PHONETIC-BASED MAPPINGS IN VOICE-DRIVEN SOUND SYNTHESIS

Jordi Janer and Esteban Maestre
Music Technology Group, Pompeu Fabra University, Barcelona

Keywords: Singing-driven interfaces, Phonetics, Mapping, Sound synthesis.

Abstract: In voice-driven sound synthesis applications, phonetics convey musical information that might be related to the sound of an imitated musical instrument. Our initial hypothesis is that phonetics are user- and instrument-dependent, but they remain constant for a single subject and instrument. Hence, a user-adapted system is proposed, where mappings depend on how subjects performs musical articulations given a set of examples. The system will consist of, first, a voice imitation segmentation module that automatically determines note-to-note transitions. Second, a classifier determines the type of musical articulation for each transition from a set of phonetic features. For validating our hypothesis, we run an experiment where a number of subjects imitated real instrument recordings with the voice. Instrument recordings consisted of short phrases of sax and violin performed in three grades of musical articulation labeled as: staccato, normal, legato. The results of a supervised training classifier (user-dependent) are compared to a classifier based on heuristic rules (user-independent). Finally, with the previous results we improve the quality of a sample-concatenation synthesizer by selecting the most appropriate samples.

1 INTRODUCTION

Technology progresses toward more intelligent systems and interfaces that adapt to users' capabilities. New musical applications are not exempt of this situation. Here, we tackle singing-driven interfaces as an extension in the musical domain of speech-driven interfaces. Most known example of singing-driven interfaces is query-by-humming (QBH) systems, e.g. (Lesaffre et al., 2003). In particular, we aim to adapt the mappings depending on the phonetics employed by the user in instrument imitation (syllabling). In this paper, singing is used to control the musical parameters of an instrument synthesizer (Maestre et al., 2006). Results may lead to the integration of such learned mappings in digital audio workstations (DAW) and music composition software.

1.1 Voice-driven Synthesis

Audio and voice-driven synthesis has been already introduced by several authors. In (Janer, 2005), the author carried out a voice-driven bass guitar synthesizer, which was triggered by impulsive voice utterances that simulated the action of plucking. Here, we aim to extend it to continuous-excitation instrument, which permits more complex articulations (i.e. note-to-note transitions). To derive control parameters from the voice signal becomes thus more difficult than detecting voice impulses. As we describe in this paper, phonetics appears to be a salient attribute for controlling articulation.

Research in state-of-the-art sound synthesis takes two main directions: more realism in sound quality, and a more expressive control. For the former, basically, most current commercial synthesizers use advanced sample based techniques (Bonada and Serra, 2007; Lindemann, 2007). These techniques provide both quality and flexibility, achieving a realism missing in early sample-based synthesizers. Secondly, in term of expressive control, synthesizers make use of new interfaces such as gestural controllers (Wanderley and Depalle, 1999), indirect acquisition (Egozy, 1995), or alternatively, artificial intelligence methods to induce a human-like quality to a musical score (Widmer and Goebl, 2004).

In the presented system, the synthesizer control parameters involve *loudness*, *pitch* and *articulation type*. We extract this information from the input voice

Janer J. and Maestre E. (2007).
PHONETIC-BASED MAPPINGS IN VOICE-DRIVEN SOUND SYNTHESIS.
In *Proceedings of the Second International Conference on Signal Processing and Multimedia Applications*, pages 109-115
DOI: 10.5220/0002141401090115

signal, and apply the mappings to the synthesizer controls, in a similar manner to (Janer, 2005) but here focusing on note-to-note articulations. The synthesis is a two-step process: sample selection, and sample transformation.

1.2 Toward User-adapted Mappings

We claim that the choice of different phonetics when imitating different instruments and different articulations (note-to-note transitions) is subject-dependent. In order to evaluate the possibilities of automatically learning such behaviour from real imitation cases, we carry out here some experiments. We propose a system consisting of two main modules: an *imitation segmentation module*, and an *articulation type classification module*. In the former, a probabilistic model automatically locates note-to-note transitions from the imitation utterance by paying attention to phonetics. In the latter, for each detected note-to-note transition, a classifier determines the intended type of articulation from a set of low-level audio features.

