



---

# Audio Engineering Society

# Convention Paper 7501

Presented at the 125th Convention  
2008 October 2–5 San Francisco, CA, USA

*The papers at this Convention have been selected on the basis of a submitted abstract and extended precis that have been peer reviewed by at least two qualified anonymous reviewers. This convention paper 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 papers may be obtained by sending request and remittance to Audio Engineering Society, 60 East 42<sup>nd</sup> Street, New York, New York 10165-2520, USA; also see [www.aes.org](http://www.aes.org). All rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.*

---

## A Parametric Instrument Codec for Very Low Bitrates

Mirko Arnold<sup>1</sup> and Gerald Schuller<sup>2</sup>

<sup>1</sup> Fraunhofer Institute for Digital Media Technology, 98693 Ilmenau, Germany  
[mirko\\_arnold@gmx.net](mailto:mirko_arnold@gmx.net)

<sup>2</sup> Fraunhofer Institute for Digital Media Technology, 98693 Ilmenau, Germany  
[shl@idmt.fraunhofer.de](mailto:shl@idmt.fraunhofer.de)

### ABSTRACT

A technique for the compression of guitar signals is presented which utilizes a simple model of the guitar. The goal for the codec is to obtain acceptable quality at significantly lower bitrates compared to universal audio codecs. This instrument codec achieves its data compression by submitting an excitation function and model parameters to the receiver instead of the waveform. The parameters are extracted from the signal using weighted least squares approximation in the frequency domain. For evaluation a listening test has been conducted and the results are presented. They show that this compression technique provides a quality level comparable to recent universal audio codecs. The usability however is at this stage limited to very simple guitar melody lines.

### 1. INTRODUCTION

With the development of sound source separation algorithms and the object based MPEG-4 framework, the types of audio signals are getting more diversified. There is not only speech and music that can be considered separately for coding but also single instrument signals. Coding those in a way appropriately matched to the instrument characteristics promises a large jump in coding efficiency, in the same manner as known from speech coding.

In sound synthesis applications, like electronic instruments and software synthesizers, a variety of instrument models is applied to create instrument sounds with a minimum of memory consumption. In a coding application the focus will be on reaching a minimum number of parameters, which have to be transmitted.

While for pure synthesis it is possible to manually tune the model parameters to achieve the most natural sound, the parameter acquisition in a coding context has to be performed automatically. Additionally, the tone onset needs to be detected.

The presented instrument codec combines onset detection with automated parameter acquisition for a simple guitar model known from synthesis applications. By transmitting only the model parameters it achieves average bitrates of around 2.5 kbit/s, depending mostly on the playing speed and the fundamental frequencies of the tones.

At the current development stage the codec is able to process plucked guitar melody lines, i.e. one tone played after another. It has to be extended to support chords and playing styles like vibrato or bending.

For evaluation purposes, a listening test has been conducted in which the instrument codec has been compared to current universal transform and partially parametric audio codecs. The test showed no preference of the other codecs over the instrument codec, which was able to achieve significantly lower bitrates compared to the universal audio codecs.

This paper is organized as follows: Chapter 2 defines the goal of the underlying work. In Chapter 3 the problem is formulated. Chapter 4 gives an overview on previous approaches and the new approach of the presented instrument codec is described in detail. Chapter 5 describes the listening test and gives an overview on its results.

## 2. GOAL

The goal of this work is the development of a simple instrument codec that can achieve significantly lower bitrates than universal transform based codecs at their minimum bit rate. The quality shall be at least comparable. The character of the instrument and the plucking style are to be preserved, i.e. those characteristics shall be recognizable after decoding.

## 3. PROBLEM

There are several steps in the development process that need to be performed. First, different synthesis models need to be studied to pick one that is suitable for use in a coding application. As we intend to reach very low bitrates and the coding shall be performed in real time, the complexity of the underlying model needs to be low, i.e. a minimum amount of parameters shall describe the model, and those parameters shall be easy to determine from the input signal. The found model then needs to be implemented in an efficient way and finally an

algorithm to extract the model parameters from the input signal needs to be developed. The focus here lies on both accuracy in parameter estimation and minimization of possible mistuning of the parameters.

