SALTO
A Spectral Domain Saxophone Synthesizer

Joachim Haas

Music Technology Group, Pompeu Fabra University
joachim.haas@iua.upf.es, http://www.iua.upf.es/mtg

Abstract

In the context of the MOSART Program, a software-based realtime saxophone synthesizer is being developed at the Music Technology Group of the IUA. Main objectives are high sound quality and flexible control over meaningful musical performance parameters. The synthesis concept is built upon the "Sinusoidal plus Residual Model". A database holds spectrally pre-analyzed sound segments, which are used as anchor-points in a continuous multi dimensional timbre space. Each spectral segment consists of a stationary sinusoidal and a non-stationary residual part. By applying spectral interpolation techniques and modeling specific spectral properties, a realistic timbre space can be synthesized. With this method it is possible to preserve the "sample-like" quality when needed by simply adding both parts together without applying any modification to them. By treating the parts individually the whole range and flexibility of spectral transformation and interpolation techniques becomes available for the synthesis process.

1. Introduction

Spectral based analysis/synthesis techniques offer a wide range of controls over sound processing and synthesizing. In contrary to other synthesis techniques (e.g. Physical Modeling) a remarkable number of musically meaningful parameters are closely related to specific spectral processing algorithms. Currently there are various approaches to the spectral analysis/synthesis [1][2]. This work has concentrated on the "Sinusoidal plus Residual Model" by X. Serra [3]. In this approach, harmonic partials are detected by examining the time varying characteristics of a sound and representing them as slowly time-varying sinusoids. The sinusoidal part $s(t)$ of the input signal $x(t)$ can therefore be modeled as

$$s(t) = \sum_{r=1}^{R} A_r(t)\cos[\Theta_r(t)] \qquad (1)$$

where $A_r(t)$ being the instantaneous amplitude and $\Theta_r(t)$ the instantaneous phase of the rth partial.

By subtracting the sinusoidal part $s(t)$ from the original input sound we obtain the residual part :

$$e(t) = x(t) - s(t) \qquad (2)$$

The residual part consists ideally of all the non-stationary components in the input signal. It can be approximated with filtered noise or other noise models to reduce the amount of spectral data [4]. The sinusoidal and residual data retrieved in the analysis process form the base to model and synthesize the target instrument in spectral domain.

2. Spectral Modeling

"Spectral Modeling" can be understood in several ways. The approach we concentrated on, is to use spectrally pre-analyzed instrument samples and combine them where necessary with abstract models or spectral interpolation. This approach combines the advantages of the well known and widely used sampling techniques with the specific properties of the Spectral Modeling Techniques. Whereas the "Spectral Samples" provide the most natural sound quality possible, spectral processing techniques allow a much wider range of sound transformations, timbre interpolation and transition modeling. The more the characteristics of an instrument are studied and understood in the evolution of a project, the more abstract models may replace the spectral samples and the interpolation techniques.

This document reflects the current state of the ongoing SALTO project. It is not intended to be a final report, it should be interpreted as an intermediate status report presenting first results.

3. SMS Analysis

Following the "Sinusoidal Plus Residual Model" a set of saxophone samples, recorded ideally in a non-reverberant environment, was analyzed and separated in sinusoidal and residual components. The resulting database currently holds information about 72 spectrally analyzed sounds and covers an ambitus of 2

octaves with 24 pitches in semitone distance, each with 3 different attack characteristics.
The spectral analysis was performed with the "SMSTools2" Software Package [4] using the following simplified processing scheme:

