EXTRACTION OF PHYSICAL AND EXPRESSIVE PARAMETERS FOR MODEL-BASED SOUND SYNTHESIS OF THE CLASSICAL GUITAR

Cumhur Erkut[1], Vesa Välimäki[1], Matti Karjalainen[1], and Mikael Laurson[2]
[1] Helsinki University of Technology
Laboratory of Acoustics and Audio Signal Processing
Espoo, Finland
[2] Sibelius Academy, Centre for Music Technology, Helsinki, Finland
Cumhur.Erkut@hut.fi, Vesa.Valimaki@hut.fi
Matti.Karjalainen@hut.fi, laurson@amadeus.siba.fi
http://www.acoustics.hut.fi

Abstract

The calibration of model parameters is a crucial subproblem for realistic model-based sound synthesis. This paper describes the revision of the calibration process of a classical guitar model, and extends the parameter extraction procedure to capture information about performance characteristics, such as the damping regimes, repeated plucks, vibrato characteristics, different pluck styles and dynamic variations.

0 Introduction

Model-based sound synthesis of musical instruments has been an augmenting research field in the 1990's. String instrument models, in particular, have provided a theoretical framework for formulation of various model-based approaches. Throughout the years, some of these approaches have gained more popularity than the others, mainly because of their computational efficiency for real-time sound synthesis purposes and because of their good sound quality. The path towards today's most powerful and realistic synthesis models emerged from the extensions by Jaffe and Smith [1] to the simple, yet efficient Karplus-Strong algorithm [2]. The extensions emphasized the physics underlying the algorithm, and favored a modular, DSP-based formulation. Later, Smith generalized these ideas to devise the theory of digital waveguide modeling [3, 4], which is the basis of the most popular model-based sound synthesis algorithms today.

An important problem in model-based sound synthesis is the calibration of the parameter values of the algorithms. The algorithms themselves define the type of sound generation mechanism, but the individual character of a specific musical instrument, such as a classical acoustic

guitar, is reproduced only when the parameter values have been carefully tuned. The calibration scheme for the acoustic guitar model in this study is based on the pitch-synchronous short-time Fourier analysis of recorded tones [5]. The harmonics of analyzed tones are modeled as time-varying sinusoidal components with individual amplitude, frequency, and phase trajectories. In [6], a heterodyne filtering approach was tested for the extraction of harmonics. It can be more efficient computationally than the full FFT-based spectral analysis.

The synthesis model of the acoustic guitar can be interpreted as a physically based source-filter model. Thus, the input signal of the synthesis engine can be obtained by inverse filtering with the inverse transfer function of the synthesis model, after the model parameters have been calibrated [5]. In [7] and [6], it was proposed that the excitation signal can alternatively be extracted by subtracting a sinusoidal model from an analyzed tone. This approach, however, requires an additional equalization of the obtained excitation signals—otherwise the initial amplitudes of harmonics will be incorrect in synthetic tones.

Using the above approaches, it is usually possible to synthesize quite a realistic replica of recorded single tones. However, several complications arise if an analyzed tone has some irregularities, such as non-exponential decay rates of harmonics or strong beating. Smoothing of amplitude trajectories can help in avoiding bias caused by such irregularities [7, 6]. Still, more sophisticated analysis, modeling, and parameter calibration methods are needed to account for these phenomena.

Moreover, the realistic synthesis of individual tones does not guarantee that the model will sound natural in every musical context. Clearly, more information is required to synthesize realistic musical phrases and pieces of music. Events such as the starting transient and transition from one note to another seem to be among the most obvious next steps in more careful analysis of guitar playing. Recent publications have tackled the finger-string interaction [8, 9], and the detailed modeling of a single pluck [10, 11] and repeated plucks [12].

This paper describes a revision of the calibration process, and extends the parameter extraction procedure to capture information about performance characteristics. The structure of the paper is as follows: In Section 1, the synthesis model used in this paper is presented. This model is canonic in the sense that it captures the most important physical phenomena taking place in classical guitar simulation, yet it has a minimum number of model parameters. The choice of the model is also governed by the consistency requirements with earlier studies [5, 6, 7].

Next, we state the inverse problem of physical modeling, i.e., the problem of calibrating the model to derive the model parameters for a given particular string instrument sound. The problem is conceptually tackled in two phases. Section 2 describes add-ons to an existing calibration scheme [5, 7], and provides a more robust calibration procedure, including a feedback path to match the fundamental characteristics of analyzed and synthesized tones, such as overall decay rates.

Section 3 extends the analysis procedure to capture information about expression. More specifically, the analysis is carried out to highlight the damping regimes, repeated plucks of a string that is already sounding, vibrato characteristics, different pluck styles and dynamic variations of a professional player from the samples recorded in an anechoic room. Finally, in Section 4, we draw our conclusions.

1 The Synthesis Model

The plucked-string synthesis model used in this study is shown in Fig. 1. It is a feedback loop that consists of three elements: a delay line of L_I unit delays, a fractional delay filter $F(z)$ that implements interpolation between sample values, and a loop filter $H_l(z)$, which has a gentle lowpass characteristic.

The delay line and the fractional delay filter together contribute to the delay time around the loop, and thus determine the fundamental frequency of the synthetic tone. If it is desired to produce a tone with fundamental frequency f_0 (in Hz), the loop delay (in samples) must be

$$L = \frac{f_s}{f_0} \tag{1}$$

where f_s is the sampling rate (in Hz). Since the required delay L is generally non-integer, a fractional delay filter is practically always needed to implement the non-integral part of the delay, which we denote by d. The fractional delay filter also causes an extra (integral) delay D_I of a few samples, which depends on the type and order of the filter [13]. Hence, the overall delay caused by fractional delay filter $F(z)$ is $D = D_I + d$. The delay line in Fig. 1 implements the rest of the loop delay, i.e., $L_I = L - D$ samples.

