REAL-TIME SPECTRAL SYNTHESIS FOR WIND INSTRUMENTS

Oscar Mayor, Alex Loscos, Jordi Bonada

Music Technology Group, Pompeu Fabra University
{omayor,aloscos,jbonada}@iua.upf.es , http://www.iua.upf.es/mtg

Abstract

A real-time spectral domain monophonic synthesizer is being developed. The idea is to achieve the sound quality given by samplers and improve the flexibility of control and model transitions between notes. The output sound is generated concatenating, transforming and synthesizing attacks and stationeries previously recorded by a real player that have been analyzed and stored in a database. A spectral technique called SPP is used in order to get better sound quality than other spectral techniques, like the sinusoidal or sinusoidal plus residual model, when transforming the sound. The aim is to get a sampler-like quality but with a high degree of flexibility to transform the sound and to model note transitions.

1 Introduction

In this article we describe the progress report of a current project and the future work that is being done. The project consists in a real-time monophonic spectral synthesizer that has been applied successfully to brass and reed instruments like the trumpet and the saxophone.

The synthesis process has been changed [1] from a sinusoidal plus residual model [2] to a pure spectral model, where the spectrum is divided in a set of regions, each region representing a harmonic spectral peak and its boundaries and we apply timbre and pitch transformations preserving the regions.

The synthesis is controlled with a set of parameters that can be changed via midi or with a complete graphical user interface.

There is a database with pre-analyzed samples that will be selected accordingly in the synthesis stage depending on the midi events that came in.

In figure 1 we can see a flow diagram of the synthesis process that is explained later in this article.

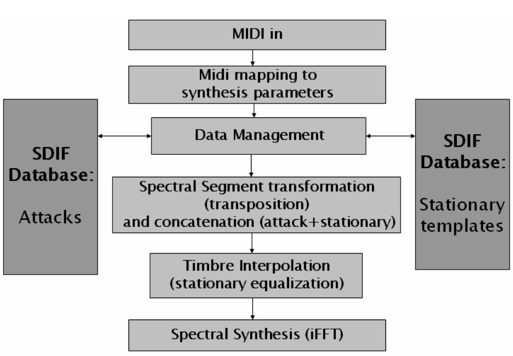

Figure 1: Flow diagram of the synthesis process.

2 Spectral Peak Processing

Spectral Peak Processing is a technique based on considering the spectrum as a set of regions, each of which belongs to one spectral peak and its surroundings [3]. The goal of such technique is to preserve the convolution of the analysis window after transposition and equalization transformations. The local behaviour of the peak region should be preserved both in amplitude and phase after transformations. To do so, the delta amplitude relative to the peak's amplitude and the delta phase relative to the peak's phase are kept unchanged after spectral transformations. The region boundary is set to be at the lowest local minimum spectral amplitude between two consecutive peaks or if there are no local minimums, at the middle frequency between two consecutive peaks.

In figure 2 we can see an example of a segmentation of one STFT spectrum. We can distinguish the little squares representing the harmonic peaks and the vertical lines representing the boundaries of each region.

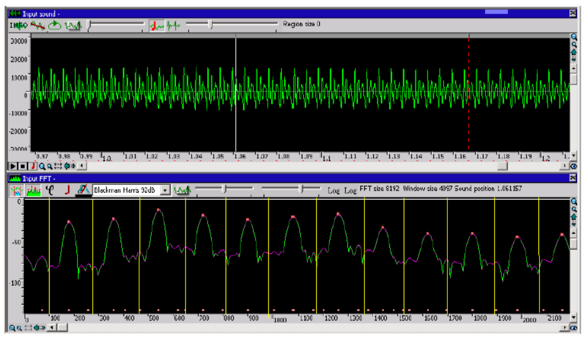

figure 2: SPP analysis region segmentation.

We can apply basically two kinds of transformations over the spectrum: equalization and transposition.

2.1 Equalization

Equalization means a timbre change. From the SPP analysis we get a spectrum with the harmonic peaks (square points in the figure) and the regions. Then we want to change the spectrum to follow a desired envelope (transversal line in the figure) that will change the timbre to be adjusted to the target one.

