Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SCD/TR 2002-165

Player modeling and control of a physical model of a string instrument [1]

Axel Nackaerts[2], Bart De Moor[3] and Rudy Lauwereins.[4]

October 2002
Accepted for the 1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)

[1]This report is available by anonymous ftp from *ftp.esat.kuleuwen.ac.be* in the directory *pub/sista/nackaerts/reports/report_MPCA02.ps.gz*

[2]E-mail: *axel.nackaerts@esat.kuleuwen.ac.be* This Work is supported by: the Flemish Government: Research Council KUL (GOA Mefisto-666), FWO: (G.0240.99, G.0407.02, G.0197.02, G.0141.03, G.0491.03, and G.0120.03, Research communities ICCoS, ANMMM, PhD and postdoc grants), AWI: Bil. Int. Research Program; IWT (PhD grants, Soft4s); EU: Eureka 2063-Impact, Eureka 2419-FLiTE, CAGE, ERNSI; Federal State DWTC (IUAP IV-02, IUAP V-22), PODO-II (CP/40); Industrial contract research (IPCOS, Electrabel, Elia, Data4s, LMS, VIB).

[3]K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SCD, Kasteelpark Arenberg 10, 3001 Leuven, Belgium, Tel. 32/16/32 17 09, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuwen.ac.be/sista-cosic-docarch.* E-mail: *bart.demoor@esat.kuleuwen.ac.be*

[4]IMEC vzw, Kapeldreef 75, 3001 Leuven, Belgium, WWW: *http://www.imec.be/.* E-mail: *rudy.lauwereins@imec.be*

Abstract

In this paper, we discuss the control of a physical model of a string instrument, the acoustic guitar. Both non-parametric and parametric inputs are described, and the specific implementation issues using the MPEG-4 Structured Orchestra Language are addressed. This results in a multi-layered model, including a MIDI interpreter, a score-level Finite State Machine and a core signal processing routine.

PLAYER MODELING AND CONTROL OF A PHYSICAL MODEL OF A STRING INSTRUMENT

Axel Nackaerts, Bart De Moor

K.U. Leuven, Dept. of Elec. Engineering, SCD
Kasteelpark Arenberg 10
3001 Heverlee, Belgium
Axel.Nackaerts@esat.kuleuven.ac.be
Bart.Demoor@esat.kuleuven.ac.be

Rudy Lauwereins

IMEC vzw
Kapeldreef 75
3001 Heverlee, Belgium
Rudy.Lauwereins@imec.be

ABSTRACT

In this paper, we discuss the control of a physical model of a string instrument, the acoustic guitar. Both non-parametric and parametric inputs are described, and the specific implementation issues using the MPEG-4 Structured Orchestra Language are addressed. This results in a multi-layered model, including a MIDI interpreter, a score-level Finite State Machine and a core signal processing routine.

1. INTRODUCTION

Physical modeling sound synthesis is based on the simulation of a model of a real musical instrument to produce synthetic output. It requires both an accurate acoustical representation of the instrument, and computationally efficient structures for real-time sound production. Today, there exist physical models for instruments ranging from the clarinet to the musical saw. The main advantage of physical modeling sound synthesis is the physical relevance of the controlling parameters, giving these class of models a distinctive advantage for live performance by a musicians. The use of physical parameters to drive a model requires a mapping from the output of sensors excited by the player, to the physical control parameters of the model. A problem arises when there is no human player, and only the score is available as an input. The note pitch, duration, attack velocity and other score information has to be translated to suitable control parameters. It is only recently that the control of the physical models has received more attention.
We start with a short overview of a physical model of the guitar in section 2 and discuss non-parametric source-filter control and parametric control of the model in section 3. Finally, we look at the practical implementation using the MPEG-4 Structured Audio Orchestra Language in section 4.