Initially, Jaffe and Smith used a first-order allpass filter to fine-tune the delay [1]. Further work has considered the use of a Lagrange interpolation filter to realize the fractional delay [14]. Both approaches have their profits and weaknesses, as discussed in [13]. The main points are that the allpass fractional delay filters exhibit a perfectly flat magnitude response, but suffer from transients if the delay is changed during operation (to produce glissando or vibrato, for instance); the Lagrange interpolation filter does not produce transients when its coefficients are changed, but it has a lowpass character which varies as a function of fractional delay– this should be accounted for in the design of the loop filter, which simulates the losses. The lowpass characteristics are improved by increasing the filter order. We have chosen a fourth-order Lagrange interpolator to overcome the transients during vibrato simulation. Given the desired fractional delay D, the coefficients of the fourth-order Lagrange interpolator $f(n)$ can be calculated setting $N = 4$ in the following equation:

$$f(n) = \prod_{k=0, k\neq n}^{N} \frac{D-k}{n-k} \quad , \quad n = 0, 1, ..., N \tag{2}$$

In this work, the loop filter $H_l(z)$ is a one-pole filter with the following transfer function [5]:

$$H_l(z) = g\frac{1+a}{1+az^{-1}} \tag{3}$$

Since the main function of the filter is to attenuate the waveform of the tone from one period to the next in a natural manner, the loop filter is typically a lowpass filter with magnitude slightly smaller than unity. This is achieved when $g = 1 - e_1$ and $a = -e_2$, where e_1 and e_2 are small

nonnegative numbers. It is essential that the magnitude response of $H_l(z)$ does not exceed unity, because otherwise the feedback loop in Fig. 1 would become unstable. We thus require that $0 < g < 1$ and $-1 < a < 0$. The loop filter also contributes to the overall delay of the feedback loop, but in practice the delay caused by the one-pole filter is very small [5].

The excitation signal $x_p(n)$ can be a short pulse that imitates a pluck, such as an impulse response of a lowpass filter. However, it is better to derive the excitation signal from a recorded acoustic guitar tone. The main advantage of this approach is that the response of the body is also incorporated in the excitation signal $x_p(n)$. In this case, the excitation signal is longer compared to a short pulse, but it is more efficient to use a long excitation signal rather than to implement the guitar body as a high-order digital filter. The idea of incorporating the body response in the excitation signal is called commuted synthesis [15, 16, 17].

Although the simple model described in this section simulates the essential characteristics of the classical guitar, a more sophisticated, state-of-the-art synthesis model includes consolidated pluck and body wavetables, a pluck-shaping filter, a pluck-position comb filter, continuously variable delay lines, and multiple string models with individual loop filters and sympathetic couplings between the strings [18].

2 Extraction of Physical Parameters

Fig. 2 shows the calibration scheme proposed earlier in [7] for extraction of the parameters for the model shown in Fig. 1. The outputs of the calibrator are the integer and fractional parts of the loop delay (not explicitly shown in the figure), the loop filter parameters g and a, and the excitation signal which is used for commuted synthesis. This calibration scheme is discussed in detail in [7, 19], here just a brief overview is given.

The *Pitch Estimation* block outputs an estimate of the nominal fundamental frequency f_0 using the well-known autocorrelation method. The fundamental frequency estimate is converted to loop delay using Eq. (1). The integer part of the loop delay is used to set the delay length in Fig. 1, the fractional part is used to tune the fractional delay filter $F(z)$. The frequency estimate is also used to calculate the complex short-time spectra by the *Pitch-Synchronous Fourier Transform*.

With a *Peak Detection* algorithm, the amplitude, frequency and phase trajectories are obtained. The amplitude and frequency trajectories are used to estimate the decay rates of individual harmonics, and the estimated decay rates are used to determine the desired magnitude response of the loop filter $H_l(z)$. The remaining blocks are for extracting an excitation signal. The amplitude and frequency trajectories, together with the phase trajectory, are used to synthesize a deterministic signal, and the residual is obtained by a time-domain subtraction. The residual signal lacks the energy to excite the harmonics in the model. The *Residual Equalization* block provides the energy needed by inverse-filtering the deterministic signal and the residual separately.

This calibration scheme is easy to operate on several test signals for demonstrating the underlying methodology. However, the analysis of the recorded tones from a very large classical guitar tone database rises up some difficulties. The segmentation of the individual sound files

results in large fluctuations at analysis points. Different recording conditions causes different SNR values, which severely contradict with the assumption of fixed noise statistics. Sounds exhibiting slow decay characteristics result in very large complex spectral data to be stored during the analysis, which are sometimes beyond the available computational resources. Moreover, in some cases, the extracted model parameters do not result in a realistic replica of the analyzed sounds. The quality of the synthesized sounds is usually improved by manual fine-tuning of the model parameters.

To overcome these difficulties, we propose the extended calibration scheme shown in Fig. 3. After extracting low-level attributes from the analysis sound, the scheme uses the STFT-based core calibration scheme of Fig. 2 to extract the model parameters. The last block of the scheme is an optimization routine for automated fine tuning of the model parameters. The rest of this section describes the low-level attributes extraction and optimization blocks in more detail.

2.1 Extraction of Low-Level Attributes

The low-level attributes of the analyzed tone are the analysis points, the envelope of the signal and the noise profile, as shown in Fig. 4. The analysis points are simple indices to the waveform to locate appropriate analysis points, i.e., the location of the beginning of the attack portion (*idxbeg*), the location of the end of attack portion (*idxpbeg*, used for fundamental frequency estimation) and the location where the analyzed signal approaches to the noise level (*idxend*). Note that the size of STFT-matrix to be calculated later depends on the indices for the beginning and the end of the analysis data.

The envelope of the analyzed data is a very short signal. It provides a measure about the overall decay characteristics of the analyzed string tone (see Fig. 4). The noise profile is a spectral estimate of channel noise characteristics. It is computed in accordance with the number of FFT-bins used for the STFT-matrix, and hence provides propitious thresholding information. By comparing each hop of the STFT-matrix with the noise profile, the bins whose energies are below the noise profile may be completely discarded. This pruning translates the STFT-matrix into a sparse one, and thus exerts storage savings.

The thresholding by noise profile may also improve the line-fitting procedure prior to loop-filter design, as shown in Fig. 5. The task here is to fit a straight line to the decay of the 11th harmonic of a test signal. The magnitude of the harmonic drops rapidly and reaches the noise level around 150th hop. Note that the magnitude of the harmonic exhibits large oscillations before reaching the noise level, which is mainly an analysis artifact. If the noise level is set to a fixed value, the linear regression algorithm may result in unrealistic slope estimates, as shown with the dashed line in the figure. When the noise level corresponding the harmonic frequency is read from the noise profile, and the line fitting algorithm is terminated 6 dB above the corresponding noise level, a better fit is obtained, as shown by a solid line.