In our experiment, subjects were requested to imitate real instrument performance recordings, consisting of a set of short musical phrases played by saxophone and violin professional performers. We asked the musicians to perform each musical phrase using different types of articulation. From each recorded imitation, our *imitation segmentation module* automatically segments note-to-note transitions. After that, a set of low-level descriptors, mainly based on cepstral analysis, is extracted from the audio excerpt corresponding to the segmented note-to-note transition. Then, we perform supervised training of the *articulation type classification module* by means of machine learning techniques, feeding the classifier with different sets of low-level phonetic descriptors, and the target labels corresponding to the imitated musical phrase (see figure 1). Results of the supervised training are compared to classifier of articulation type based on heuristic rules.

2 IMITATION SEGMENTATION MODULE

In the context of instrument imitation, singing voice signal has a distinct characteristics in relation to traditional singing. The latter is often referred as *syllabling* (Sundberg, 1994). For both, traditional singing and syllabling, principal musical information involves pitch, dynamics and timing; and those are independent of the phonetics. In vocal imitation, though, the role of phonetics is reserved for determining articulation and timbre aspects. For the former, we will use phonetics changes to determine the boundaries of musical articulations. For the latter, phonetic aspects such as formant frequencies within vowels can signify a timbre modulation (e.g. brightness). We can conclude that unlike in speech recognition, a phoneme recognizer is not required and a more simple classification will fulfill our needs.

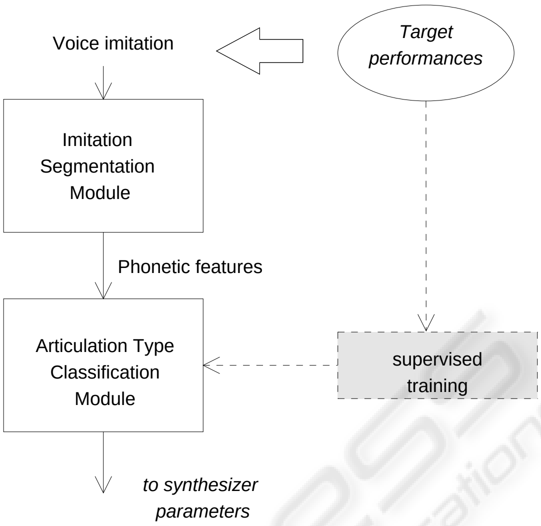
Voice imitation
Target performances
Imitation Segmentation Module
Phonetic features
Articulation Type Classification Module
supervised training
to synthesizer parameters

Figure 1: Overview of the proposed system. After the imitation segmentation, a classifier is trained with phonetic low-level features and the articulation type label of target performance.

In Phonetics, one can find various classifications of phonemes depending on the point of view, e.g. from the acoustic properties the articulatory gestures. A commonly accepted classification based on the acoustic characteristics consists of six broad phonetic classes (Lieberman and Blumstein, 1986): vowels, semi-vowels, liquids and glides, nasals, plosive, and fricatives. Alternatively, we might consider a new phonetic classification that better suits the acoustic characteristics of voice signal in our particular context. As we have learned from section 2, a reduced set of phonemes is mostly employed in syllabling. Furthermore, this set of phonemes tends to convey musical information. Vowels constitute the nucleus of a syllable, while some consonants are used in note onsets (i.e. note attacks) and nasals are mostly employed as codas. Our proposal envisages different categories resulting from the previous studies in syllabling (Sundberg, 1994). Taking into account syllabling characteristics, we propose a classification based on its musical function, comprising: *attack*, *sustain*, *release*, *articulation* and *other* (additional).

Table 1: Typical broad phonetic classes as in (Lieberman and Blumstein, 1986), and proposed classification for syllabling on instrument imitation. This table comprises a reduced set of phonemes that are common in various languages.

CLASS	PHONEMES
Speech Phon. classes	
Vowels	[a] , [e] , [i], [o], [u]
Plosive	[p], [k], [t], [b], [g], [d]
Liquids and glides	[l], [r], [w], [y]
Fricatives	[s], [x],[T], [f]
Nasal	[m], [n],[J]
Syllabling Phon. classes	
Sustain	[a] , [e], [i], [o], [u]
Attack	[p], [k], [t], [n], [d]
Articulation	[r], [d], [l], [m], [n]
Release	[m], [n]
Other (additional)	[s],[x],[T], [f]

2.1 Method Description

Our method is based on heuristic rules and looks at the timbre changes in the voice signal, segmenting it according to the phonetic classification mentioned before. It is supported by a state transition model that takes into account the behavior in instrument imitation. This process aims at locating phonetic boundaries on the syllabling signal. Each boundary will determine the transition to one of the categories showed in table 1. This is a three steps process:

