A Real-Time Analysis and Resynthesis Instrument for Transformation of Sounds in the Frequency Domain

Todor Todoroff
Faculté Polytechnique de Mons
todor@musique.fpm.ac.be

Abstract

We have programmed several external MAX/FTS modules running on the JSPW-16. They perform FFT analysis and resynthesis and allow different kinds of transformations in the frequency domain: denoising and transformation through thresholding or the frequency bins, transposition and pitch shifting, very flexible filtering with real-time interpolation between several transfer functions and cross-synthesis with real-time control on the cepstral smoothing of the spectral envelope of an auxiliary signal. All these transformations may be combined and most of the parameters values may be accessed in real time. This allows complete control over the creation of complex sound morphologies.

I Introduction

There are several programs performing sound transformations through FFT analysis and resynthesis. Programs like IRCAM SVP [Decalè and Pointe, 1991; Serra, 1993] and AudioSculpt produce high-quality sounds with a high degree of versatility. But we wanted more than just tools to process sounds: we wanted a real instrument that could be played in real time with full gestural control over the transformation parameters. We designed our instrument primarily in order to allow composers of acoustic music to produce complex sound morphologies with constant audio feedback. But the goal also matches the needs of composers who wish to perform live transformation of instrumental sounds. Our external modules were used by Emmanuel Nenot for the creation of his last work, "Lichtung 2", for instruments and two NeXT Cubes fitted each with 1 JSPW-H boards. The implementation was done by Eric Daubressé, musical assistant at IRCAM.

Rather than using standard MAX modules as proposed by Scelsi and Lippe [1994], we choose to write external objects in order to be able to implement more functionalities. By combining several modules in one single external object we significantly reduce delays normally introduced in the signal flow when transferring signals between modules. And the considerable amount of processing time usually wasted in control objects may be used for other purposes: for example, writing an external object to interpole between several large sets of data allows a kind control that would produce the slips if we had used dozens of edit and multiply objects. On the other hand, we didn't want to sacrifice the modularity which is without any doubt one of the great features of MAX. We therefore designed three modules: an synth- actually performs the analysis and resynthesis, while cepstral smooth- extracts the cepstrally smoothed envelope of an auxiliary signal and gabinet- generates an amplitude filtering transfer function with a very flexible control. The output of these last two modules may be combined in any way prior to sending them to the main module. Figure 1 depicts schematically how one may combine these three modules.

Figure 1: Connecting the external modules together.
2 The an\_synth\_ module

This main module implements an overlap-add analysis and resynthesis. The FPT sizes, up to 1024 points, and the overlapping factor, up to 75 percent, are defined by the creation arguments of the object. The window-type, as well as all the other parameters, may be modified in real-time by sending the right message to the module and chosen between 8 of the most common ones (different windows may produce different timbres when filtering is used). Three kinds of simultaneous available frequency-domain transformations are proposed.

2.1 Thresholds: a lower threshold may be applied and allows for efficient denoising of signals. The threshold control may be added to a frequency-dependent threshold drawn by the user in a MAX table. The Frequency bins falling below the threshold may either be suppressed, attenuated or reinforced. When increasing the threshold while suppressing the components matching the threshold, the signal is first denoised, then transformed up to a point where only the resonances are heard. One might describe the effect as a kind of liquefaction of the sound. An upper threshold function may be used simultaneously and offers the same controls. The effect is more difficult to describe as is more dependent on the sound being transformed. If, for example, it contains a combination of steady and percussive sounds, the latter may be increased above level or attenuated hardly without modifying the steady sounds.

2.2 Spectrum multiplication: the second signal input may be used to multiply the real and imaginary parts of each frequency bin by an external signal. cepstral\_smooth\_ and gibbard\_ where specially designed for this purpose, but other modules may be used as well.

2.3 Frequency axis modifications: we were not able to implement phase extraction as we needed to be able to run everything on one single ISIP\_W\_1e card. But we implemented a simple algorithm to perform transpositions, pitch-shifting and non-linear distortion of the frequency axis. Those transformations are far from perfect, but may nevertheless produce very interesting sounds and create various kinds of inharmonicities. The results are specially interesting when controlling the parameters with line modules with a grain of 4, which corresponds to the number of cycles between successive analysis and synthesis windows when using 1024 points FPTs with 75 percent overlapping.

