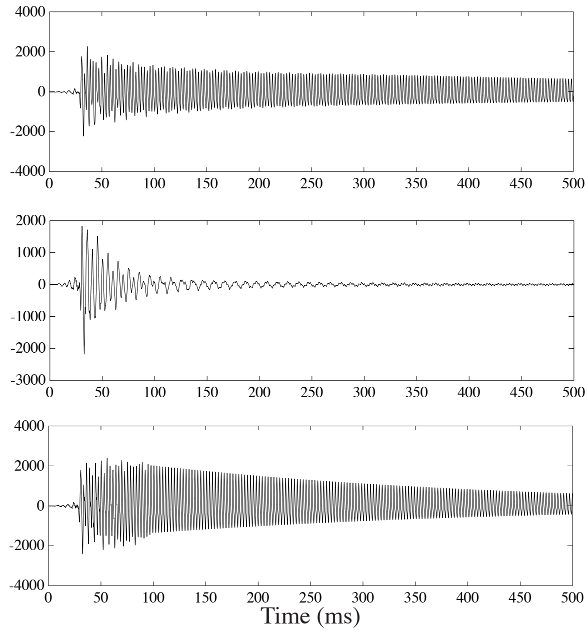

Fig. 9 *a) Original guitar tone, b) the inverse filtered signal, and c) the resynthesized signal.*

results can be obtained only with careful manual adjustment.

Iterative IIR Filter Design

The most efficient structure for the loop filter is obviously an infinite impulse response (IIR) filter with both poles and zeros. As an example we designed a first-order IIR filter for the case of Fig. 8 using an iterative algorithm that alternately adjusts the filter coefficients to reach the minimum of the approximation error. The magnitude response of an IIR loop filter matched to the given points is shown in Fig. 8. In this example the loop filter is

$$H_l(z) = \frac{0.9152 + 0.1889z^{-1}}{1 + 0.1127z^{-1}} \qquad (8)$$

It can be seen that such a low-order filter is not able to model the gain of each harmonic but merely has the same general shape as the desired spectrum.

In the design of this filter we used an error weighting function that penalizes the errors at the points near unity. This guarantees that the resulting magnitude response will not exceed unity and that the match will be best for the lowest harmonics whose attenuation rate can be heard easily. Due to the lack of reliable gain estimates at high frequencies we used zero values in the target spectrum to bring about the low-pass behavior.

3.4 Resynthesis

The waveform of the guitar tone analyzed in Fig. 7, its residual after inverse filtering, and the resynthesized version are shown in Fig. 9. The loop filter of Eq. (8) was used for this example. The excitation signal in the resynthesis was formed by truncating the first 100 ms of the residual of Fig. 9b. It can be seen that the attack of the resynthesized signal is nearly identical with the original one. The resynthesized tone sounds definitely like the guitar but when compared with the original it can be perceived that it attenuates a little more rapidly. Nevertheless this approach results in much more realistic synthesis than the earlier models that used artificial excitation. Naturally the resynthesis can still be enhanced by a more carefully designed or higher order loop filter.

Although in principle a different loop filter and excitation should be used for each tone we tried synthesis using the same excitation and loop filter but a different delay line length and found the result convincing. This means that only a relatively small number of excitation signals need be stored. This is an advantage of the model-based guitar synthesizer over sampling synthesis where a large amount of sampled data has to be stored to achieve good sound quality on a wide playing range.

4 Control of the Model

The current model of the guitar has a large number of parameters many of which must be updated every time a note is played. In most situations it is impractical to control all of them directly but to send basic control information only and compute the unsupplied values automatically. At this stage the current musical context can be also taken into account. Furthermore, the physical model of a human performer (e.g. limited speed of finger movement) should be included to achieve realistic results.

In the following we discuss some methods of controlling the real-time synthesis model. The control interfaces used here translate the control input (e.g. notes) into model parameters (see Fig. 10).

4.1 Control Situations

In a usual live performance situation the control interface has to respond immediately to an incoming control event. This implies that it cannot take future events into account which leads to poor control of note transitions. If strictly real-time performance is not needed introducing a reasonable *processing delay* solves this problem. Furthermore, in guitar synthesis control this method makes optimal string allocation possible.

Using off-line preprocessing of the input the musical context can be analyzed in detail and an internal representation or a new MIDI sequence can be generated. If the input is a music description language the synthesis can be controlled explicitly by including special information into the score. However, most parameters can be adjusted automatically in this case as well.

4.2 Control Interfaces

We added various control interfaces to our new model in order to try it in both real-time and off-line situations. The interfaces are written in Common Lisp / CLOS. Built upon a set of low-level functions (used for updating the model parameters) we implemented an object-oriented sequencer which allows for automatic performance generation (see also [Friberg, 1991] or [Bresin *et al.*, 1992]). For MIDI access from Lisp, we use Hyperlisp [Chung, 1992].