The SPP equalization is done by calculating for each region the amount needed so that each harmonic's amplitude matches the desired envelope (in the figure we can see the displacement marked with arrows). Then this amount is added to all the bins that belong to a region (amplitude linear addition). Therefore, the spectrum amplitude of a region will be just shifted up or down and it will keep its local behaviour. The phase is unchanged.

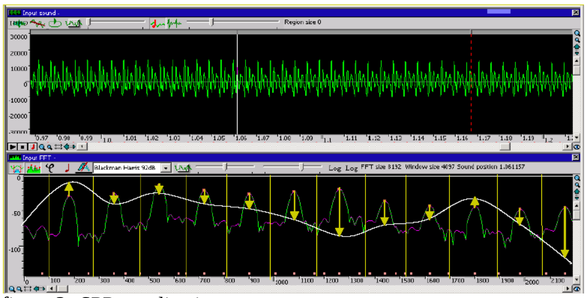

figure 3: SPP equalization

2.2 Transposition

Transposition means to change the pitch of a spectral frame by multiplying the harmonic's frequencies by a constant value. In SPP, this operation can be done by shifting SPP regions in frequency.

In the figure 4 we can see graphically this process for transposing up. The arrows show the displacement in frequency that is applied to each harmonic. At the bottom we can see the resulting spectrum. As we can see, the amount of frequency shifting calculated for each harmonic peak is applied as a constant to its whole region, so the linear frequency displacement for all the bins in a region will be the same. Therefore, the local amplitude spectrum of each region will be kept as it is, thus preserving the window convolution with each harmonic peak.

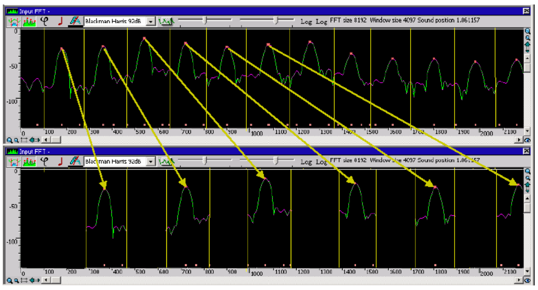

figure 4: SPP transposition

The frequency displacement will be a non integer value in most cases. Therefore the spectrum should be interpolated. A spline interpolation of 3rd order to deal with that is a good compromise of quality versus computational cost.

In the previous figure we can see an example of transposition to a higher pitch. The SPP regions are separated in the resulting spectrum with holes that are filled up with spectral amplitudes of constant amplitude -200dB.

In the other hand, when we are transposing to a lower pitch, the SPP regions will overlap in the resulting spectrum. This overlapping is achieved by adding the complex values at each spectral bin.

3 Database

As we are not creating new sounds, but imitating a real instrument like the trumpet or the saxophone, we have a complete recording of these instruments performed by a professional player. These recordings are analyzed using the SPP technique and stored in a database that will be used by the synthesis process to synthesize the output sound.

For each instrument we want to synthesize, we will have in the database 4 different attacks recorded for each pitch from very soft to very hard. The attack corresponds to only the first tens of milliseconds of a note, usually less than 0.1 seconds.

The database also contains some stationary templates (two or three pitches for each octave) also with up to four different dynamics from very low/soft to very hard/aggressive. These stationary templates will be at least 1 or 2 seconds long and have to be recorded without any kind of expression like vibrato or reverb. As we don't have stationeries for all the notes in the scale, we will have to transpose the same stationary to achieve some different pitches, we will always use a lower pitch stationary than the desired pitch to synthesize and will transpose it up, to get better results and avoid the overlap between regions when we transpose down.

Once these sounds are analyzed, the output of the analysis is stored in the database in an SDIF file [4] that will be loaded in memory at the beginning of the sy nthesis process.

4 Real-Time Synthesis and Processing

The synthesizer is controlled in real-time by midi, the idea is that the processing core receives the midi signals and chooses the segments accordingly to the processing algorithm and concatenates them to produce a natural sound.

4.1 Midi Control and Mapping

The current implementation of the synthesizer can be controlled with a complete GUI but also via MIDI. An input midi file will produce an output wave file with the melody synthesized, but the most interesting way of control is the implementation of real-time midi control with a simple midi keyboard or with the midi breath controller Yamaha WX5.