## 4. APPROACH

Since the tasks can be grouped in finding a suitable synthesis model for plucked string instruments and obtaining the parameters for that model from the input signal, previous approaches of both tasks are shortly summarized in this chapter.

### 4.1. Previous Approaches

#### 4.1.1. Synthesis Model

There are a lot of synthesis techniques that work with instrument models. The “commuted digital waveguide synthesis”, proposed independently in [1] and [2], was identified as the most promising approach concerning both the simplicity and quality constraints. It is based on the digital waveguide method, which in its basic form is a discrete-time representation of the d’Alembert solution to the wave equation, i.e. it directly describes the propagation of the wave components, which in superposition fully represent the wave field. This approach was first used by Kelly and Lochbaum in their speech synthesis model [3].

The principle of waveguide synthesis is depicted in Figure 4.1. This represents for example the propagation of transversal waves in a string which is fixed at both ends. The excitation function could be the initial displacement, velocity or acceleration of the string. The length of the delay line  $L_l$  follows from the sampling frequency  $f_s$  and the fundamental frequency  $f_0$  of the tone:

$$L_l = \frac{f_s}{f_0}. \quad (1)$$

Note that the values for  $L_l$  do not necessarily have to be integral numbers. This issue has to be dealt with in the implementation.

The reflection filters  $R_A(z)$  and  $R_B(z)$  can model both the reflection losses and the losses of the string itself like for example friction. These losses can be combined at the ends because of the commutativity of linear systems.

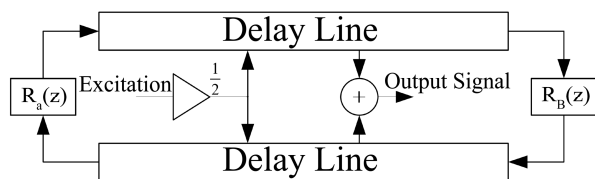


Figure 4.1: One-dimensional waveguide with reflection filters  $R_A(z)$  and  $R_B(z)$

The first step towards commuted waveguide synthesis is to reduce the structure to a single loop with only one delay line as described in [4]. This is achieved by again making use of the commutativity principle. In the model in Figure 4.1 the characteristics of the output signal are determined by the excitation function and the filter responses, but also by the input and output positions in the delay lines. The effect caused by those positions in combination with the two delay lines can as well be represented by an input and output filter at both ends of a loop structure with a single delay line. By changing the ordering of the output filter and this string loop, both the input and the output filter can be integrated into the excitation function. The remaining single delay line model is depicted in Figure 4.2.  $X(z)$  represents the excitation signal with integrated input and output filters. The string loop consists of a delay line  $z^{-L}$  and the loop filter  $H_l(z)$ . In order to not only allow discrete fundamental frequencies, the fractional delay filter  $F(z)$  offers the possibility to fine tune the delay.

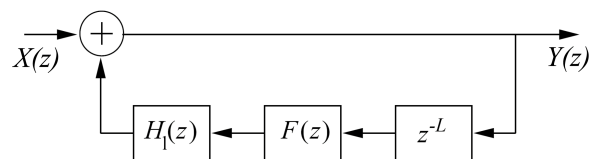


Figure 4.2: Single delay line model as in [5].

At this stage the model only represents the string itself. For acoustic guitar sounds, the response of the guitar body is essential for the tonal character. Since the body is a quite complex structure it is often implemented by using resonators representing the most prominent modes. This resonator structure can be used in cascade or in parallel to the string model as pointed out in [6], or it can be integrated into the excitation function. This last option is called commuted waveguide synthesis, which has been chosen for the presented codec.

### 4.1.2. Parameter Estimation

For the commuted waveguide synthesis model, the parameters to be estimated are the fundamental frequency  $f_0$ , the coefficients of the loop filter  $H_l(z)$  and the excitation function  $X(z)$  (see Figure 4.2).

