# SIMULATING IDIOMATIC PLAYING STYLES IN A CLASSICAL GUITAR SYNTHESIZER: RASGUEADO AS A CASE STUDY

*Mikael Laurson, Vesa Välimäki and Henri Penttinen,*

Department of Signal Processing and Acoustics,
Aalto University
Espoo, Finland
laurson@siba.fi

## ABSTRACT

This paper presents our research efforts to synthesize complex instrumental gestures using a score-based control scheme. Our specific goal is to simulate the rasgueado technique that is popular especially in flamenco music. This technique is also used in the classical guitar repertoire. Rasgueado is especially challenging as ordinary music notation is not adequate to represent the dense stream of notes required for a convincing simulation. We will take two approaches to realize our task. First, we use the practical knowledge of how the actual performance is accomplished by the human player. A second, complementary, approach is to analyze an excerpt from real guitar playing. Our main focus here is to extract the onset times and the amplitudes of the recoded gesture. Next we combine the results from the two analysis steps using a constraint-based approach to find possible pitch and fingering sequences. Finally we translate the findings to our macro-note scheme that allows us to fill algorithmically a musical score.

## 1. INTRODUCTION

Synthesizers use currently two basic approaches to simulate the classical guitar. First, several sample-based realizations can be found as commercial packages. The sound quality with this approach can be very high. The problem here is however that it is almost an impossible task to record the instrument completely. As a result the best available sample-based guitar synthesizers have lately become extremely large (for instance the size of the classical guitar sample instrument offered by Vienna Symphonic Library is already over 3.5 GB). Besides the unpractical size problem the user has also to face the increasing complexity of how to control the instrument. Our main focus in this paper–the rasgueado technique where the player hits the strings with the right-hand nails in very rapid fashion–is a gesture that is hard to use in a musical performance situation when limited only to prerecorded sound material.

The second approach is to synthesize the strings, the body and the interaction of the player using various physics-based synthesis methods. Our approach in this article is the digital waveguide modeling technique [1], [2], [3]. The modeling of the instrument results in realizations that are more economic size-wise than the sample-based approach. The control of these instruments is also more straightforward as physics-based instruments typically have a state and thus can react autonomously to incoming control information.

Instrumental gestures can be studied with several techniques. The traditional and informal way is to observe closely how the player performs the gesture (a typical example is the teacher/student relationship where the student tries to mimic the teacher as closely as possible). Recently other techniques have become available. The performer can be interviewed, filmed, recorded and/or analyzed by using different types of sensors that allow one to record the physical body movements of the performer. A review on these techniques can be found in [4].

This paper utilizes two complementary techniques: first we use the practical knowledge of the performer how the gesture is performed (Sections 2 and 4). Second we use an audio recording of the same gesture and try to extract from the signal important aspects that deal with the timing and amplitude of the individual attacks of the rasgueado effect (Section 3). After this we use our findings to re-synthesize the original signal (Section 5). For the re-synthesis phase we will add new features to our guitar synthesizer in order to be able to simulate the rasgueado technique. We end with a discussion how we can extend the ideas so that the rasgueado effect could be used in different musical contexts (Section 6).

## 2. HOW HUMANS REALIZE RASGUEADO GESTURES

Rasgueado means 'strummed' in Spanish, but in English the word is usually used only to denote the rhythmically complex kind of strumming that is characteristic of flamenco and classical guitar. In the following we use the classical guitar fingering notation system for the plucking hand, known as 'pimac', where p (pulgar, thumb), i (indio, index finger), m (medio, middle finger), a (annular, ring finger), and c (chiquita, little finger). The simplest rasgueado is performed from the basic right-hand position using the c, a, m, and i, or 'cami' for short, fingers (thus we leave the thumb away).

1) The fingers are curled into the palm of the hand.