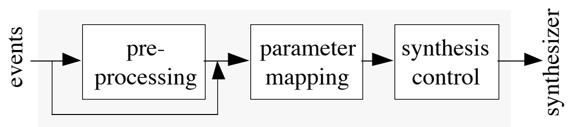

Fig. 10 *Internal structure of the control interface.*

The Lisp Sequencer

The input of the Lisp sequencer is a score description language with Lisp syntax that enables mixing of high-level (notes, chords, sequences) and medium or low-level control information (like string used for the note, pluck type, the timing of the arpeggios, or the binding of successive notes).

The sequence of notes is transformed into a low-level control sequence in four passes: parse, perform, expand and play. Each pass can be explicitly controlled by supplying specialized functions.

MIDI Control

MIDI events are processed in real time by calling CLOS method functions specialized both for the event type and the instrument class. For additional flexibility each string can have method functions of its own that are called after a note assignment by the guitar object.

For a more realistic live performance using a MIDI keyboard we developed a new concept similar to [Garton, 1992]. The MIDI interpreter provides different *performance styles* (e.g. classical, flamenco, blues, pop, or jazz). In addition to the appropriate timbre these presets provide most of the characteristic playing techniques of the given style, like different strokes, strums, hammer-on, pull-off, slide, tremolo, harmonics, pluck position change, mapped to keys, key combinations or controllers, all easily playable from a keyboard. The styles are used together with *enhanced MIDI control modes*. In Solo mode the interpreter can assume that only a single note is intended to be played at one time, thus certain playing techniques (e.g. bend, slide, hammer-on, or pull-off) can be recognized. The Strumming mode is used for chord playing essential in many styles.

String Allocation

A special problem of controlling the guitar synthesis is the selection of the appropriate string for a given note since on the real guitar a note could usually be played on many strings resulting in a different sound quality. Another consideration is that not all possible note combinations can be played since the strings have different playing ranges (see Fig. 11). For realistic simulation it should be taken into account that the fret range used for the notes of a chord or for successive notes is limited.

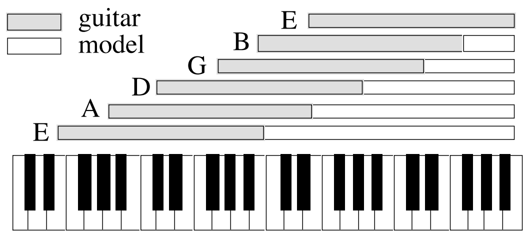

Fig. 11 Playable range of the different strings.

A simple solution is that a note is assigned to the *highest free string* that can play it. From the guitar-player's viewpoint this is a strategy of playing each note as close to the first position as possible. This strategy does not take into account the playability of a given sequence of notes and sometimes it even refuses to play a chord that could otherwise be played in a higher position or in different note order.

If real-time performance is not needed a backtracking algorithm for optimizing the movement of the fingers can be used. This approach makes it also possible to automatically add *synthetic fret-noise* at sudden position changes.

5 Summary

Improvements and extensions to earlier physical models of plucked strings were introduced. The main contribution of this work is that more natural synthetic sounds than before which imitate the acoustic guitar can be produced in real time. This was achieved by interpreting the guitar as a cascade of linear subsystems: the excitation, the string, and the body. Thereafter the parts in the chain were reordered in order to use the body response as the input. The input can be, e.g., a measured body response or a signal obtained by inverse filtering a recorded guitar sound.

We also discussed how to estimate a digital filter that models the frequency-dependent attenuation in the delay loop of a waveguide string model. Modeling of more complex behavior of the string, like double-length behavior and nonlinearities, was studied.

We have implemented the described guitar model on a single TMS320C30 signal processor using the QuickC30 software environment [Karjalainen, 1992] running on an Apple Macintosh computer. This DSP system can run a six-string guitar model in real time at a sampling rate of 22.05 kHz. The synthesis model can be controlled either from MIDI or by a special Lisp-based sequencer.

Acknowledgments

We are grateful to Mr. Toomas Altosaar for his comments on an earlier version of this paper. Special thanks are due to Mr. Jukka Savijoki for his kind co-operation in recording high-quality guitar sounds. This work was supported by the Academy of Finland.

References

[Adrien and Rodet, 1985] Jean-Marie Adrien and Xavier Rodet. Physical models of instruments: A modular approach, application to strings. In *Proc. ICMC'85*, pp. 85-89, Vancouver, 1985.

[Bresin *et al.*, 1992] Roberto Bresin, Giovanni De Poli, and Alvise Vidolin. Symbolic and sub-symbolic rules system for real-time score performance. In *Proc. ICMC'92*, pp. 211-214, San Jose, 1992.