There exist a variety of approaches for fundamental frequency estimation. An overview of the underlying principles is given for example in [7]. In [6] the pitch detection is performed using a method based on the autocorrelation approach. The results of this method are very reliable when choosing the right initial parameters. The instrument codec presented here uses a similar approach with some added pre processing to increase the reliability.

As it is possible for pure sound synthesis purposes to determine good sounding parameters partly by hand or to use very sophisticated estimation algorithms, there are not many approaches for pure automated estimation of the loop filter parameters and the excitation function, which are usable in a coding application.

One algorithm to be mentioned here is described in [6]. There, a sinusoidal representation of the input signal is obtained by observing the decay of the prominent modes found in the signal over the length of a tone. From these so called “energy decay reliefs” the parameters of the loop filter  $H_l(z)$  (see Figure 4.2) are computed. The excitation signal is obtained by computing the difference of the original and the sinusoidal representation in the time domain. This residual signal is then equalized in order to achieve a correct loudness level. As the model utilizes a separate body response calculation and synthesis of this response in parallel paths using two resonators, two more excitation functions are determined from the input signal. For pure synthesis purposes this offers the flexibility of changing the attributes for the player-string-system and the body separately.

### 4.2. New Approach

The approach presented in this paper is a combination of the commuted waveguide synthesis, as described in the previous chapter, and a simplified parameter extraction algorithm, which is described in detail in this chapter.

### 4.2.1. Necessary Parameters

As we use the commuted waveguide synthesis model, the parameters to be transmitted are those identified in chapter 4.1.2. These are the fundamental frequency of the tone  $f_0$ , the coefficients of the loop filter  $H_l(z)$  and the excitation function  $X(z)$ .

There is of course the possibility to work with wavetables instead of transmitting the whole excitation function, but these would probably get too big, if the instruments character shall really be preserved.

A solution to further reduce the achievable bit rates could be a wavetable containing more universal input filter functions. Those could make it possible to transmit much shorter excitation functions while still retaining the character of the plucking event and the instrument. In this direction further research is required.

The codec presented here transmits a new excitation function each time a new onset is detected.

### 4.2.2. Fundamental Frequency Estimation

The approach for estimating the fundamental frequencies of the tones is based on the autocorrelation method, i.e. searching the most prominent peak in the autocorrelation function of a short snippet of a tone.

Choosing a proper length of the snippet is crucial for the reliability of the algorithm. Therefore a preprocessing step is performed that roughly estimates the fundamental frequency by simply counting extremes in the signal. The window length is then set to three to five times the  $f_0$  estimate, which leads to a very prominent peak at the displacement that corresponds to the fundamental frequency of the tone.

After this step both the delay line length  $L_n$  and the remaining fractional delay  $L_f$  can be determined by using the rounded result of equation (1) and its remainder.

A suitable structure for the fractional delay filter  $F(z)$  was proposed in [8]:

$$F(z) = \frac{\alpha_D + z^{-1}}{1 - \alpha_D z^{-1}}. \quad (2)$$

$\alpha_D$  is determined by

$$\alpha_D = \frac{1 - L_f}{1 + L_f}, \quad (3)$$

with  $L_f$  being the fractional delay explained above.

### 4.2.3. Onset Detection

As will be described later in this chapter, exact onset detection is of great importance for the extraction of the excitation function. Therefore it is performed in two steps. First, the onset position is determined roughly by block-wise energy estimation and checking for sudden increases, and second, the first wave peak is determined in that block.

### 4.2.4. Extraction of the Excitation Function

The first milliseconds after the plucking event do not differ much for different realistic choices of the loop filter. Therefore it is possible to use appropriate default model parameters for synthesis when only considering this early decay stage. Defined properties of the early response of that default model to different excitation functions can then be used as variables to be matched to the input signal characteristics.

In particular the difference between the frequency content of this early response and the corresponding part of the input signal was used as the variable to be minimized. The length of the response was chosen depending on the fundamental frequency of the tone.

Possible candidates for the excitation function were chosen by simply taking snippets of different lengths from different positions around the detected onset position directly out of the input signal. The beginning and end points were set to zero crossings with short fade in and fade out transitions before feeding them to the waveguide model.