2) The little (c) finger is allowed to fall downwards across the strings one by one from bass strings to top strings, followed in turn by the a, m and i fingers, thus producing four strummed chords. This gesture could be notated as follows:

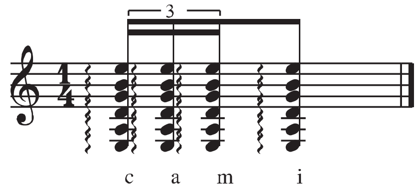


Figure 1: *Rasgueado notation.*

This notation is highly simplified and cannot be used directly in simulation contexts. A more realistic notation can be found in Figures 5 and 6 where the individual upward strums are overlapping creating a complex wash of sound. There are obviously numerous rasgueado variations where the player could use the thumb, the right-hand fingers could strike the strings both upwards and downwards, the gesture could be longer resulting in a continuous roll, etc.

## 3. ANALYSIS OF A RECORDED GESTURE

In this section we analyze a recorded section that contains the opening chord from Fantasia Sevillianas by Joaqiun Turina. The piece begins with a simple rasgueado gesture similar to the one shown in Figure 1. We analyze the onset time and amplitude of the individual events caused by the right-hand nails of the player.

Figure 2(a) shows the waveform of the recording, which has been made in the anechoic chamber. The sample rate is 44.1 kHz. We can infer that at most 24 events (four fingers times six strings) should be extracted from the signal of Figure 2(a). It is possible that the number of events is smaller than that, because the performer does not necessarily manage to hit every string using the right-hand 'cami' fingers. Furthermore, two or more events could occur practically simultaneously, which would make it difficult for the signal analysis process to separate them.

It is apparent that onsets cannot be directly extracted from the signal of Figure 2(a). The string vibrations starting after each onset overlap with following events. To avoid the disturbing effect of string vibration, the signal is filtered with an FIR bandpass filter of order 300, which has its passband between 11 kHz and 20 kHz. In this frequency region, the partials of guitar tones are weak and decay very fast. However, the attack transients caused by nail-string contacts are wideband signals, which appear as short events even at high frequencies [5]. Figure 2(b) shows the filtered signal. It is seen that now many transients can be clearly observed, since harmonics have been suppressed. Inspection of the filtered signal reveals that there are no events in the first 30 ms, which contains only hiss. Additionally, apparently there are no events after about 320 ms. In the following we will focus on the time span from 30 ms to 330 ms.

We first compress the filtered signal to reduce the level difference between the softest and strongest onsets. The signal envelope is estimated by first full-wave rectifying the signal (using the ABS function) and by smoothing it with a moving average filter of length 220 samples (5 ms). The signal is then divided sample-by-sample by the envelope function to cancel the envelope variations. Furthermore, the local tall signal peaks are clipped with a soft limiter (a hyperbolic tangent waveshaper). The resulting signal is again scaled so that its absolute maximum value is one. Figure 3(a) shows that the level fluctuations have been reduced.

Next, the energy of the compressed signal is computed. Figure 3(b) displays the energy curve, which has been obtained by squaring and by applying a short moving average filter. Here the length of the averaging window is only 44 samples (1 ms) to allow detection of closely located onsets in the next processing stage.

Finally, onset events are detected by thresholding the energy curve. The threshold is set to an appropriate level so that 24 highest peaks (the expected number) get detected. The points of the energy curve where the tallest peaks start to rise determine the onset times. The corresponding amplitude of each event is read from the envelope of the original signal at those time locations. Figure 4 presents the original signal together with the detected onsets (amplitude vs. time). These data are used in the following to develop a macro-note scheme that automatically generates similar gestures.

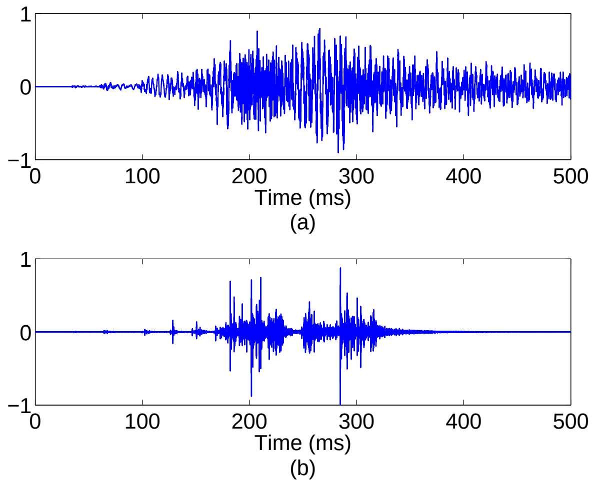


Figure 2: *(a) The recorded rasgueado signal and (b) its filtered version (11 kHz ... 20 kHz)l.*