2. PHYSICAL MODEL

The model used in this work is the fully coupled digital waveguide model, where two single delay line digital waveguide models are used for every string to simulate the dual-polarization nature of guitar strings. A coupling matrix combines the outputs of the individual string models to determine the inputs of both the string models and the body model. The characteristic feature of this guitar model is the feedback from body model to the strings [1],[2]. The parameters for the string models can be estimated using the Short Time Fourier Transform, subband Hankel Singular Value Decomposition, or various other methods [3], [4], [5], [6]. The coupling parameters require a better insight into the actual structure of the instrument. Heuristic rules, analytical approximations and general non-linear optimization have been used to accurately determine the coefficients [7],[8],[9]. This model requires input for all the string models and the body. The string models can be grouped by two, and the input signal $\mathcal{I}$ is distributed over the horizontal and vertical polarizations according to the pluck angle θ (see figure 2):

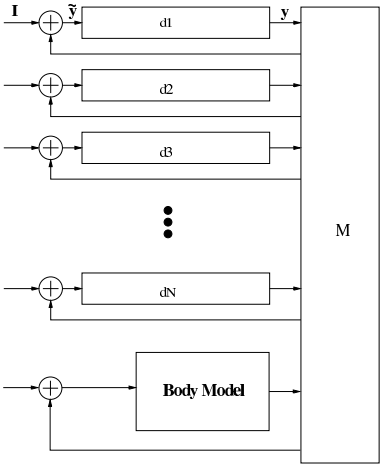

Figure 1: *The fully coupled digital waveguide model for an acoustic guitar uses 12 string models and a body model, coupled by a coupling matrix M.*

$$\mathcal{I}_{\text{hor}} = \mathcal{I}\sin(\theta) \quad (1)$$

$$\mathcal{I}_{\text{ver}} = \mathcal{I}\cos(\theta). \quad (2)$$

This reduces the inputs to six. Depending on the particular implementation, one could use either displacement, velocity or force waves in the digital waveguide. For the parametric control approach, displacement waves are the easiest to use and understand, as they have a very intuitive meaning in plucked string playing.

3. CONTROL

There are several possibilities for the control of the digital waveguide model. One could alter the model parameters, or just apply

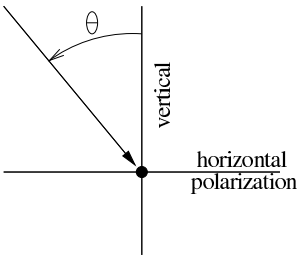

Figure 2: The pluck angle θ

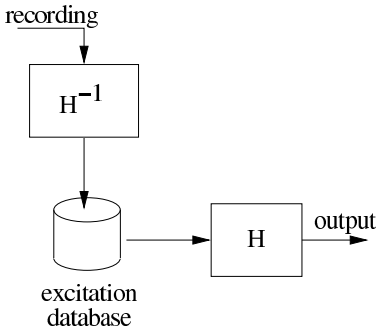

Figure 3: The principle of the Source-Filter approach.

an appropriate input. The input could be precomputed and based on an actual recording, or could be generated as needed, according to a suitable parameterization. Both methods have their specific advantages.

3.1. Source-Filter approach

In the source-filter approach, a database of pluck inputs is used (figure 3). A set of recorded notes played with different techniques is filtered by the inverse of the guitar model and used as input during the synthesis [10]. If the model is an accurate representation of the instrument, the resulting inverse-filtered signal will be a pluck-like signal and can be used to excite the model. This approach has the advantage that a very natural and realistic note attack is obtained, but the disadvantage that the excitation has to be essentially hand-picked from a large database.

3.2. Parametric approach

In the fully parametric approach, the input signal is generated during playback from a set of rules. These include pluck type, pluck position, pluck angle, initial displacement and initial velocity. The parametric approach is better suited for the automatic interpretation of a score. The parameters can be determined such that a particular desired output is obtained. A few pluck types are shown in figure 4.