The computation of the low-level attributes is conducted as follows: A coarse estimate for the location of index *idxbeg* as the maximum of the analyzed sound is carried out as an initial step. An interval before the coarse beginning point is used to extract the noise characteristics. From this interval, several segments are transformed to frequency domain using the FFT algorithm, and then averaged to obtain the noise profile estimate. This noise profile is used as a

threshold to fine-tune the beginning point. The beginning point is required to be above 6 dB from overall noise energy. This method allowed us to capture very low-amplitude finger-string interactions prior to plucking process.

The envelope of the analyzed signal is extracted by rectifying and downsampling the original signal by factor of 512. The anti-aliasing lowpass FIR filter after downsampling carries out the smoothing task. The remaining two analysis indices are directly extracted from the envelope: The end point of the attack portion *idxpbeg* is the point of 20 dB drop after the attack and *idxend* is the point where the analyzed sound remains 6 dB above the overall noise energy. The 20 dB drop after the attack portion is the point were transients decay completely, and 6 dB above the noise level provides enough headroom for thresholding, as shown in Fig. 5.

2.2 Optimization

The optimization block in the right-hand side of Fig. 3 starts with the initial estimates of the loop-filter parameters and the equalized excitation signal. Using these parameters, a string tone is synthesized and the synthesis envelope is extracted using the same procedure used before to extract the analysis envelope. Analysis and synthesis envelopes are compared in the MSE sense, and if they do not match, a constrained ($0 < g < 1$) MSE minimization is performed using the loop-filter parameters as independent variables.

Note that the convergence of this nonlinear optimization problem clearly depends on the initial values of the loop-filter parameters and that there is always a risk that the algorithm converges to a local minimum, instead of the global minimum. However, in practice, the use of the core calibrator outputs as an initial guess minimizes the possibility that the algorithm sticks to a local minimum. Fig. 6 shows a typical synthesis envelope before and after the optimization routine. The unoptimized synthesis signal (dashed), obtained from the outputs of the core calibrator, has a milder overall decay compared to that of the analysis signal (dash-dotted). When the optimized parameters are used for synthesis, the optimized synthesis signal (solid) is obtained. It is evident that the match to the original overall decay is improved after the optimization. This is also verified by informal listening tests.

3 Extraction of Expressive Parameters

So far, the canonic classical guitar model under study is assumed to be a linear, time-invariant (LTI) system. The calibration procedure of this LTI system is based on the analysis of the individual tones. In the synthesis phase, once the model is excited, the parameters are not updated until the synthesized tone decays completely.

The expressive parameters, on the other hand, are usually hidden within the tone-to-tone transitions, or they govern the variations within a single tone. Hence, to proceed with the extraction of expressive parameters, we relax the assumptions about time-invariance. This relaxation is not very straightforward, since the analysis tools for time-varying systems are far from their maturity. Therefore, we choose the following approach: We regard the transitions as new note events with individual excitation signals (as in the case of damping in Section 3.1,

or repeated plucks in Section 3.2). We run the calibration scheme for the steady-state parts of the analysis sounds, and the transient excitation signal accounts for the rest of the phenomena. Rather than obtaining unique parameters for the transient part, we emphasize the need that the model is still well-behaved under transient regimes.

The synthesis of the vibrato regimes extracted in Section 3.3 also has some inherent problems. The time-variance of the delay-line length blurs the time-invariance assumptions of commuted synthesis. However, since the extracted vibrato frequencies are rather slow (less than 10 Hz), the synthesized tones with vibrato (i.e., time-varying delays) do not exhibit perceivable differences from the tones with no vibrato (i.e., fixed delays).

The analysis of different pluck styles (Section 3.4) and dynamic variations (Section 3.5) are less problematic, since they result in an LTI operation on the excitation signal prior to synthesis. Within the framework of commuted synthesis, separate excitation signals may be stored for different plucking styles, although an additional pluck-shaping filter [18], exterior to the model used in this study, may be separately calibrated using the extracted information.

3.1 Damping

In classical guitar playing, the damping of the string is achieved by the use of the flesh of a right-hand finger to absorb energy from the vibrating string. Fig. 7 shows an example of a damped guitar tone. After the usual pluck, the player touches the string again at the time instant $t_d \approx 185$ ms. The touch essentially divides the string into two portions and provides another excitation, which invokes the low-frequency body resonances [12]. For a very brief instance, two portions of the string vibrate separately, with a common lossy boundary condition at the point of the touch. Hence, the division is responsible for two simultaneously ringing tones with different frequencies, depending on the place of the touch. After $t = t_{db}$, only the low-frequency body resonances are persistent in the sound.

There are several strategies to simulate the damping process in a string synthesis model. An obvious choice is to change the loop-filter parameters to obtain a different decay characteristic to match the decay resulted by the damping. Without any transient-elimination scheme [20], the altering of the loop-filter coefficient may result in audible clicks in the sound, hence a better approach is to alter just the loop-filter gain, keeping the loop-filter coefficient unaltered. Setting the gain to a very small value can simulate the rapid damping of the tone, but this approach lacks both simultaneously ringing tones and the excitation of the body resonances.

A better approach is to run the extended calibration routine of Fig. 3 for the undamped part and the damping part separately to obtain the excitation signals and the loop-filter parameters for both regimes. This approach provides very realistic synthesis for damped guitar tones. For the analysis of the damping part, the hop-size for STFT is set to $1/100th$ of the window length. The optimization routine is required to vary only the loop gain g to minimize the MSE between analysis and synthesis envelopes, and the the loop-filter coefficient a is kept unchanged.

3.2 Repeated Plucks

When an already vibrating string is plucked again, a phenomenon similar to damping takes place; the flesh of the second right-hand finger temporarily damps the string. The analysis of the recorded samples shows that this regime is very short (between 20 and 60 ms). However, cutting this short segment from the sound file clearly effects the naturalness of the sound.