1. *Extraction of acoustic features.*
2. *Computation of a probability for each phonetic class based on heuristic rules.*
3. *Generation of a sequence of segments based an a transition model (see Fig. 3)*

Concerning the feature extraction, the list of low-level features includes: energy, delta energy, Mel-Frequency Cepstral Coefficients (MFCC), deltaMFCC, pitch and zero-crossing. *DeltaMFCC* is computed as the sum of the absolute values of the MFCC coefficients derivative (13 coeffs.) with one frame delay. Features are computed frame by frame, with a window size of 1024 and a hop size of 512 samples at 44100Hz. This segmentation algorithm is designed for a real-time operation in low-latency conditions.

From the acoustic features, we use a set of heuristic rules to calculate boundary probabilities for each phonetic class. Unlike for an offline processing, in a real-time situation, this algorithm is currently not able to distinguish between *Articulation* and *Release* phonetic classes.

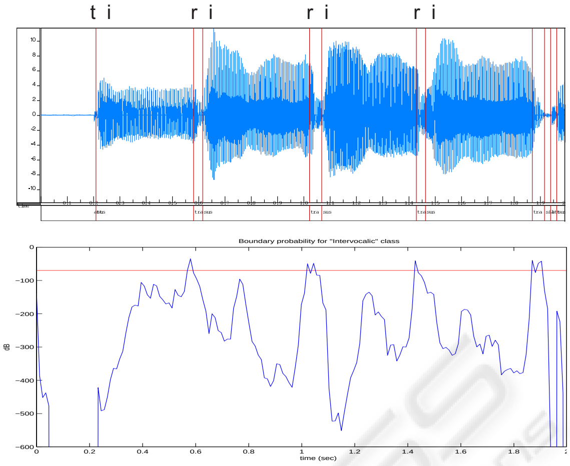

Figure 2: Syllabling Segmentation (from top to bottom): phonemes, waveform, labels and boundary probability for *intervocalic* class (horizontal line representing the threshold b_{thres}).

In a first step, in order to generate continuous probabilities, and to attain a more consistent behaviour, we employ gaussian operators to compute a cost probability $f_i(x_i)$ for each voice feature x_i(see Eq. 1). Observe that for each voice feature x_i, function parameters μ_i, σ_i and T_i are based on heuristics. In the table 2, we list the voice features used for the six considered boundary categories $B_j, j = \{0 \ldots 5\}$. Then, for each boundary probability B_j, a weighted product of all voice feature probabilities is computed, with $w_i = 1$ or $w_i = 1/f_i(x_i)$, whether a given phonetic class j is affected by a voice feature i.

$$f_i(x_i) = \begin{cases} \exp \frac{(x_i - \mu_i)^2}{2\sigma_i^2}, x_i > T_i \\ 1, \qquad\qquad x_i \leq T_i \end{cases} \tag{1}$$

$$B_j = \prod_i w_i \cdot f_i(x_i) \tag{2}$$

This is a frame-based approach, computing at each frame k a boundary probability for each phonetic class j, $p_j(x[k]) = p(B_j|x[k])$. At each frame, to decide if a boundary occurs, we take the maximum of all four probabilities $p(B|x[k])$ and compare it to a empirically determined threshold b_{thres}.

$$p(B|x[k]) = \max_{0<5} [p_j(x[k])]$$

Finally, in order to increase robustness when determining the phonetic class of each segment in a sequence of segments, we use a state transition model. The underlying idea is that a note consists of an onset, a nucleus (vowel) and a coda. In addition, a group of notes can be articulated together, resembling *legato* articulations on musical instruments. Thus, we need

Table 2: Description of the attributes use in the boundaries probability for each category. B_j is the boundary probability for the class j; x_i are the voice features.

j	B_j	x_i
0	*Attack*	energy, dEnergy
1	*Sustain*	energy, dEnergy, dMFCC, zcross
2	*Articulation*	dEnergy, dMFCC, zcross, pitch
3	*Release*	dEnergy, dMFCC, zcross, pitch
4	*Other*	zerocross, dMFCC
5	*Silence*	energy, dEnergy, pitch

Table 3: Averaged results of the onset detection compared to a ground-truth collection of 94 files. The average time deviations was -4.88 ms.