4. MPEG-4 SAOL

In the MPEG-4 multimedia coding standard, a programming language was defined specifically for algorithmic sound synthesis. The *Structured Audio Orchestra Language* (SAOL) is based on other music programming languages like CSound, and includes

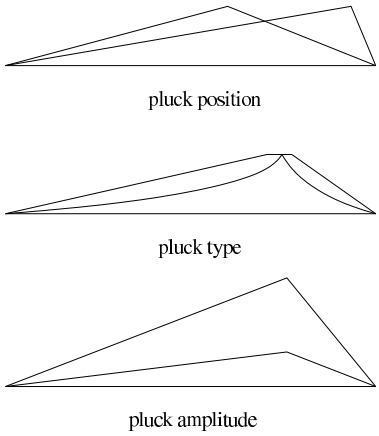

Figure 4: The effect of three control parameters on the pluck shape.

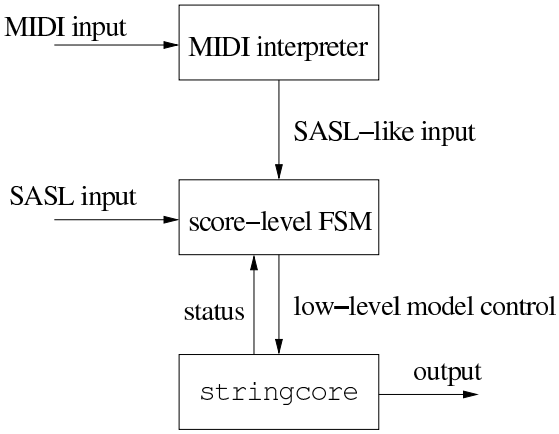

Figure 5: *A complete implementation of a physical model of a guitar consist of three parts: a MIDI interpreter, a score-level Finite State Machine, and the core datapath-like synthesis routine.*

a full set of signal processing primitives (opcodes). It is time-aware, bus-oriented and allows the definition of several virtual *instruments* and user-defined opcodes.

The simplest solution is to use the MPEG-4 native *Structured Audio Score Language* (SASL) as the input format. However, this requires that the composer translates the score into a series of physically relevant inputs, which is very time-consuming. It is much easier to use the standard MIDI format and to rely on a semi-intelligent algorithm to allocate the notes to the strings and frets, and to generate musically acceptable input for the excited strings.

4.1. SASL input

Using SASL allows the programmer to define any control parameter directly. The control structure of figure 5 is used. The guitar model is not accessed directly by the SASL input, but by a score-level Finite State Machine. The FSM translates the physical control parameters like string number, fret number, pluck type, position etc. into a sequence of elementary commands for the digital waveguide model `stringcore`. The `stringcore` routine has a very low-level functionality. The algorithm recognizes com-

Action	**Commands**
damp a string	alter coupling
pluck a string	accept input
select fret	**if** string active
	begin
	set coupling (damp)
	wait 100ms
	end
	rescale delay line
	set coupling for new delay length

Table 1: *Breakdown of the most common SASL-level events into* `stringcore` *low-level control.*

mands to rescale the delay lines, alter the coupling coefficients, or to accept input. Some actions, like a fret change, require a sequence of commands. The most commonly used low-level commands are summarized in table 1.

4.2. MIDI input

MIDI (Musical Instrument Digital Interface) is a communication standard covering both the hardware and software aspects of information interchange between digital musical instruments. The MIDI protocol is an entire music description language in binary form. Each word describing an action of musical performance is assigned a specific binary code. MIDI was designed for keyboards, so many of the actions are percussion oriented. In MIDI, several sets of virtual "controllers" are defined and can be used to transfer performance-related data. MIDI was designed for streaming data, and as a result, note duration is the time difference between a `Note On` and a `Note Off` event, or, the duration of the note is unknown at the time of attack. A typical music score for string instruments, described in MIDI, typically only consists of the `Note On`, `Note Off`, note number, and `velocity` parameters. This is very little information to effectively control the physical model of the instrument.
In the MPEG-4 interpretation of the MIDI standard, every MIDI note received by de orchestra results in a new instance of the instrument requested. The note is then sent to that particular instance, and when the `Note Off` message is received, that instance dies. This is obviously not what we want, as all the notes should be played on the same virtual instrument. To solve this, the MIDI interpreter dynamically creates a unique instance of the instrument, and sets a flag through a global variable to direct all other MIDI notes to that instance. This is schematically represented in figure 7. We consider two different types of MIDI input: multichannel MIDI, where one MIDI channel is used for every string; and single-channel MIDI. With the multichannel MIDI input, the composer has to decide on which string a particular note has to be played. This is a relatively practical solution and gives an extra degree of freedom to the composer, as notes of the same pitch but played on different strings have a different timbre. In the single channel case, the MIDI interpreter receives a sequence of notes to be played, and has to decide on which string every note is played. Due to the streaming nature of MIDI, it is not always true that the optimal solution can be found, as the problem is inherently non-causal. A human player reads the future notes to be played and makes a decision based on the minimal movement of the left hand. A very simple string/fret allocation algorithm tries to minimize the fret over all strings (in other words, play in low positions), taking into account the status of each strings.