There are two kinds of incoming MIDI information, the initial controls and the continuous controls. In the next tables we can see this controls and who changes them in the case of the midi keyboard and the breath controller:

	MIDI keyboard	Yamaha WX5
Initial pitch	Key pressed	Fingering position
Attack velocity	Key velocity	Initial breath speed
Note On/off	Press/Release key	Fingering changes

table 1: initial controls

	MIDI keyboard	Yamaha WX5
Note volume	After-touch value	Continuous breath speed
Pitch modulation	Pitch bend wheel	Lip pressure

table 2: continuous controls

These controls are mapped to parameters of the synthesizer, following a divergent mapping that means that each midi control can be mapped to more than one synthesis parameter [5], in the next table we can see which parameters are controlled by which control.

MIDI controls	Synthesis Parameter
Initial pitch	Pitch (note)
Attack velocity	Attack type Synthesis volume
Note on/off	Note on/off Transition recognition
Note volume	Volume Stationary timbre interpolation Note off
Pitch modulation	Relative pitch modulation

table 3: mapping controls/synthesis parameters

4.2 Attack Stage

The midi attack velocity value that arrives with a note on will decide which kind of attack will be selected from the database to synthesize the output sound; the higher the attack velocity the harder the attack will be selected. The attack is just synthesized by doing the iFFT of the database spectral segment chosen because we have attacks for all different pitches. As we are not modifying any characteristics of the original attack, we are not equalizing or transposing the spectrum, the output result after synthesizing will have "sampler-like" sound quality.

When the attack ends, a stationary template segment will be concatenated, accordingly to the note volume MIDI control at the end of the attack.

4.3 Stationary Stage and Interpolation

The stationary part is selected depending on the continuous note volume control at the end of the attack region, the attack and the stationary segment chosen will be concatenated.

During the stationary part, timbre interpolation between stationeries with other dynamics will be performed when the player reduces or increases the volume [6]. It is not enough to decrease or increase the amplitude of the spectrum to achieve a natural sound quality because in wind instruments like the trumpet or the saxophone, when the dynamic changes, the timbre also changes, so the spectrum's envelope has to be equalized from the base spectrum, which is the stationary segment being synthesized, to the target spectrum which is the spectrum of the template with a higher or lower dynamic in case that we are increasing or decreasing the volume.

This equalization has to be applied gradually, assigning weights to the base and target spectrum envelope depending on the continuous note volume, so we obtain a continuous timbre space. The idea is that we will have 4 quantized points in the timbre space, each one assigned to a note volume level, so when the note volume is near one of this points the target spectrum is changed to the desired template and the nearer the value is from the quantized point, the higher weight is given to the target template between 0 and 1 and the lower weight is given to the base template also between 0 and 1, then the average envelope is calculated following a weighted linear interpolation between base and target. The base spectrum which is always given by the template chosen after the previous attack will be equalized to the average envelope calculated.

In case of synthesizing stationeries that are longer that the recorded ones we will loop in between the same stationary until the note ends or a note transition arrives.

4.4 Release Stage

The release stage is realized with a simple 3 frames fade-out at the end of the stationary just to avoid clicks in the output sound, the effect is quite similar to the sound produced when a saxophone player stops the reed with the tongue.

5 Transition modelling

One of the most important features that makes a spectral synthesizer more attractive than a sampler or other time domain synthesis methods based on wave tables [7] is the possibility to model note transitions.

A transition is detected when the incoming MIDI produces a new note on before the note off of the previous note on. This is realized always in the breath controller Yamaha WX5, because when you change the fingering position, if the breath pressure is bigger than zero, the controller sends the new note on for the new fingering and immediately the note off of the previous fingering. In case of the midi keyboard we can simulate a note transition, pressing the new key before releasing the previous one.

To model these note transitions, from real recordings we extract the pitch and amplitude envelope during the transition and then build a model that will be applied in the sy nthesis process when concatenating two stationeries at different pitches.