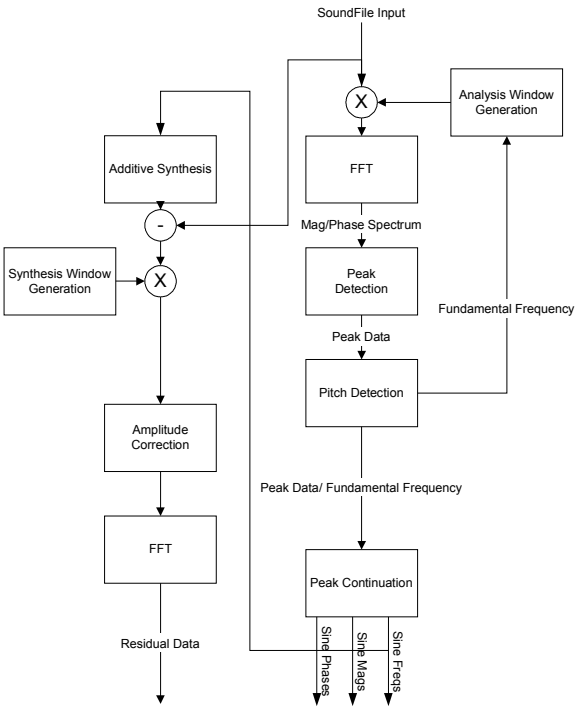

Figure1: SMS Analysis Process

The resulting information can be classified in two groups:
a.) General Spectral Properties
This group reflects general information about the instrument's spectral behavior related to e.g. note transitions (legato), vibrato, note-release, timbre creation or specific playing styles. It has crucial importance for deriving abstract mathematical models.
b.) SMS Spectral Analysis Data
The SMS Spectral Analysis Data includes the complete frequency domain representation of an analyzed sound according to the "Sinusoidal plus Residual Model". It is directly used in the synthesis process and therefore exported to SALTO.
The sinusoidal components are represented by "Sinusoidal Tracks" describing frequency, magnitude and phase evolution of each partial. The tracks correspond directly to the sound's partials, as we only work with harmonic sounds all having a clearly defined fundamental frequency.
The residual part ideally contains no stationary components and is stored in a frame-based stream. In practice, some undetected harmonic components always remain in the residual and may cause some audible artifacts in the synthesis process, especially if the corresponding sinusoidal part is e.g. pitch-shifted.
Additionally, the Fundamental Frequency is calculated for every analysis frame and stored together with the other frame-based data.
The SMS Spectral Analysis Data is stored for each analyzed audio sample in a generic SMS File Format (.sms). To import the data in SALTO, a conversion into the more universal SDIF Format (Sound Description Interchange Format) [6] is done via the SMS-Commandline Interface[4]. The SDIF Format is organized as a sequence of time-tagged frames, representing one or more data "streams".

4. The SALTO Synthesis

The SALTO Synthesizer is built on a set of platform independent synthesis classes written in C++. In the first implementation stage MIDI Input Data can be received either from a keyboard or a Yamaha WX5 Breath-Controller. The incoming MIDI is mapped to a Timbre Vector containing information about attack characteristics, velocity and timbre of a note event. With this information one of the "Spectral Segments" is selected and accessed in the SDIF Database. According to the analysis frame rate of 172 frames/s, each 0.58 ms a "Synth-Frame" - container is filled with spectral data in the Data Management Section and passed through the various DSP stages. It holds the current frame-related sinusoidal track information, the residual spectrum and synthesis parameters.
To control the timbre of the initial Spectral Segment, its sinusoidal part can be morphed with predefined spectral templates. A Timbre Template Database holds therefore spectral templates with characteristic "piano" or "forte" spectra.
In the spectral transformation stage a pitch shift algorithm is implemented. The pitch shift value is controllable in real time via MIDI (In Yamaha-WX5-mode the player's lip pressure is mapped to this pitch shift value).
Additionally an other pitch correction algorithm can be activated to either adjust the spectral samples which are not exactly in tune or to model a certain pitch behavior of the Saxophone (e.g. increase pitch at the end of the 2nd octave or simulate tuning characteristics of specific tones).
All pre-calculable operations are done at analysis time or while starting up the software in order to optimize realtime processing speed. In the current implementation all data is entirely loaded into RAM before starting the audio synthesis process. This provides easy and fast access to all spectral data. In a later stage with a growing database it's recommendable to only have short parts of the spectral segments permanently in memory. The remaining part of a specific segment should be loaded on demand while its first part is already being played back.