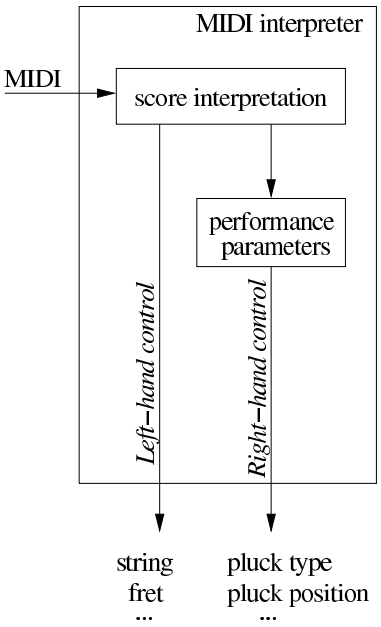

Figure 6: *The MIDI interpreter generates SASL-like output for both the left hand control parameters (string number, fret), and the right hand control parameters (pluck type, position, angle, amplitude, velocity, etc.).*

$$\forall s, \text{fret}_s = \frac{\log(f_{\text{note}}/f_{0,s})}{\log \sqrt[12]{2}}. \quad (3)$$

Of all strings where $0 \leq \text{fret}_s \leq 25$, select the one that is not active, and play the note on that string. If no string is found, select the string with $\min(\text{fret}_s)$, damp it and play the note. For relatively simple scores, most of the time the right string is chosen, but is easy to find a piece where this does not work at all (e.g. *Leyenda* by Albéniz, where a sequence of same-pitched notes has to be played on different strings).

5. CONCLUSIONS

The control of a physical model is both easy and complicated. The main advantage for a human player is that sensor inputs are directly mapped to the parameters controlling the model. If the sound has to be synthesized based on a score, the situation is more complex, and artificial input has to be generated. Part of this task can be delegated to the composer of the piece when either SASL or multichannel MIDI are used. When the usual single-channel MIDI is used, a string/fret allocation algorithm is needed, which is in essence a player model. The actual input of the model could be either selected from an excitation signal database or could be generated during synthesis by the use of a parametric input model. The most practical and compact solution is the combination of a single-channel MIDI input interpreter and a parametric pluck generator. The best sound quality and musical expression is however achieved with a multichannel MIDI model, excited with samples out of a recomputed database.

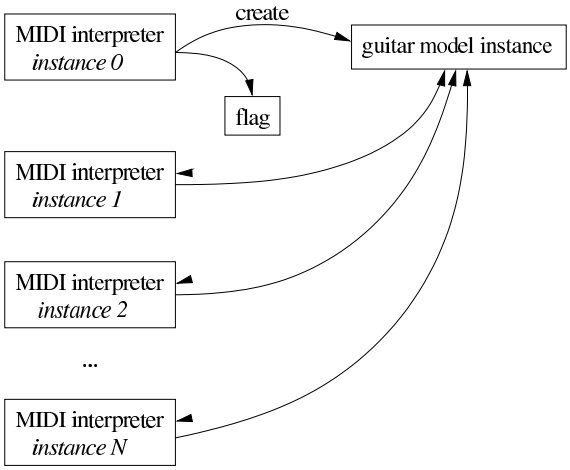

Figure 7: *The first instance of the MIDI interpreter creates a dynamic instance of the guitar model. All subsequent instances of the interpreter direct their output to that particular instrument instance.*