Although the reduction of the hop-size and extraction of a separate excitation signal for each transient between repeated plucks is possible, this approach expands the number of wavetables very rapidly. In terms of sound synthesis, good results are obtained, when the damping excitation signals (Section 3.1) are used between the repeated plucks.

3.3 Vibrato

The vibrato in the classical guitar is produced by repeatedly stretching of the string to fluctuate the fundamental frequency in a regular fashion. To extract information about particular choices of a professional player (see Section 5), we recorded examples of typical slow and fast vibrato regimes on various strings and fret positions. In analysis phase, the time-varying fundamental frequency of each tone is extracted using the autocorrelation method on short segments. Fig. 8 shows two examples of typical slow and fast vibrato regimes.

After the player initiates the vibrato at *starting time* t_0, the fundamental-frequency variation converges rapidly to a nearly sinusoidal waveform. We denote the time between the starting time and the convergence as the *transient time* t_t. While t_0 varies according to the score, the transient time t_t is typically around 0.5 s.

The lowest frequency during the vibrato is the nominal fundamental frequency of the tone without vibrato. The maximum deviation of the fundamental frequency depends on the string and fret. A nonlinear least-squares sine-fit to available analysis data after t_t gives the mean vibrato rates of 1.4 Hz and 4.9 Hz for the slow and the fast vibrato, respectively. The amplitude of the vibrato (i.e., the maximum frequency deviation) is systematically smaller for the fast case. The extracted vibrato patterns are overlayed with the actual vibrato patterns in Fig. 8.

3.4 Different Pluck Styles

An efficient way to model different pluck styles is to store a minimum number of excitation signals in wavetables, and to obtain the variations within the styles with the aid of a pluck-shaping filter. This approach was first mentioned in [1]. Here we describe our observations and derive the coefficients of the filter for different pluck styles. The transfer function of the pluck-shaping filter is the same as that of the loop-filter in Eq. (3).

The analysis data here consists of all possible variations of excitations with three different fingers (thumb, index and middle), two different styles (apoyando and tirando) and three different dynamics (piano, forte, and fortissimo). The calibration scheme of Fig. 3 is used to extract the excitation signals, and the differences are sought in the time and frequency-domain representations of the excitation signals.

No differences between index and middle finger excitations are observed by analysis. The thumb excitation on the other hand, is perceived different than both the index and middle finger excitations. In [12] this difference is explained as a result of the damping introduced by the flesh of the thumb. However, the differences are mainly in a very short initial segment of the attack section, and they are hard to pinpoint by analysis.

The gradual rise in the attack portions of *tirando* compared to sharp attacks of *apoyando* [11] were absent in the excitation signals. Generally speaking, the spectral characteristics of the excitation signals did not exhibit clearly-defined tendencies. However, the tirando excitation signals are systematically weaker than apoyando signals, and the scaling factor is found to be 3.5 dB on average.

Piano, forte, and fortissimo excitations, on the other hand, exhibit well-defined characteristics. Both the piano and forte excitations can be obtained from the fortissimo excitation signals by proper scaling and low-pass filtering. The coefficients of the pluck-shaping are determined by deconvolving the fortissimo excitation signals with forte and piano excitation signals and performing a weighted fit. For apoyando, the gains are found to be $g_f = 0.5403$ and $g_p = 0.4842$, for the forte and piano excitations, respectively. The pluck-shaping filter coefficients are $a_f = -0.9292$ and $a_p = -0.9944$. The transfer functions of the pluck-shaping filter, corresponding the forte and piano excitations are shown in Fig. 9. For simulating the tirando plucking style, one has to divide the gains approximately by a factor of 1.5.

3.5 Dynamic Variations

An experienced player has a very steady control over the dynamics, and usually the undesired dynamic variations are very small. However, if included in the model, these small variations avoid the monotony of the synthesized sounds. We asked the performer to keep the dynamics constant while repeatedly plucking the string. The analysis of the recorded signals and the extracted excitation signals show that the variations are very small indeed: The fluctuations in the average power of the excitation signals are only 0.0668 and 0.0359 percent for forte and mezzoforte, respectively. This small deviations may be modeled as a first-order random process, and the process may directly deviate the gain of the pluck-shape filter to simulate these small variations.

4 Conclusions

In this paper, we reported our revisions of the calibration process for a classical guitar synthesizer. After informal listening tests, we conclude that the added optimization block significantly improves the quality of the synthesized sounds.

The expressive parameter extraction procedures discussed in the paper rely heavily on the experimental data available. Although we tried to span a considerable area of the performance possibilities during measurements and analysis, there is still a vast amount of performance characteristics, such as the variation of the plucking point, the pizzicato excitations, portamento and glissando, left to be explored. The variables which are connected to statistical analysis may

be improved using a larger data-base of samples, and preferably with several repetitions.

To demonstrate the extracted physical and expressive parameters discussed in this paper, the examples of this paper, as well as short synthesized phrases and musical pieces will be available online at the URL: `http://www.acoutics.hut.fi/~cerkut/calib`. These demonstrations will serve as a validation criteria of the proposed physical and expressive parameter extraction procedures.

5 Acknowledgments

This work has been supported by Academy of Finland. The authors wish to thank Mr. Klaus Helminen who played the classical guitar samples used in this study and Mr. Sami Brandt who helped in the development of the MATLAB software during the summer of the year 1998.

References

[1] D. A. Jaffe and J. O. Smith, "Extensions of the Karplus-Strong plucked-string algorithm," *Computer Music Journal*, vol. 7, no. 2, pp. 56–69, 1983. Also published in Roads C. (ed). 1989. *The Music Machine*, pp. 481–494. The MIT Press. Cambridge, Massachusetts, USA.

[2] K. Karplus and A. Strong, "Digital synthesis of plucked-string and drum timbres," *Computer Music Journal*, vol. 7, no. 2, pp. 43–55, 1983. Also published in Roads C. (ed). 1989. *The Music Machine*. pp. 467-479. The MIT Press. Cambridge, Massachusetts.

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

[4] J. O. Smith, "Principles of digital waveguide models of musical instruments," in *Applications of Digital Signal Processing to Audio and Acoustics* (M. Kahrs and K. Brandenburg, eds.), pp. 417–466, Boston, Massachusetts, USA: Kluwer Academic Publishers, 1998.