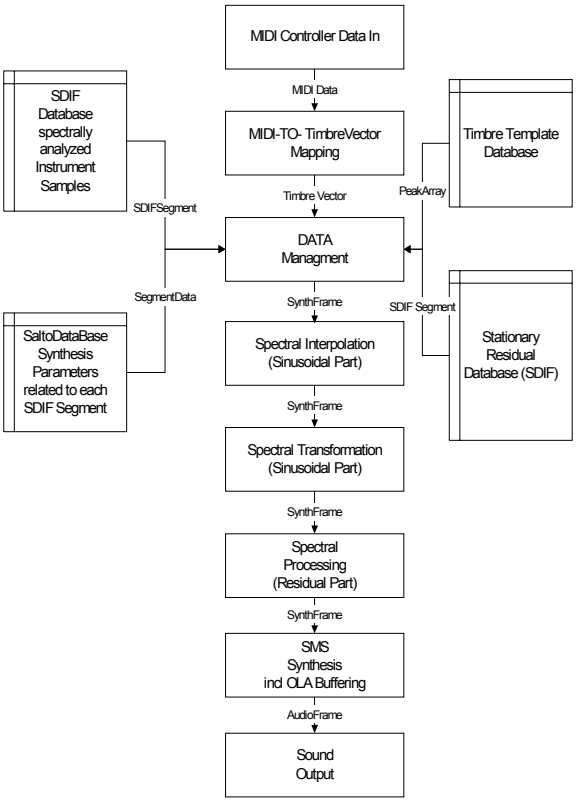

Figure 2: SALTO Synthesis Flow

5. Midi Mapping

The incoming MIDI Information of SALTO consists basically of two different parameter groups:

a.) initial controls

- initial pitch (fingering)
- initial attack velocity
- note on/note off
- external controller data

b.) continuous controls

- breath speed
- lip pressure
- external controller data

Depending on the gesture controller (Keyboard or Breath Controller) the MIDI input data has to be mapped differently to the synthesis control parameters. Following the proposal in [7] MIDI-Mapping Strategies could be classified in three different types:

- One-to-One Mapping (1 input controller to one synthesis parameter)
- Divergent Mapping (1 input controller to various synthesis parameters)
- Convergent Mapping (various input controllers to 1 synthesis parameter)

We decided to either move towards divergent or convergent mapping strategies, depending on the number of available gesture controls.

As a keyboard controller usually offers only a small number of gestural outputs, divergent mapping has to be applied here. Unfortunately, this leads clearly to a restriction in expressivity control of the synthesized instrument. To reduce this lack of control, we use the divergent mapping only for the initialization of a note event. While the note is being played, the player still has relative access to some of the synthesis parameters which were already set in the initialization step. By providing this additional relative controls via an external MIDI Controller Desk, the player is able to perform crescendo, decrescendo or relative timbre changes and control the legato transition handling. The current midi mapping for a keyboard + external controller is:

Initial MIDI Controller	*Synthesis Parameter*
Pitch (fingering)	Pitch
Attack Velocity	Attack character Synthesis Volume Timbre Interpolation
Note-On/ Note-Off	Note-On/Off Transition Recognition

Table 1: Initial Midi Mapping (Keyboard)

Relative Controller	*Synthesis Parameter*
Pitch Modulation Wheel	Relative Pitch
Modulation Wheel	Relative Volume
Ext. Controller 1	Transition Time
Ext. Controller 2	Relative Timbre

Table 2: Additional MIDI Controllers (Keyboard)

As opposite to common controllers like MIDI-keyboards, a Breath Controller is able to send continuous MIDI Control Data. In the case of the Yamaha WX5, two independent information streams are transmitted continuously: air speed and lip pressure. Additionally, the WX5 can be set up to send an initial velocity value with each Note-On event, depending on the performed attack type (the higher the value, the sharper or shorter the attack has been performed). This information is directly used to select Spectral Segments with the according properties. Currently the incoming MIDI Data is mapped in the following way :

Initial MIDI Controller	*Synthesis Parameter*
Pitch (fingering)	Pitch
Attack Velocity	Attack character Synthesis Volume
Note-On	Note-On

Table 3: Initial MIDI Controllers (Breath Controller)

Continuous Controller	*Synthesis Parameter*
Lip Pressure	Pitch Modulation
Breath Speed	Volume Timbre Interpolation (Note Off) (Transition Recognition)

Table 4: Relative Midi Mapping (Breath Controller)

6. Single Note Synthesis

The synthesis process of a single note consists classically of three regions: The attack stage, the stationary stage and the release stage. Due to the "Sinusoidal Plus Residual" approach each of this stages is build out of its sinusoidal and residual part. Note that each SDIF Sample contains both, original sinusoidal and original residual part of a saxophone note, even if they are treated differently in the synthesis process. The analyzed samples are not cut directly after their attacks, we keep at least 0.5 s of the stationary-state sound for random mirror looping purposes.