After picking the best excitation function some corrections have to be made in order to achieve a correct amplitude level of the tone. The length of the excitation function directly corresponds to the energy fed to the waveguide model. Therefore, the loudness of the synthesized tone is determined by the length of the excitation function. This issue is resolved by attenuating the excitation function appropriately. An additional gain factor  $g_E$  is transmitted, that describes this attenuation.

The energy loss at the plucking event, resulting from this attenuation, is addressed by adding the excitation function, attenuated with the factor  $(1-g_E)$ , to the resynthesized tone.

#### 4.2.5. Loop Filter Parameters

It was shown for example in [9] that a one-pole filter with the following structure is sufficiently accurate for high quality synthesis:

$$H_l(z) = g_l \frac{1 + a_l}{1 + a_l z^{-1}}, \quad (4)$$

with the gain factor  $g_l$  and filter coefficient  $a_l$ , which need to be estimated from the signal characteristics.

This is done again by matching the frequency content of the synthesized signal to the original at a position during a more steady, slowly decaying part of the signal. First, the coefficient  $a_l$  is estimated. This is performed by iteratively minimizing the mean squared error between the normalized amplitude spectrum of the original and the resynthesized tone. With this method,  $a_l$  was found to be sufficiently accurate after four iteration steps. What remains is to adjust the energy with the gain factor  $g_l$ .

#### 4.2.6. Transmission and Resynthesis

After successful parameter extraction, the data to be transmitted consists of the following:

- Excitation function  $x(n)$
- Excitation gain factor  $g_E$
- Onset position
- Fundamental frequency  $f_0$
- Loop filter parameters  $g_l$  and  $a_l$

There were no efforts made to efficiently code these parameters yet, except reducing the bit resolution of the excitation function to 8 bit and the sampling frequency to 22050 Hz if it was higher in the original signal.

Of course there are a lot of options to further reduce the bit rate using, for example, prediction and entropy coding for the excitation function or the application of

psychoacoustic models. Further research is needed to choose an appropriate approach for this task.

With the mentioned bit resolution and sampling frequency restrictions, the instrument codec achieves an average bit rate of around 2.5 kbit/s. The bit rate directly depends on the playing speed and the fundamental frequencies of the tones, i.e. fast and low frequency melodies will need the highest bit amount.

The resynthesis step is pretty straightforward. All received parameters are fed to the instrument model. Additionally, at the tone onset positions the excitation function needs to be added to the signal with the factor  $(1-g_E)$ , as described in section 4.2.4., in order to achieve correct amplitude levels.

## 5. LISTENING TEST

For evaluation purposes, signals of three different kinds of guitars were recorded in a quiet recording studio environment (classical guitar – microphone recording; western guitar – microphone and built-in piezo pickup recording; and electric guitar – pickup recording). These included small melody snippets and a scale played over two octaves. Depending on the playing speed the codec achieved bitrates between 1 and 6 kbit/s.

Using these recordings a MUSHRA listening test was conducted to find out how the instrument codec compares to universal transform based and partly parametric audio codecs at their minimum bit rate (HE-AAC, Ogg Vorbis, AMR-WB+).

The test items were grouped into “microphone recordings” and “pickup recordings”. Figure 5.1 shows the averaged result of the listening test with the “pickup recordings” test set and corresponding average bit rates are shown in Table 5.1. Figure 5.2 and Table 5.2 show the same for the “microphone recordings” test set.

The results show no significant preference of one codec over the others while the average bitrates ranged from 2.4 kbit/s (the presented instrument codec) to 14 kbit/s (Ogg Vorbis).

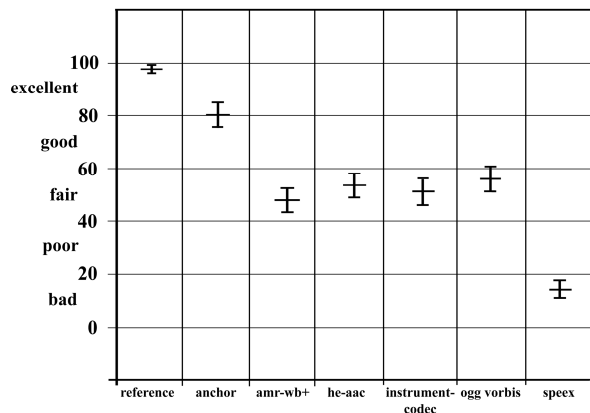


Figure 5.1: Average results of the “pickup recordings” test set. The Figure shows averages and 95% confidence intervals.