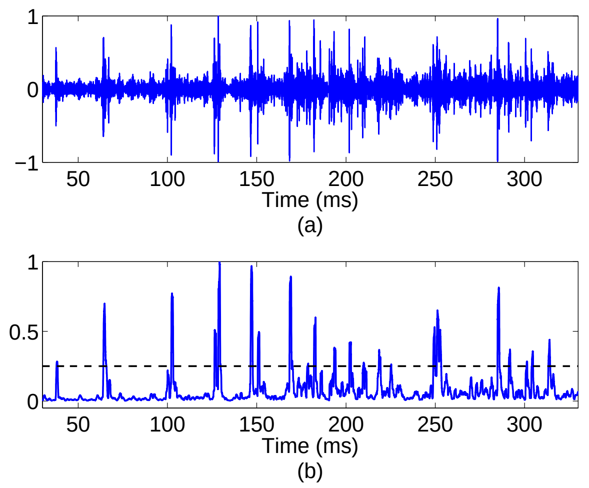


Figure 3: *The compressed signal and (b) its local energy (solid line) and the detection threshold at 0.25 (dashed line).*

## 4. SCORE-BASED SIMULATION OF RASGUEADO

The analysis presented in the previous section results in a flat list of onset times and amplitudes (see Figure 4). Next we will distribute this information to the right-hand 'cami' fingers in order to be able to simulate the rasgueado gesture (see Section 2). We also know that each finger strums one by one the strings from bass to top. Thus we need to associate each onset to a 'cami' right-hand finger. This kind of a problem can be solved using a constraint-based approach where the search-space consists of a list of variables where each onset is combined either with the 'c', 'a', 'm',

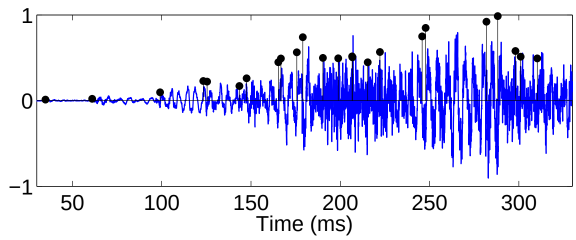


Figure 4: *The original signal (solid lines) and the detected note onset (dots). The onset amplitudes have been looked up from the envelope estimate, and are scaled by 3.14 for visualization.*


Figure 5: *One possible four-part realization from the constraint solver. Each 'cami' right-hand finger has a dedicated part.*

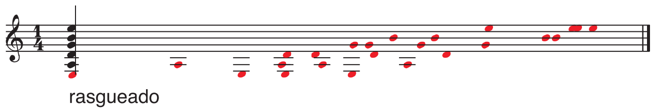


Figure 6: *The final one-part realization used to control the guitar synthesizer. The black notes with the label 'rasgueado' are not played and they give the harmonic skeleton of the gesture. The red notes without stems, in turn, are realized by the synthesizer resulting in a rasgueado gesture.*

or 'i' fingers. After finding the fingerings we can infer the missing pitch information based on our knowledge of how the gesture has been realized (i.e. the first 'c' event is played on the sixth string, the second 'c' event on the fifth string, and so on). The result, in turn, can be represented as a four-part score (see Figure 5), where the first part is played with the 'c' finger, the second part with 'a', the third part with 'm', and the fourth part with 'i' (see the labels in front of the staff system).

The tools presented in Sections 4-6 (i.e. the constraint solver, the music notation based control system, and the sound synthesis engine) are all inside our visual programming environment PWGL [6].