	Mean	Stdev
Correct detections (%)	90.78	15.15
False positives (%)	13.89	52.96

to identify these grouped notes, often tied with liquids or glides. The figure 3 describes the model for boundary transitions.

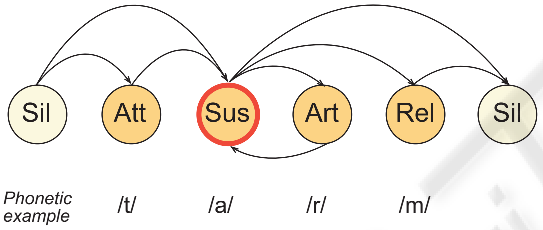

Figure 3: Model for the segment to segment transition for the different phonetic classes.

2.2 Evaluation

With the proposed method, we are able to segment effectively phonetic changes and to describe a voice signal in the context of instrument imitation as a sequence of segments. An evaluation of the algorithm was carried out, by comparing automatic results with a manual annotated ground truth. The ground truth set consists of 94 syllabling recordings. Syllabling examples were voice imitations by four subjects of sax recordings with an average duration of 4.3*sec*. For the evaluation, onsets are considered those boundaries labeled as *sustain*, since it corresponds to the beginning of a musical note. The averaged results for the complete collection is shown in table 3.

3 ARTICULATION TYPE CLASSIFICATION MODULE

The mapping task aims to associate phonetics to different type of musical articulations. Although, we envisage three types of musical articulations: 1) *silence-to-note*, 2) *note-to-note* and 3) *note-to-silence*, this paper focuses only on *note-to-note* transitions. Since, phonetics are assumed to be user-dependent, our goal is to automatize this process by learning the phonetics employed by a particular user. In a real application, this would be accomplished during a user configuration stage. We compare the supervised training results to a user-independent classifier based on heuristic rules.

3.1 Experiment Methodology

For the voice imitation performances, we asked four volunteers with diverse singing experience to listen carefully to target performances and to imitate those by mimicking musical articulations. The supervised training takes the *articulation label* of a target performances, and a *voice imitation* performance. Target performances are sax and violin recordings, in which performers were asked to play short phrases in three levels of articulation.The number of variations is 24, covering:

- *articulation* (3): legato, medium and staccato.
- *instrument* (2): sax and violin.
- *inter-note interval* (2): low and high.
- *tempo* (2): slow and fast.

All target performance recordings were normalized to an average RMS, in order to let subjects concentrate on articulation aspects. Subjects were requested to naturally imitate all 24 variations with no prior information about the experiment goals. Variations were sorted randomly in order to avoid any strategy by subjects, and this process was repeated twice, gathering 48 recordings per subject.

In the Table 4, we can observe the results of user-dependent supervised training for the four subjects, using two (*staccato* and *legato*) and three (*staccato*, *normal* and *legato*) classes for articulation type. The classification algorithm used in our experiments was the J48, which is included the WEKA data mining software [1]. Due to the small size of our training set, we chose this decision-tree algorithm because of its interesting properties. Namely, due to its simplicity, this algorithm is more robust to over-fitting than other

[1] `http://www.cs.waikato.ac.nz/~ml/weka/`

more complex classifiers. The attributes for the training include phonetic features of note-to-note transitions. Three combinations of phonetic features within a transition were tested: 1) MFCC(1-5) of the middle frame; 2) MFCC(1-5) of the left and right frames; and 3) difference of the left and right MFCC frames to the middle frame.

In addition, we present also in Table 4 the results of a user-independent classifier (2 classes) based on heuristic rules. The rules derive from the boundary information from the *imitation segmentation module*. When a note onset is preceded by a *articulation* segment, then it is classified as legato. We observe in the table 5 that the mean percentage of correctly classified instances using different phonetic features as input attributes, and in the last row the results using heuristic rules.

3.2 Discussion

In a qualitative analysis of the imitation recordings, we observed that phonetics are patently user-dependent. Not all subjects were consistent when linking phonetics to articulation type on different target performances. Moreover, none of the subjects were able to distinguish three but only two types of articulation in the target performances (staccato and normal/legato).

From the quantitative classification results, we can also extract some conclusions. Similar results were obtained classifying in two and three classes, when compared to the baseline. When looking at the dependency on the imitated instrument, better performance is achieved by training a model for each instrument separately. It indicates some correspondence between imitated instrument and phonetics. Concerning the set of phonetic features used as input attributes for the classifier, results are very similar (see table 5). The heuristic-rule classifier uses the output of the imitation segmentation module. If a silence segment is detected since the last note, the transition is classified as *staccato*, else as *legato*. This simple rule performed with an accuracy of 79.121%, combining sax and violin instances in the test set.