[5] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy, "Physical modeling of plucked string instruments with application to real-time sound synthesis," *Journal of the Audio Engineering Society*, vol. 44, pp. 331–353, May 1996.

[6] V. Välimäki and T. Tolonen, "Development and calibration of a guitar synthesizer," *Journal of the Audio Engineering Society*, vol. 46, pp. 766–778, Sept. 1998.

[7] T. Tolonen and V. Välimäki, "Automated parameter extraction for plucked string synthesis," in *Proceedings of the Institute of Acoustics*, vol. 19, pp. 245–250, Sept. 1997. Presented at the International Symposium on Musical Acoustics, Edinburgh, UK.

[8] M. Pavlidou and B. E. Richardson, "The string-finger interaction in the classical guitar," in *Proceedings of the International Symposium on Musical Acoustics*, (Dourdan, France), pp. 559–564, July 1995.

[9] M. Pavlidou and B. E. Richardson, "The string-finger interaction in the classical guitar: theoretical model and experiments," in *Proceedings of the Institute of Acoustics*, vol. 19, pp. 55–60, Sept. 1997. Presented at the International Symposium on Musical Acoustics, Edinburgh, UK.

[10] G. Cuzzucoli and V. Lombardo, "Physical model of the plucking process in the classical guitar," in *Proceedings of the International Computer Music Conference*, (Thessaloniki, Greece), pp. 172–179, Sept. 1997.

[11] G. Cuzzucoli and V. Lombardo, "A physical model of the classical guitar, including the player's touch," *Computer Music Journal*, vol. 23, no. 2, pp. 52–69, 1999.

[12] C. I. Duruöz, "Synthesis of transients in guitar sounds," in *Proceedings of the International Computer Music Conference*, (Thessaloniki, Greece), pp. 188–191, Sept. 1997.

[13] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay–Tools for fractional delay filter design," *IEEE Signal Processing Magazine*, vol. 13, pp. 30–60, Jan. 1996.

[14] M. Karjalainen and U. K. Laine, "A model for real-time sound synthesis of guitar on a floating-point signal processor," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, (Toronto, Canada), pp. 3653–3656, 1991.

[15] J. O. Smith, "Efficient synthesis of stringed musical instruments," in *Proceedings of the International Computer Music Conference*, (Tokyo, Japan), pp. 64–71, Sept. 1993.

[16] M. Karjalainen, V. Välimäki, and Z. Jánosy, "Towards high-quality sound synthesis of the guitar and string instruments," in *Proceedings of the International Computer Music Conference*, (Tokyo, Japan), pp. 56–63, International Computer Music Association, Sept. 1993.

[17] M. Karjalainen and V. Välimäki, "Model-based analysis/synthesis of the acoustic guitar," in *Proceedings of the Stockholm Music Acoustic Conference*, (Stockholm, Sweden), pp. 443–447, 1993.

[18] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: From the Karplus-Strong algorithm to digital waveguides and beyond," *Computer Music Journal*, vol. 22, no. 3, pp. 17–32, 1998.

[19] T. Tolonen, "Model-based analysis and resynthesis of acoustic guitar tones," Master's thesis, Helsinki University of Technology, Espoo, Finland, Jan. 1998. Report 46, Laboratory of Acoustics and Audio Signal Processing. See http://www.acoustics.hut.fi/∼ttolonen/thesis.html.

[20] V. Välimäki and T. I. Laakso, "Supression of transients in variable recursive digital filters with a novel and efficient cancellation method," *IEEE Transactions on Signal Processing*, vol. 46, pp. 3408–3414, Dec. 1998.

Figures

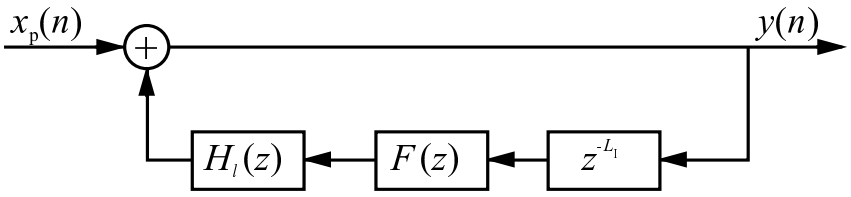

Figure 1: Block diagram of plucked-string model used in this study. From left to right, the one-pole loop-filter $H_l(z)$ implements the frequency-dependent decays of the harmonics, the 4th order Lagrange interpolator $F(z)$ accomplishes the fine-tuning of the model, and the delay line z^{-L} implements the loop delay.