The constraint-based approach is used here as it allows us to use rules in a modular and interactive way. By adding gradually more rules to the system the user can observe how the solutions produced from the raw material take shape in an interesting and meaningful way. Some of the rules are quite simple and obvious (for instance when the sequence starts the following rules apply: the first onset has to played with the 'c' finger; no 'm' finger before 'a'; no 'i' finger before 'm' and 'a'). Some rules are more complex and deal with how the fingers are distributed in the whole sequence (for instance: disallow more than two adjacent equal fingers: thus there should never be more than two, say, 'c's in a row). Typically the constraints solver finds several results which are equally valid. This in turn can be used to good effect as it allows one to have several realizations of the rasgueado gesture.

The four-part score is finally converted to our macro-note scheme [7]: now each note is associated with fingering and pitch information (Figure 6). This score is used to generate control information for our classical guitar synthesizer. In the next section we briefly discuss the specific additions to our synthesizer that were made in order to gain more convincing results.

## 5. SYNTHESIZING THE RASGUEADO GESTURE

The synthesis part of the rasgueado simulation is challenging due to the interaction of the right-hand nails of the player with the strings of the instrument. Preliminary synthesis tests showed us a difference in the attack of the synthesis. When using an ordinary pluck excitation the synthesis results in a too soft attack. Thus, we recorded a special set of excitations where the player hits the string with the right-hand nails outwards (in an ordinary pluck the player moves the finger inwards), resulting in a much more brighter and noisy sound than with an ordinary pluck.

The guitar synthesis algorithm is based on six single-delay loop (SDL) digital waveguide (DWG) [8] strings. The parameters for the excitation and filter blocks in the string model were obtained as follows. First, the inharmonicity of the tones were determined with the automatic method proposed in [9] and the outliers were fine-tuned manually. The excitation pulse was obtained by subtracting the inharmonic partials of a recorded tone using sinusoidal modeling and by filling the dips in the spectrum by adding the attack part of an inverse filtered sinusoidal model, as described in [10]. The overall decay behavior and frequency dependent decay were modeled with a first order one-pole filter H(z) [10]. The transfer function is $H(z) = b/(1 + az^{-1})$, where $b = g(1 + a)$. The $g$ and $a$ parameters of the filter control the overall decay and frequency dependent decay, respectively. The string is tuned with a third-order Lagrange filter applying the fractional part of the delay [11].

In addition, we noticed that we had to adjust the amplitude envelope that simulates the damping when the player touches a string that is already vibrating. In normal playing style the envelope scales the loop filter gain, $g$, so that it drops to zero. Now, when simulating the rasgueado effect, the envelope of the loop gain is decreased to 90% of its full value before the synthesis model is excited again. This way the string is only partially damped, mimicking the continuous sound in real rasgueado playing.

## 6. VARIATIONS AND RESULTS

The result found in Figure 6 is our attempt to simulate closely the original recoded rasgueado gesture. In this section we discuss some ideas how we can reuse this material in different musical contexts. One simple and quite effective way to have slight variations of the original material is to scale the offset times. This results, especially when combined with amplitude modifications, in more sharp or more broad and gentle versions. We can also make slight variations by choosing different results from the constraint solver (see Section 4).

One idea is to simulate a continuous roll that is realized by playing the gesture several times but at each iteration the offset time is incremented resulting in a sequence of overlapping gestures. To make the result less mechanical we scale the speed and amplitude contour of each gesture. The audio results related to Sections 3-6 can be found in the following webpage: http://www.acoustics.hut.fi/go/dafx10-rasgueado/.

Our final simulation example is the guitar part of the first movement Concierto de Aranjuez. The Concierto de Aranjuez is a composition for classical guitar and orchestra by the Spanish composer Joaquin Rodrigo. The Guitar part excerpt is given in Figure 7. The realization of this example can be found in the PWGL home page: www.siba.fi/pwgl/pwglsynth.html. This example demonstrates how we can combine ordinary guitar playing (e.g. plain chords, scales) with complex rasgueado gestural information. All chords labelled as 'rasgueado' are extended by the macro-note scheme (see Figure 7).

## 7. CONCLUSIONS

This article presents some recent developments that allow us to simulate rasgueado gestures within a musical performance context. We approach the problem from two points of view. First, we discuss how the performer executes the effect. Second, we analyze a recorded rasgueado excerpt in order to get the onset times and amplitudes of the individual attacks. This information is combined resulting in a re-synthesis of the original gesture. Finally we give several examples how the control material can be reused in different musical contexts.