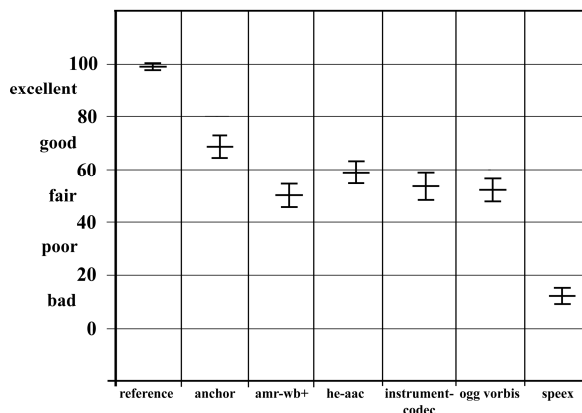


Figure 5.2: Average results of the “microphone recordings” test set

Codec	AMR-WB+	HE-AAC	Instrument-codec	Ogg Vorbis	Speex
Bit Rate (kbit/s)	6.4	13.2	1.9	13.7	4.9

Table 5.1: Average bit rate of the different audio codecs for the “pickup recordings” test set

Codec	AMR-WB+	HE-AAC	Instrument-codec	Ogg Vorbis	Speex
Bit Rate (kbit/s)	6.5	13.6	2.9	14.2	4.9

Table 5.2: Average bit rate of the different audio codecs for the “microphone recordings” test set

6. CONCLUSION

The presented instrument codec reaches significantly lower bitrates than universal audio codecs. The listening test showed no quality drawbacks compared to the other codecs used. In object based coding contexts or in combination with robust source separation such a specialized instrument codec could lead to more efficient coding results.

However the codec is still limited to very simple guitar melody lines. For universal use the codec must be extended to be able to reproduce chords and playing styles like bending or tremolo.

Further research could also lead to even more efficient coding with the use of wavetables or additional coding of the model parameters.

7. REFERENCES

[1] Karjalainen M., Välimäki V., Jánosy Z., “Towards high-quality sound synthesis of the guitar and string instruments”, Proc. Int. Computer Music Conf., Tokyo, Japan, 1993, pp 56–63

[2] Smith J. O., “Efficient synthesis of stringed musical instruments”, Proc. Int. Computer Music Conf., Tokyo, Japan, 1993, pp 64–71

[3] Kelly J. L., Lochbaum C. C., “Speech Synthesis”, Proc. 4th Int. Congr. Acoustics, Copenhagen, Denmark, 1962, pp 1–4

[4] Karjalainen M., Välimäki V., Tolonen T., “Plucked-String Models: From the Karplus-Strong Algorithm to Digital Waveguides and Beyond”, Computer Music Journal 22(3), 1998, pp. 17-32

- [5] Välimäki V., Huopaniemi J., Karjalainen M., Jánosy Z., “Physical modeling of plucked string instruments with application to real-time sound synthesis”, *Journal of the AES* 44(5), 1996, pp. 331-335
- [6] Tolonen T, “Model-Based Analysis and Resynthesis of Acoustic Guitar Tones”, Master Thesis, Helsinki University of Technology, 1998
- [7] Rabiner L R, Cheng M J, Roseberg A E, et al., “A Comparative Performance Study of Several Pitch Detection Algorithms”, *IEEE Trans. on Acoust., Speech and Sig. Processing* 24(5), 1976, pp. 399-418
- [8] Välimäki V, “Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters”, PhD thesis, Helsinki University of Technology, Finland, 1995
- [9] Välimäki V, Huopaniemi J, Karjalainen M et al., “Physical modeling of plucked string instruments with application to real-time sound synthesis”, *Journal of the AES* 44(5), 1996, pp. 331-353