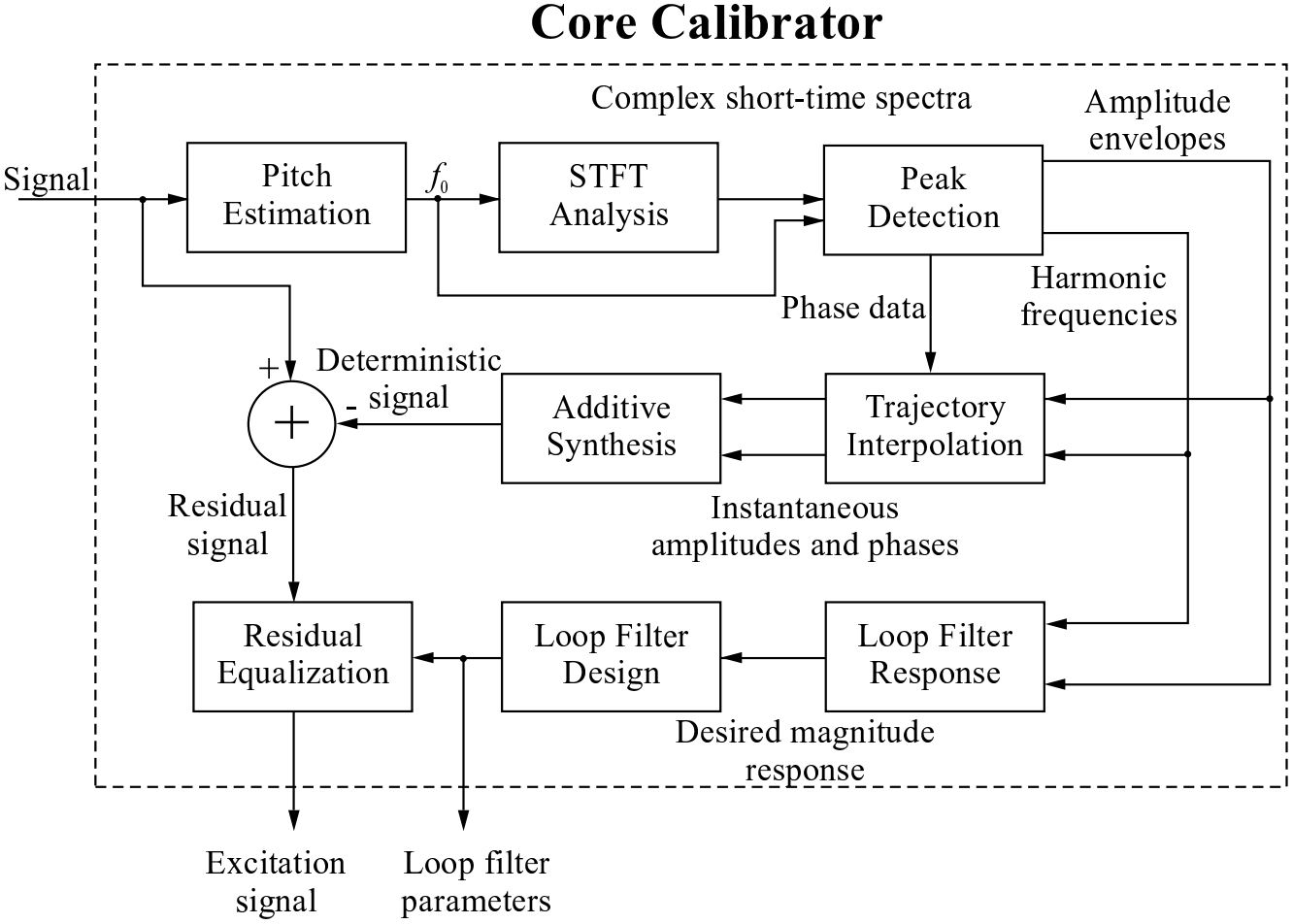

Figure 2: The STFT-based calibration scheme proposed earlier in [5, 7].

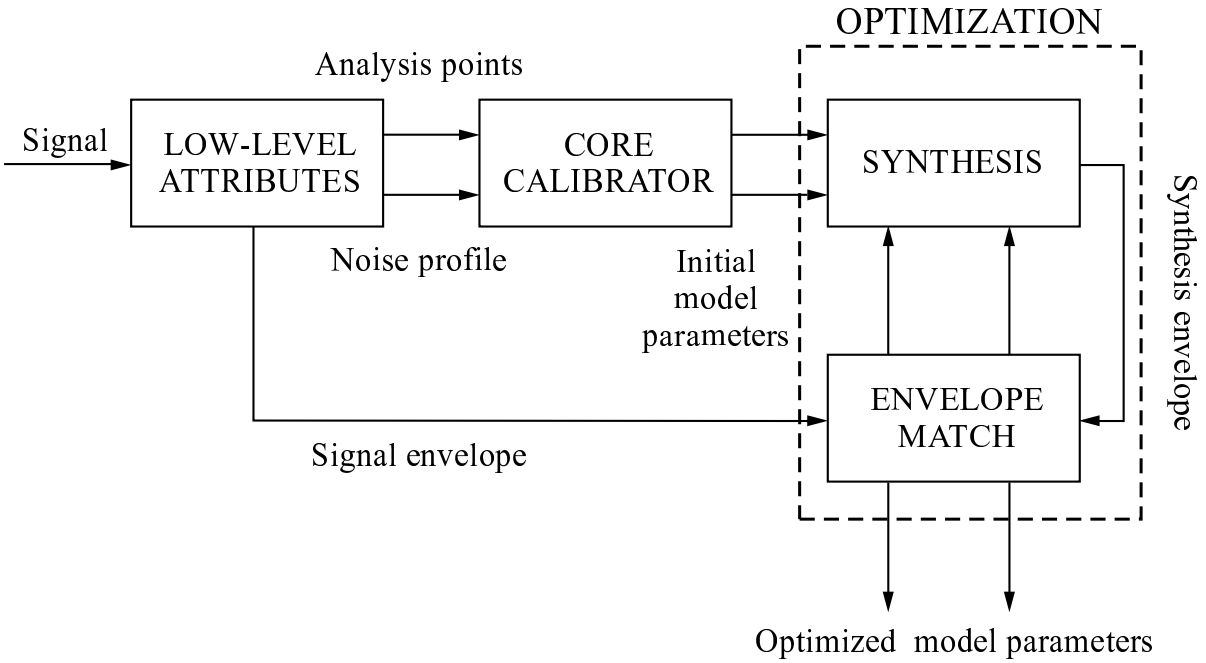

Figure 3: An extended calibration scheme. The scheme passes the low-level attributes to the core calibration scheme of Fig. 3, then runs an optimization routine to match the overall decay of the analyzed and synthesized sounds.