Comparing the overall results of the user-dependent supervised training, we can conclude that there is no significant improvement over the user-independent classifier based on heuristic rules.

4 SYNTHESIS

With the output of the modules described in sections 2 and 3, the system generates corresponding transcriptions, which feed the sound synthesizer. We re-use the ideas of the concatenative sample-based saxophone synthesizer described in (Maestre et al., 2006). Transcription includes *note duration*, *note MIDI-equivalent pitch*, *note dynamics*, and *note-to-note articulation type*. Sound samples are retrieved from the database taking into account similarity and the transformations that need to be applied, by computing a distance measure we describe below. Se-

Table 4: Results of the supervised training with 3 classes(staccato, normal and legato) and 2 classes (staccato and legato) using ten-fold cross-validation. MFCC (first five coefficients) are taken as input attributes. Results of a classifier based on heuristic rules with 2 classes(staccato and legato).

SUPERVISED TRAINING: 3 CLASSES baseline = 33%	
description	*correct* (%)
subject1- sax	57.727
subject1- violin	44.5455
subject1- sax-violin	51.5909
subject2- sax	67.281
subject2- violin	67.2811
subject2- sax-violin	51.2415
subject3- sax	41.7391
subject3- violin	48.7365
subject3- sax-violin	40.2367
subject4- sax	41.7722
subject4- violin	42.916
subject4- sax-violin	38.3648
SUPERVISED TRAINING: 2 CLASSES baseline = 66%	
description	*correct* (%)
subject1- sax	83.1818
subject1- violin	71.3636
subject1- sax-violin	78.6364
subject2- sax	93.5484
subject2- violin	67.699
subject2- sax-violin	80.5869
subject3- sax	70.4348
subject3- violin	72.2022
subject3- sax-violin	69.0335
subject4- sax	64.557
subject4- violin	73.3333
subject4- sax-violin	66.6667
HEURISTIC RULES: 2 CLASSES baseline = 66%	
description	*correct* (%)
subject1- sax-violin	82.2727
subject2- sax-violin	79.684
subject3- sax-violin	76.3314
subject4- sax-violin	78.1971

Table 5: Mean percentage for all subjects of correctly classified instances using: *1)*MFCC (central frame); *2)*MFCC+LR (added left and right frames of the transition); *3)*MFCC+LR+DLDR (added difference from left to central, and right to central frames); *4)* Heuristic rules.

attributes	*sax*	*violin*	*sax+violin*
1	77.930	71.698	73.730
2	80.735	72.145	74.747
3	81.067	72.432	75.742
4	–	–	79.121

lected samples are first transformed in the frequency-domain to fit the transcribed note characteristics, and concatenated by applying some timbre interpolation around resulting note transitions.

4.1 Synthesis Database

We have used an audio sample database consisting of a set of musical phrases played at different tempi, played by a professional musicians. Notes are tagged with several descriptors (e.g. MIDI-equivalent pitch, etc.), among which we include a legato descriptor for consecutive notes, that serves as an important parameter when searching samples (Maestre et al., 2006). For the legato descriptor computation, as described in (Maestre and Gómez, 2005), we consider a *transition* segment starting at the begin time of the release segment of the first note and finishing at the end time of the attack of the following one, computing the *legato* descriptor LEG (Eq. 3)by joining start and end points on the energy envelope contour (see Figure 4) by means of a line L_t that would ideally represent the smoothest case of detachment. Then, we compute both the area A_2 below energy envelope and the area A_1 between energy envelope and the joining line L_t to define our legato descriptor.