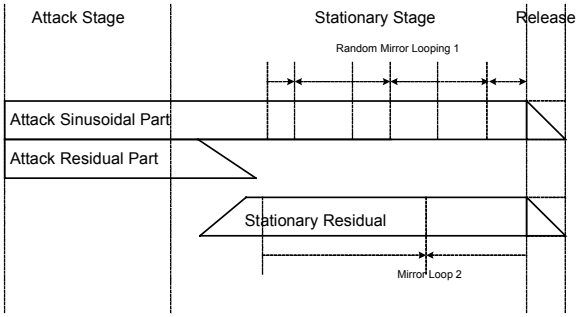

Figure 3: Single Note Synthesis

6.1 The Attack Stage

In the attack stage of a note, the sinusoidal and residual parts are recombined and form therefore the original "natural" sample. It must be noted that adding the both unmodified parts together leads to a perfect reconstruction of the original signal [3]. This offers a "sample-like" sound quality in the most sensitive part of the synthesis. At the beginning of the sounds (the first few tens of milliseconds) the harmonicity condition, which is fundamental for the SMS approach, is not fulfilled. In consequence the sinusoidal/residual splitting can't be applied successfully and we are obliged to treat both parts together without any modification. Even a minor transformation or modification of the original signal could become audible and may sound quite unnatural. Even small changes in phase relation or in the relation of the sinusoidal and the residual part would change severely the original sound character.

In an earlier implementation stage, we tried to combine the sinusoidal part of one "Spectral Sample" with the various residual parts of different attack types. This approach didn't sound as well as expected. Because the condition of slowly varying sinusoids is not fulfilled in the early attack stage, the combination of independent components didn't lead to successful results.

In a future implementation, a time-expansion/ compression algorithm will be implemented, to provide higher flexibility in changing the attack characteristics.

6.2 The Stationary Stage

After some tens of milliseconds the spectral sample shows already stationary behavior. In our current implementation, the original residual is cross-faded to an analyzed recording of a breath-only sound. "Breath-only" has the meaning of only blowing in the instrument without generating a complete tone. With this trick we avoid most of tonal components due to SMS-Analysis-Errors and gain more flexibility in recombining the transformed sinusoidal part with the residual. This Pseudo-"Stationary Residual" has a loop length of about 4 s to avoid repetition patterns and is currently used throughout all 24 different pitches.

The sinusoidal synthesis enters in a random mirror loop to stretch the latter part of the spectral segment as long as the note is supposed to sound. Each sample has individual predefined loop limits which can be edited by the user. Once entered the loop mode, the frames within the loop limits are played alternating forwards and backwards with a changing random loop-length.

6.3 The Release Stage

In the current implementation the release stage is only in a very provisory state and not studied in detail yet. It is realized by a fade-out within 3 frames to avoid clicks at the sound end. A saxophone player normally stops a note by stopping the reed with his tongue. This action is very fast and similar to a quick fade out.

6.4 Timbre Interpolation

One of the well known characteristics of a wind instrument is the timbre change depending on the "performed" volume. Its quite obvious that if the instrument is played louder, the spectrum gets "brighter". The amplitudes of higher harmonics become more present in a spectrum of a louder tone. In [8] is stated that the spectral evolution of a crescendo or decrescendo trumpet tone is independent of the crescendo speed and its "direction". For higher harmonics (>=10th) with lower amplitudes other unknown factors or a certain random deviation becomes noticeable. Comparisons of various saxophone spectra showed that the same assumptions can be made for the SALTO Synthesizer.

To realize the volume-dependent timbre changes, we use a "Timbre Template Database" which holds templates of "piano"- and "forte"-spectra for different pitches. Each template spectrum holds information about which partials being present and about the related magnitudes. Depending on the performed volume, the current "Spectral Sample" can be

morphed continuously either towards a piano or a forte spectrum.
As stated before it is not suitable to apply transformations already on the very first attack frames. Therefore the user has to predefine a starting point for each spectral sample, from which on the interpolation algorithm is activated. More information about spectral morphing can be found in [9][10]. For the time being we only interpolate the spectral envelope of the stationary sinusoidal.

7. Note Transitions

One of the key features of a spectral-domain-based synthesizer in comparison with any "Giga-Sampler"-like instrument is the modeling of note transitions. Common samplers can't model a legato transition between two succeeding notes. Our preliminary results have to be proven more generally and refined in detail but they build a very good starting point.

7.1 Downward Transition

Figure 4 shows an example of a slow legato transition of one semitone downwards. The first curve reflects the behavior of the fundamental frequency, the middle curve the related overall amplitude of the sinusoidal part and the lower curve the overall residual amplitude over time (all semi-logarithmic).