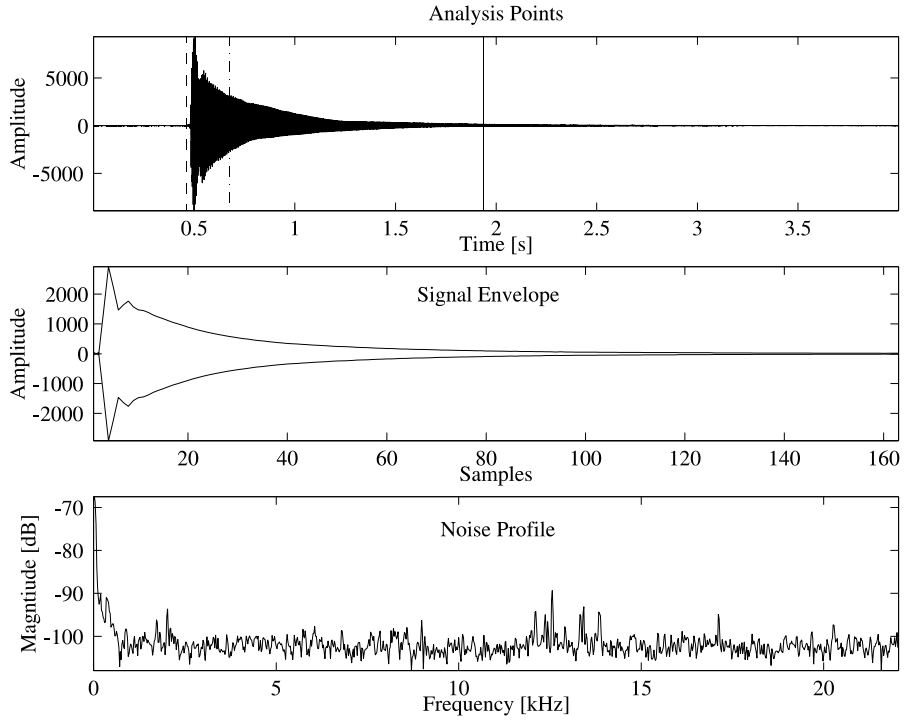

Figure 4: The low-level attributes of the analyzed sound. The uppermost plot shows the beginning (dashed), the pitch extraction (dash-dotted), and the end (solid) indices. The middle plot shows the analysis envelope, and the lowermost plot shows the extracted noise profile.

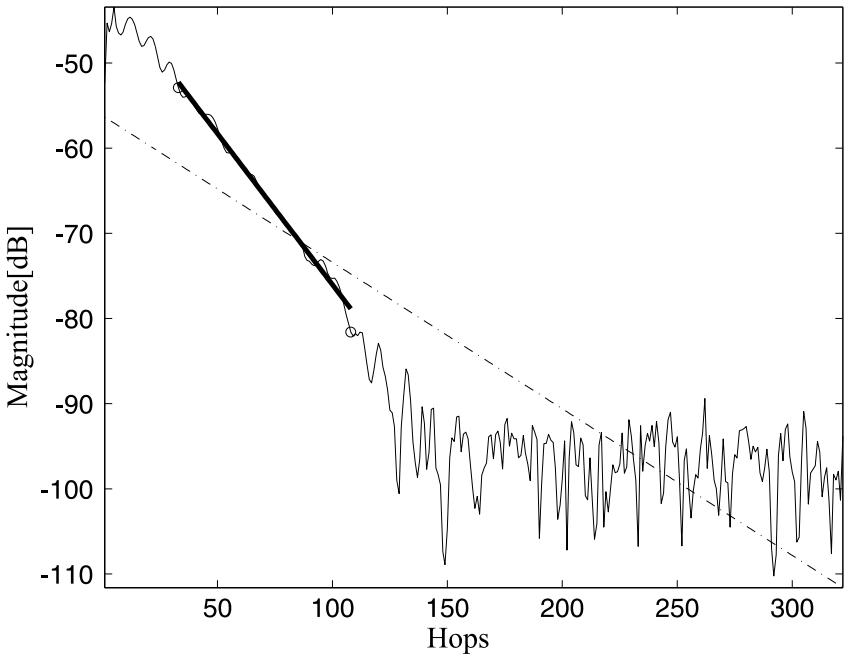

Figure 5: Line-fit to the 11th harmonic. The dashed line shows the fitted line to match the decay of the harmonic without any thresholding, whereas the thick solid line shows the line-fit with a threshold depending on the noise profile.

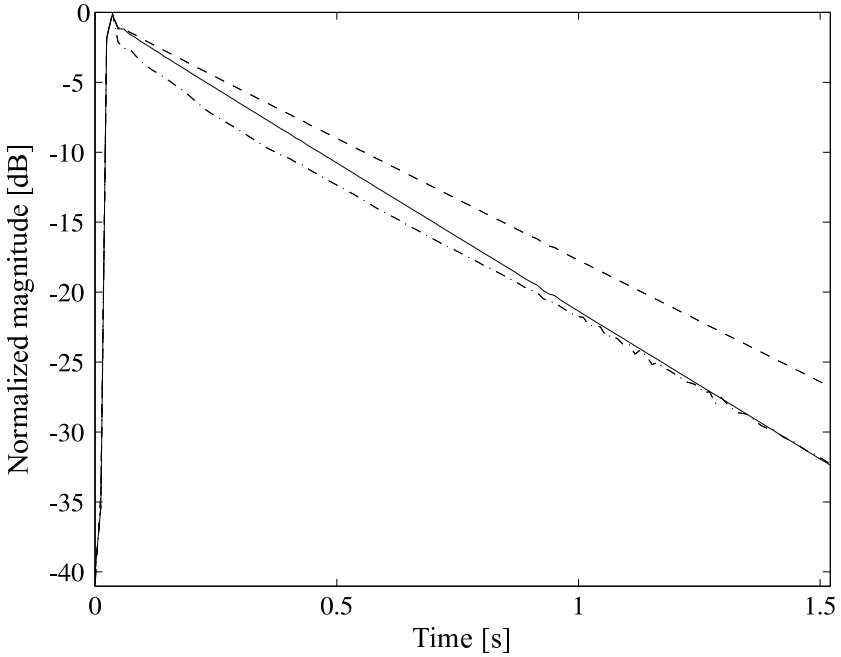

Figure 6: Overall decay optimization. The synthesis output of the core calibration routine (dashed) differs from the original waveform (dash-dot) in overall decay characteristics. The output of the optimization block (solid) displays a superior match.