The rasgueado technique belongs to one of the most challenging tasks when simulating more complex playing styles in guitar playing. We believe that this kind of work presented in this article is essential if the target is to go beyond isolated notes in guitar synthesis. Future work includes more systematic analysis of larger musical excerpts that are utilizing a broader repertoire of rasgueado playing styles. Furthermore, the improvement of the synthesis part should include also a more careful analysis of the damping that occurs when the right-hand nails are touching a vibrating string.

## 8. ACKNOWLEDGEMENTS

This work has been financed by the Academy of Finland (project no. 122815 and no. 128689).

## 9. REFERENCES

[1] J. O. Smith, "Physical Modeling Using Digital Waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, 1992.

[2] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Janosy, "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.

[3] M. Laurson, C. Erkut, V. Välimäki, and M. Kuuskankare, "Methods for modeling realistic playing in acoustic guitar synthesis," *Computer Music Journal*, vol. 25, no. 3, pp. 38–49, 2001.

[4] R. I. Godoy and M. Leman, Eds., *Musical Gestures Sound, Movement, and Meaning*, Routledge, Taylor & Francis Group, New York and London, 2010.

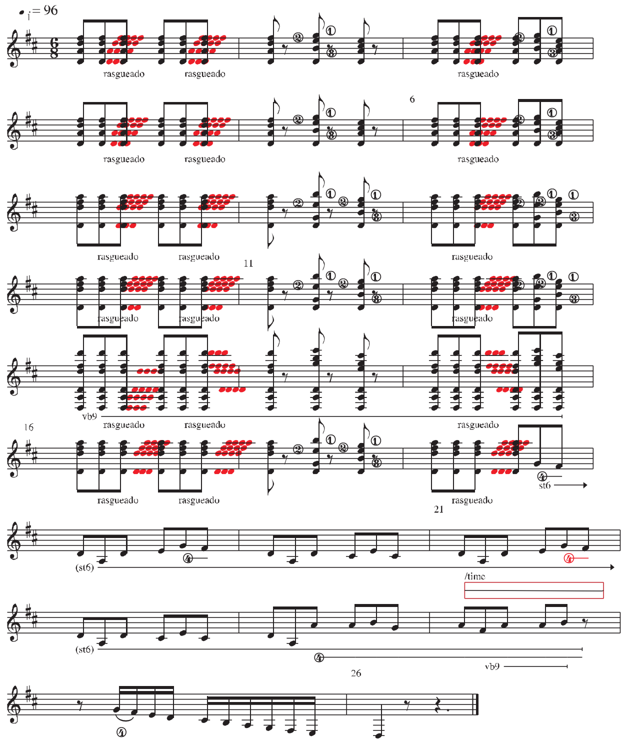


Figure 7: *Concierto de Aranjuez. This example contains macro-note rasgueado effetcs, ordinary chords, and scales. The red notes have been generated by our macro-note scripting language.*

[5] S. Hainsworth, *Beat tracking and musical metre analysis*, in Signal Processing Methods for Music Transcription, A. Klapuri and M. Davy, Eds., Springer, pp. 101-129, 2006.

[6] M. Laurson, M. Kuuskankare, and V. Norilo, "An Overview of PWGL, a Visual Programming Environment for Music," *Computer Music Journal*, vol. 33, no. 1, pp. 19–31, 2009.

[7] M. Laurson and M. Kuuskankare, "Towards Idiomatic and Flexible Score-based Gestural Control with a Scripting Language," in *Proceedings of NIME'08 Conference*, Genova, Italy, 2008, pp. 34–37.

[8] 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.

[9] J. Rauhala, H.-M. Lehtonen, and V. Välimäki, "Fast automatic inharmonicity estimation algorithm," *Journal of the Acoustical Society of America*, vol. 121, no. 5, pp. EL184–EL189, 2007.

[10] V. Välimäki and T. Tolonen, "Development and calibration of a guitar synthesizer," *J. Audio Eng. Soc.*, vol. 46, no. 9, pp. 766–778, 1998.

[11] 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, no. 1, pp. 30–60, 1996.