[Chung, 1992] Joseph T. Chung. *Hyperlisp Reference Manual*, MIT Media Laboratory, 1992.

[Fletcher and Rossing, 1991] Neville H. Fletcher and Thomas D. Rossing. *The Physics of Musical Instruments*, Springer Verlag, New York, 1991.

[Friberg, 1991] Anders Friberg. Generative rules for music performance: A formal description of a rule system. *Computer Music Journal*, **15**(2): pp. 56-71, 1991.

[Garton, 1992] Brad Garton. Virtual performance modelling. In *Proc. ICMC'92*, San Jose, pp. 219-222, 1992.

[Jaffe and Smith, 1983] David Jaffe and Julius O. Smith. Extensions of the Karplus-Strong plucked string algorithm. *Computer Music J.*, **7**(2): pp. 56-69, 1983.

[Karjalainen and Laine, 1991] Matti Karjalainen and Unto K. Laine. A model for real-time sound synthesis of guitar on a floating-point signal processor. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'91)*, pp. 3653-3656, Toronto, 1991.

[Karjalainen *et al.*, 1991] Matti Karjalainen, Unto K. Laine, and Vesa 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.

[Karjalainen, 1992] Matti Karjalainen. Object-oriented programming of DSP processors: A case study of QuickC30. In *Proc. IEEE ICASSP'92*, pp. V-601–V-604, San Francisco, 1992.

[Karjalainen *et al.*, 1993] Matti Karjalainen, Juha Backman, and Jyrki Pölkki. Analysis, modeling, and real-time synthesis of the kantele, a traditional Finnish string instrument, In *Proc. IEEE ICASSP'93*, pp. 229-232, Minneapolis, 1993.

[Karplus and Strong, 1983] Kevin Karplus and Alex Strong. Digital synthesis of plucked-string and drum timbres. *Computer Music Journal*, **7**(2): pp. 43-55, 1983.

[Laakso *et al.*, 1992] Timo I. Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K. Laine. Real-time implementation techniques for a continuously variable digital delay in modeling musical instruments. In *Proc. ICMC'92*, pp. 140-141, San Jose, 1992.

[Laakso *et al.*, 1993] Timo I. Laakso, Vesa Välimäki, Matti Karjalainen, and Unto K. Laine. Digital filter approximation of fractional delay—A tutorial review. To be published, 1993.

[Laine, 1988] Unto K. Laine. Digital modelling of a variable length acoustic tube. In *Proc. Nordic Acoustical Meeting (NAM'88)*, pp. 165-168, Tampere, 1988.

[Laroche and Jot, 1992] Jean Laroche and Jean-Marc Jot. Analysis/synthesis of quasi-harmonic sounds by the use of the Karplus-Strong algorithm. In *Proc. of the Second French Congress on Acoustics*, 1992.

[McIntyre *et al.*, 1983] M. E. McIntyre, R. T. Schumacher, and J. Woodhouse. On the oscillations of musical instruments. *Journal of the Acoustical Society of America*, **74**(5): pp. 1325-1345, 1983.

[Serra, 1989] Xavier Serra. *A System for Sound Analysis/Transformation/Synthesis Based on a Deterministic Plus Stochastic Decomposition,* Ph. D. dissertation. CCRMA Tech. Report STAN-M-58, Stanford University, California, 1989.

[Smith, 1982] Julius O. Smith. Synthesis of bowed strings. In *Proc. ICMC'82*, Venice, 1982.

[Smith, 1983] Julius O. Smith. *Techniques for Digital Filter Design and System Identification with Applications to the Violin*, Ph. D. dissertation. CCRMA Tech. Report STAN-M-14, Stanford University, California, 1983.

[Smith, 1987] Julius O. Smith. *Music Applications of Digital Waveguides.* CCRMA Tech. Report STAN-M-39, Stanford University, California, 1987.

[Smith, 1993] Julius O. Smith. Physical modeling by digital waveguides. *Computer Music Journal*, **17**(4), 1993.

[Sullivan, 1990] Charles R. Sullivan. Extending the Karplus-Strong algorithm to synthesize electric guitar timbres with distortion and feedback. *Computer Music Journal*, **14**(3): pp. 26-37, 1990.

[Välimäki *et al.*, 1993*a*] Vesa Välimäki, Matti Karjalainen, and Timo I. Laakso. Fractional delay digital filters. In *Proc. IEEE Int. Symp. on Circuits and Systems*, pp. 355-358, Chicago, 1993.

[Välimäki *et al.*, 1993*b*] Vesa Välimäki, Matti Karjalainen, and Timo I. Laakso. Modeling of woodwind bores with finger holes. In *Proc. ICMC'93* (this proceedings), Tokyo, 1993.