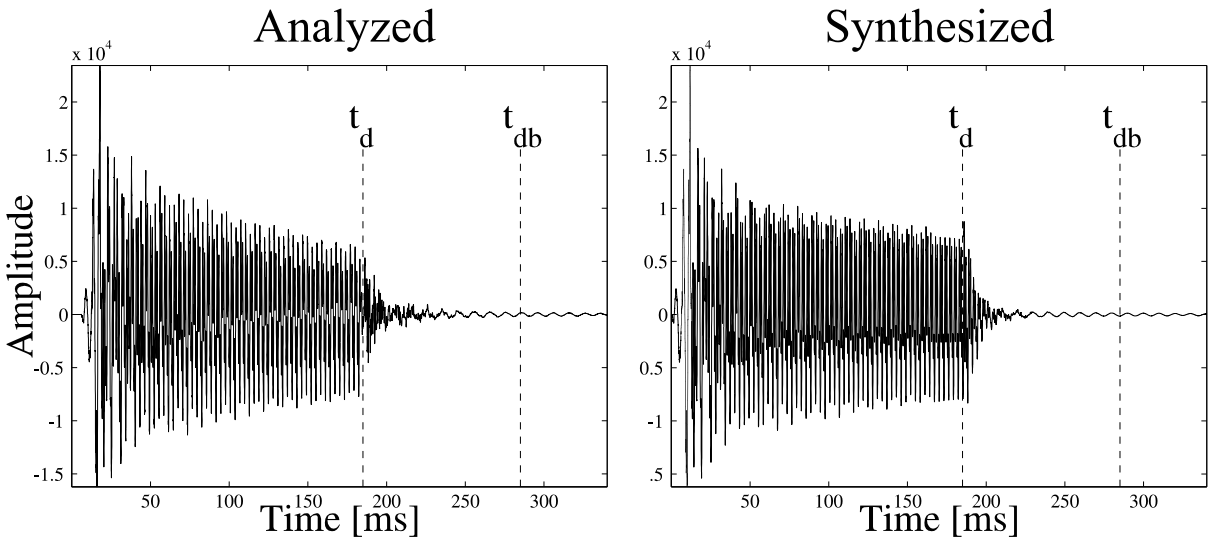

Figure 7: A damped tone example. The player damps the string at time instant $t = t_d$, and after $t = t_{db}$, only the lowest body resonances are persistent. The synthesis tone simulates a similar behavior.

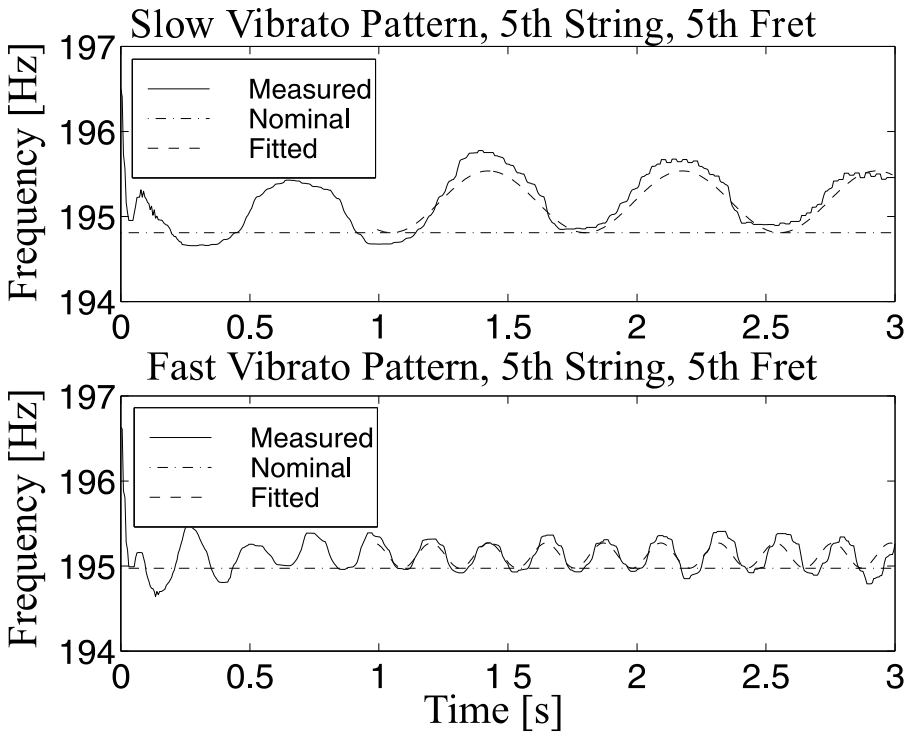

Figure 8: Two examples of fundamental-frequency deviations during vibrato. The fundamental frequency (solid) fluctuates between the nominal fundamental frequency (dash-dotted) and a maximum vibrato depth. The sine-fit (dashed) gives a good match after the transient time.

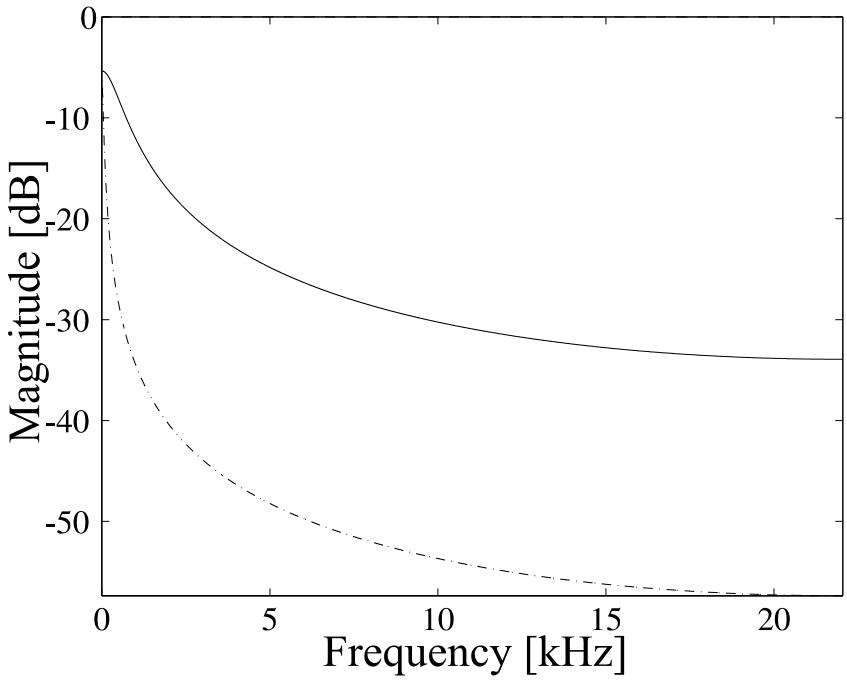

Figure 9: The magnitude responses of the pluck-shape filter for apoyando forte (solid) and apoyando piano (dashed) plucks.