In the next figures we can see two examples of note transitions up (figure 5) and down (figure 6), the pitch and amplitude evolution is quite similar in both cases. From these real recordings analyzed we can build a model where the pitch and amplitude variation is realized during five frames (~30ms) like in most of the examples and the lowest point of the amplitude variation coincides with the middle of the pitch transition. The amplitude variation is for most of the cases around a decrease of 5 to 10 dB depending on the overall amplitude of the notes involved in the transition.

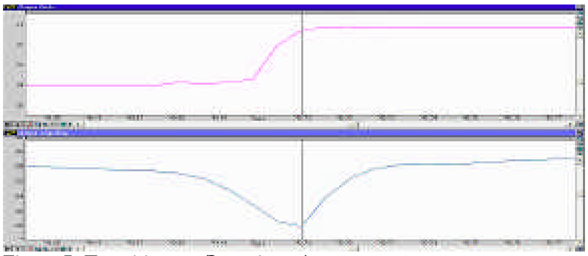

Figure 5: Transition up (3 semitones)

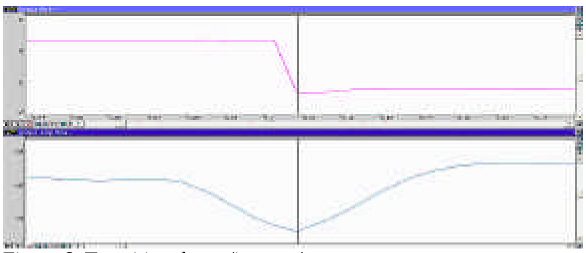

Figure 6: Transition down (1 octave)

In figure 7 we can see the model created, that as a first approach will be the same for all the transitions up or down without taking into account the number of semitones of the interval or the notes location in the register. We should control the number of frames of the transition depending on the dynamic of the notes implied. Usually for piano dynamics, the number of frames for the transition will be bigger and for forte dynamics or when the previous notes are very short, the number of transition frames should be shortened to avoid too long transitions that modulate the whole target stationary producing a vibrato effect.

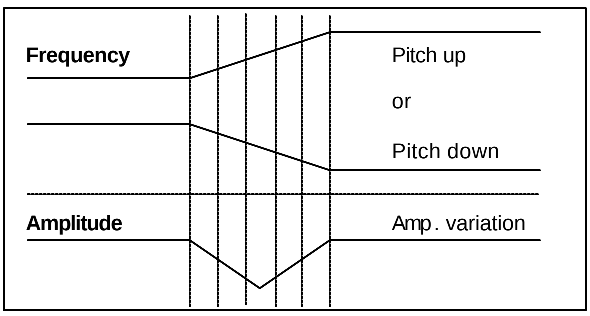

Figure 7: Pitch and amplitude model for transitions (up or down)

6 Conclusions and Future Work

Implementing the new SPP synthesis technique the sound quality achieved is quite better than in the previous implementation that used SMS. The main features that have been improved in this implementation are the timbre interpolation between different dynamics and the memory saving because we need less spectral segments in the database to do the synthesis. The next steps in the project will focus in the implementation of polyphonic synthesis and harmonization, based on to combine various instruments to synthesize a complete wind section that harmonizes a melody being played by a single player.

8 Acknowledgments

The authors of this article would like to thank Joachim Haas who was the main developer in the first implementation of the synthesizer.

7 References

[1] Haas, J., “SALTO: A Spectral Domain Saxophone Synthesizer”, Proceedings of the Mosart Workshop, Barcelona 2001.

[2] Serra X., Bonada J., et al., “Integrating Complementary Spectral Models in the Design of a Musical Synthesizer”, Proceedings of the ICMC, Thessaloniki 1997.

[3] Bonada, J., Loscos, A., “Spectral Peak Processing Documentation” Internal MTG paper, 2001.

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

[5] Rovan J.B., Wanderley M.M., Dubnov S., Depalle P., “Instrumental Gestural Mapping Strategies as expressive Determinants in Computer Music Performance”, AIMI International Workshop, Genova.

[6] Serra, M.H., Rubine, D., Dannenberg, R., “Analysis and Synthesis of Tones by Spectral Interpolation” Journal of the Audio Engineering Society, 38(3) (March 1990), pp. 111-128.

[7] Derenyi, I., Dannenberg, R. “Synthesizing Trumpet Performances” in Proceedings of the International Computer Music Conference. San Francisco (1998).