The system performs sample retrieval by means of computing a euclidean feature-weighted distance function. An initial feature set consisting on MIDI pitch, duration, and average energy (as a measure of dynamics), is used to compute the distance vector. Then, some features will be added depending on the context. For note-to-note transitions, two features (corresponding to the left and right side transitions) are added: legato descriptor and pitch interval respect to the neighbor note.

$$LEG_1 = \frac{A_1}{A_1 + A_2} = \frac{\int_{t_{init} \le t \le t_{end}} (L_t(t) - E_{XX}(t))dt}{\int_{t_{init} \le t \le t_{end}} L_t(t)dt} \quad (3)$$

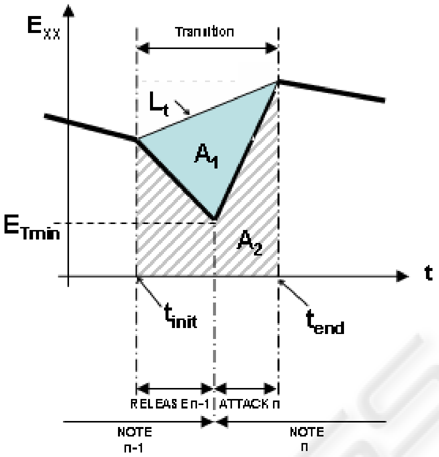

Figure 4: Schematic view of the **legato** parameter extraction

4.2 Sample Transformation and Concatenation

The system uses spectral processing techniques (Amatriain et al., 2002) for transforming each retrieved note sample in terms of amplitude, pitch and duration to match, in the same terms, the target description. After that, samples are concatenated following the note sequence given at the output of the performance model. Note global energy is applied first as a global amplitude transformation to the sample. Then, pitch transformation is applied by shifting harmonic regions of the spectrum while keeping the original spectral shape. After that, time stretch is applied within the limits of the sustain segment by repeating or dropping frames.

5 CONCLUSION

The presented work is a proof-of-concept toward user-adapted singing-driven interfaces. A novel segmentation method is introduced, which benefits from the phonetic characteristics of vocal instrument imitation signals. Referring to the articulation type, reported results of the classifier of supervised training that adapts to user behaviour, are comparable to using a user-independent classifier based on heuristic rules. In the final implementation, the mappings of articulation type to the synthesizer derive from the latter classifier. The results of this first experiment, enlightened us aspects about phonetics and instrument imitation that should be further investigated. For instance, we could use the introduced syllabling segmentation module to define, for each class of Table 1, a subset of phonemes employed by a given user.

ACKNOWLEDGEMENTS

This research has been partially funded by the IST project SALERO, FP6-027122. We would like to thank all participants in the recordings.

REFERENCES

Amatriain, X., Bonada, J., Loscos, A., and Serra, X. (2002). *DAFX - Digital Audio Effects*, chapter Spectral Processing, pages 373–438. U. Zoelzer ed., J. Wiley & Sons.

Bonada, J. and Serra, X. (2007). Synthesis of the singing voice by performance sampling and spectral models. *IEEE Signal Processing Magazine*, 24(2):67–79.

Egozy, E. B. (1995). Deriving musical control features from a real-time timbre analysis of the clarinet. Master's thesis, Massachusetts Institut of Technology.

Janer, J. (2005). Voice-controlled plucked bass guitar through two synthesis techniques. In *Int. Conf. on New Interfaces for Musical Expression, Vancouver*, pages 132–134, Vancouver, Canada.

Lesaffre, M., Tanghe, K., Martens, G., Moelants, D., Leman, M., Baets, B. D., Meyer, H. D., and Martens, J. (2003). The mami query-by-voice experiment: Collecting and annotating vocal queries for music information retrieval. In *Proceedings of the ISMIR 2003, 4th International Conference on Music Information Retrieval, Baltimore*.

Lieberman, P. and Blumstein, S. E. (1986). *Speech physiology, speech perception, and acoustic phonetics*. Cambridge University Press.

Lindemann, E. (2007). Music synthesis with reconstructive phrase modeling. *IEEE Signal Processing Magazine*, 24(2):80–91.

Maestre, E. and Gómez, E. (2005). Automatic characterization of dynamics and articulation of monophonic expressive recordings. *Procedings of the 118th AES Convention*.

Maestre, E., Hazan, A., Ramirez, R., and Perez, A. (2006). Using concatenative synthesis for expressive performance in jazz saxophone. In *Proceedings of International Computer Music Conference 2006*, New Orleans.

Sundberg, J. (1994). Musical significance of musicians' syllable choice in improvised nonsense text singing: A preliminary study. *Phonetica*, 54:132–145.

Wanderley, M. and Depalle, P. (1999). *Interfaces homme - machine et création musicale*, chapter Contrôle Gestuel de la Synthèse Sonore, pages 145–63. H. Vinet and F. Delalande, Paris: Hermès Science Publishing.

Widmer, G. and Goebl, W. (2004). Computational models of expressive music performance: The state of the art. 3(33):203–216.