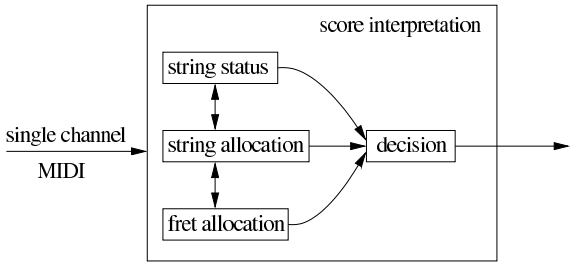

Figure 8: *The score interpretation part of the MIDI interpreter decides on which string and which fret the incoming MIDI note will be played.*

6. ACKNOWLEDGMENTS

Axel Nackaerts is a Research Assistant with the I.W.T. (Flemish Institute for Scientific and Technological Research in Industry) and can be contacted at tel. (+32) 16-321800 fax (+32) 16-321970, email `Axel.Nackaerts@esat.kuleuven.ac.be` and mail: ESAT, Kasteelpark Arenberg 10, B-3001 Leuven. Bart De Moor is full professor at the K.U.Leuven. Rudy Lauwereins is professor extra-ordinary at the K.U.Leuven. This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven. This Work is supported by: the Flemish Government: Research Council KUL (GOA Mefisto-666), FWO: (G.0240.99, G.0407.02, G.0197.02, G.0141.03, G.0491.03, and G.0120.03, Research communities ICCoS, ANMMM, PhD and postdoc grants), AWI: Bil. Int. Research Program; IWT (PhD grants, Soft4s); EU: Eureka 2063-Impact, Eureka 2419-FLiTE, CAGE, ERNSI; Federal State DWTC (IUAP IV-02, IUAP V-22), PODO-II (CP/40); Industrial contract research (IPCOS, Electrabel, Elia, Data4s, LMS, VIB). The scientific responsibility is assumed by the authors.

7. REFERENCES

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

[2] Axel Nackaerts, Bart De Moor, and Rudy Lauwereins, "Coupled string guitar models," in *Mathematics and Simulation with Biological, Economical and Musicoacoustical Applications*, C.E. D'Attellis, V.V. Kluev, and N.E. Mastorakis, Eds. 2001, pp. 289–294, WSES Press.

[3] M. R. Schroeder, "New method for measuring reverberation times," *J. Acoust. Soc. Amer.*, vol. 37, pp. 232–235, 1965.

[4] J. Laroche, "A new analysis/synthesis system of musical signals using prony's method. application to heavily damped percussive sounds," in *Proc. IEEE ICASSP-89*, Glasgow, May 1989, pp. 2053–2056.

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

[6] Axel Nackaerts, Bart De Moor, and Rudy Lauwereins, "Parameter estimation for dual-polarization plucked string models," in *Proc. of the International Computer Music Conference (ICMC 2001)*, 2001, La Habana, Cuba.

[7] Gabriel Weinreich, "Coupled piano strings," *Journal of the Ac. Soc. of Am.*, vol. 62, no. 6, pp. 1474–1484, December 1977.

[8] Axel Nackaerts, Bart De Moor, and Rudy Lauwereins, "Parameter estimation for coupled string models," Tech. Rep. TR 02-01, Dept. Elektrotechniek, ESAT-SCD, Katholieke Universiteit Leuven, 2002, ftp://ftp.esat.kuleuven.ac.be/pub/sista-/nackaerts/reports/report_JAES02.ps.gz.

[9] Axel Nackaerts, Bart De Moor, and Rudy Lauwereins, "Measurement of guitar string coupling," in *Proc. of the International Computer Music Conference (ICMC 2002)*, Göteborg, Sweden, September 2002, pp. 321–324.

[10] 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, no. 5, pp. 331–353, 1996.