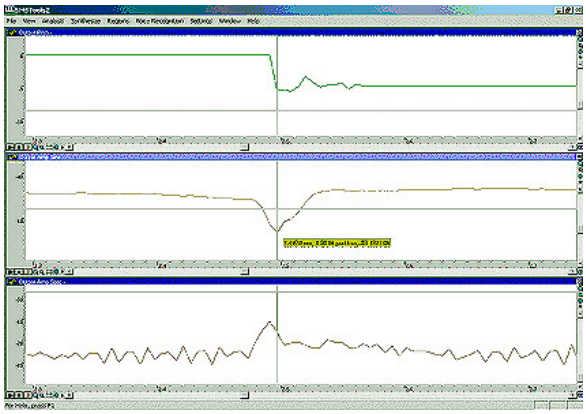

Figure 4: Legato; C# - C; 276Hz - 261Hz

Fundamental Frequency :
The frequency transition seems to be quite linear and the end point coincides in time with the point of the lowest Sinusoidal Amplitude. The frequency-transition-time t1 was always around 0.01s, no matter if the legato was performed fast or slow. With a frame-time t_{Frame}= 1s/172,27 = 5,8ms the frequency transition has a duration of about 2 or 3 frames.
Overall Amplitude Sinusoidal Part :
Linear amplitude fade with a2 being around 6 dB. The fade time t2 depends on whether the legato was performed slow or fast (between 0.02 s and 0.08 s).
Overall Amplitude Residual Part :
The peak in the overall amplitude curve of the residual coincides always with the beginning of the frequency transition. By listening only to the residual sound at this position, it could be identified as the pad beating on the corpus of the instrument (characteristic "plop" sound). In the current implementation, this problem is not considered yet. Possible solutions could be a transition database or a temporal increase and decrease of the stationary residual level. Due to the nonlinear behavior of the original sound at this position, the first solution seems to be much more successful.
Most of the examined downwards legatos (8 of 10) showed more or less the same behavior leading to the following first model :

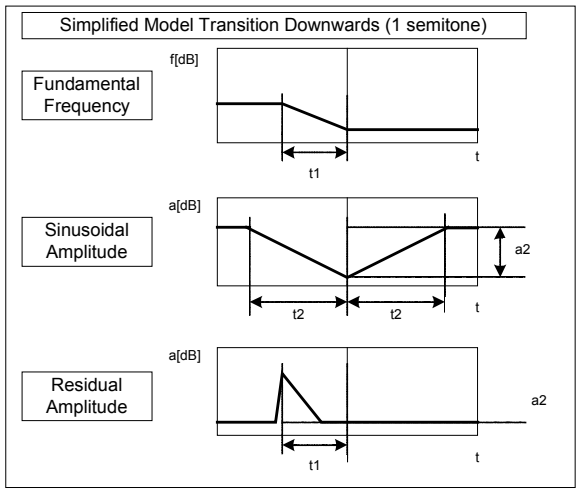

Figure 5: Transition Model Downwards

7.2 Upward Transition

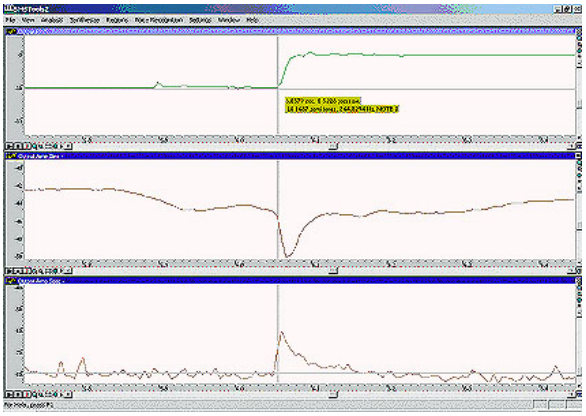

Figure 6: Legato Upwards; B-C; 246Hz -260Hz

Fundamental Frequency :
The transition time t1+t1 in the upward transitions is around 0.02s, therefore around 3 frames. The centre of the transition coincides with the lowest point of the sinusoidal part.
Overall Amplitude Sinusoidal Part :
The sinusoidal amplitude is faded like in the downward transition. The fade-out time t2 depends on

the performed legato, the amount of the fade out is around a2 =6dB.

Overall Amplitude Residual Part :

The peak of the residual amplitude coincides with the beginning of the frequency transition. Like in the downward transition the residual amplitude is not considered yet.

The behavior of the examined upwards transitions leads to the following model:

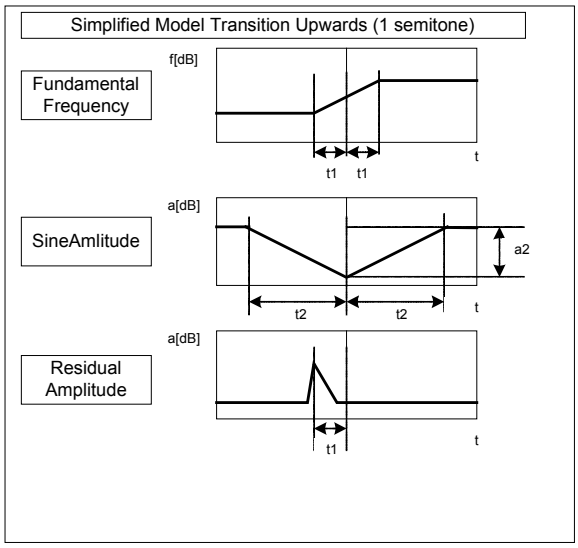

Figure 7: Transition Model Upwards

8. Conclusions

The first implementation of the SALTO Synthesizer looks quite promising. The combination of almost sample-like sound quality with extended expressive controls could open new perspectives in electronic instrument synthesis. Even if sampling devices nowadays become more and more sophisticated due to the increasing memory resources, processing power and improved sound conversion [11], the lack of musical expressive control is obvious. On the other hand, PM Techniques could provide a much higher degree of expressiveness but, with a few exceptions, the results are often far from sounding natural in comparison with sampled sounds.

Next steps will therefore be the improvement of note transition models and the implementation of a time stretching algorithm to gain more flexibility respectively the attack behavior. The synthesis approach will extended to other instruments like e.g. trumpet or flute.

Another important improvement to work on is clearly the gesture control and the related MIDI Mapping.

Unclear is, at the time being, how the model could be extended to extreme and in-harmonic sounds of an instrument to cover the whole range of an instrument's timbre space. The question of how to implement special playing styles like e.g. growling, multiphonics is neither adressed in the common implementation .

9. Acknowledgments

This research is supported by the European MOSART (Music Orchestration Systems in Algorithmic Research and Technology) project HPRN-CT-2000-00115. The author would like to thank X. Serra and all members of the Music Technology Group for their great and helpful support.

10. References

[1] Macon W., "Applications of Sinusoidal Modeling to Speech and Audio Processing", Ph.D. Dissertation, Georgia Institute of Technology, 1993

[2] Serra X., Bonada J., et al "Integrating Complementary Spectral Models in the Design of a Musical Synthesizer", Proceedings of the ICMC 97, Thessaloniki, Greece, pp152-158, 1997.

[3] Serra X., "A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition", Ph.D. Dissertation, Stanford 1989

[4] Serra X., "Musical Sound Modeling with Sinusoids plus Noise", in G.D. Pioli, A. Picalli, S.T. Pope and C. Roads, editors, Musical Signal Processing, Swets and Zeitlinger Publishers, Lisse 1997

[5] Wright M., Chaudhary A., Freed A., Wessel D., "New Applications of the Sound Description Interchange Format", Proceedings of the ICMC 1998

[6] Wright M., "SDIF Specification" http://cnmat.CNMAT.Berkley.EDU/SDIF/Spec.html

[7] Rovan J.B.,Wanderley M.M.,Dubnov S., Depalle P., "Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance", KANSEI –The Technology of Emotion, AIMI International Workshop, Genova

[8] Dannenberg R., Derenyi I., "Synthesizing Trumpet Performances", Proceedings of the ICMC, SanFrancisco, 1998

[9] Slaney M., Covell B., Lassiter B., "Automatic Audio Morphing", International Conference on Acoustics, Speech and Signal Processing, Atlanta,1996

[10] Polansky L., Erbe T. "Spectral Mutation in Soundhack", Soundhack Manual, Lebanon, Frog Peak Music, 1994

[11] Hind, Nicky "Extensions to Sample Playback Technology: Multiple Loop Points and Alternative Articulation", CCRMA Stanford, 1994