3 The cepstral\_smooth\_ module

This module performs an FFT analysis of its signal input. It then performs the FFT of its log magnitude spectrum, the cepstrum. After having zeroed a number of cepstral coefficients according to the messages sent at the input of the module, it inversion. FPT is computed, the values are converted back to a linear scale and sent to the output. Combined with an\_synth\_, cepstral\_smooth\_ produces very high quality vocoder-type effects. With its 512 filter bands, compared to 10 or 20 bands in commercially available analogue vocoders, it preserves remarkably the intelligibility of the voice and is even able to superimpose its harmonic structure on the original signal. The real-time control over the smoothing parameter is something that we had never seen or heard before. It proves to be very effective.

![Figure 2: Cepstral smoothing, transposition and frequency-shifting.](image)

ICMC Proceedings 1996 133 Tedoroff
We also added the possibility to freeze one analysis frame, as well as to transpose and frequency-shift the spectrum. The tables on the left of Figure 2 show one fourth spectrum of a vowel signal where the smoothing was increased from top to bottom. The tables on the right show the last spectrum envelope transported, frequency-shifted, and passed through a combination of both. All those values may be changes in real-time.

4 The gubarti- module

This module offers the most complex and flexible control. It is designed to generate filters in combination with an_synth- filters. Filters may be defined with up to 128 points, 300 of them may be stored in or recalled from memory presets; those banks of presets may be saved to or loaded from disk, and it is possible to interpolate in real-time between up to eight different presets. Figure 3 shows one possible MAX patch to control and display the values of this module.

![Figure 3: MAX patch to control a gubarti-module with 16 ($f$, $h$) points, 32 presets and 8 filter interpolation.](image)

The process of creating a transfer function is based on the definition of (frequency, amplitude) points. This might be done by moving virtual faders, through number boxes, message boxes or with the help of any MIDI controller. Frequencies may be entered either in Hz or by the number of the corresponding FFT point. When using gubarti- with a 1024 point FFT or synth- module, the values are updated every 512 samples. The user may switch at any moment to one out of three different modes of using those points to generate a transfer function: making either a bandpass filter by drawing horizontal lines, or a bandreject filter by linking the points with straight lines in a linear or logarithmic scale. Figure 4 shows the results for those modes on the left side.

Sound morphologies are easy to control when using MIDI faders assigned to the amplitude and/or frequencies of the points used to define the transfer function. But even more interesting results are obtained when performing interpolation between several presets. Two kinds of interpolation have been implemented. The first one computes the transfer function for every preset chosen and then adds them after multiplication by their respective weights (see top right table in Fig. 4). Those weights might be generated with the mouse using a program like Circapp developed by Gerhard Eckel at IRCAM or with our interpolation object [Toderoff and Traube, 1986], or with any other reason.

The second kind of interpolation doesn't interpolate on the transfer functions, but directly on the definition points. A new set of interpolated points is then created and the user may choose between the three modes already described to connect the transfer functions. This form of interpolation leads not only to faders in and out of transfer functions, but also to the points frequencies being interpolated, creating spectral glissandos (quite niceable in bottom right table in Fig. 4).

Even while interpolating, it is always possible to modify any or all of the points being interpolated, as well as changing the values of any point of those presets, their modes or the kind of interpolation.

Banks of presets may be saved to disk, allowing one to construct a library of transfer functions. And, as it is also possible to load small banks of presets into larger ones and to copy presets from one location to another, one can make a new bank by taking presets from different banks already saved to disk.

Toderoff
5 Summary

We have designed a set of modules performing various kinds of transformations of sound signals. As these transformations are introduced in the frequency domain, they are very intuitive to work with. Thanks to real-time and simultaneous access to every parameter, the user may quickly experiment with different settings. And the interpolation facilities of the gaboris module are an invitation to experiment with complex and yet precise sound morphologies, either for composing new music or to transform live instrument sounds in concert. This real-time control also allows many kinds of interaction with a sequencer or with a MAX patch, making it a useful tool to experiment complex dynamic timbre modulations. We are currently experimenting different types of parameter mapping to control this instrument with MIDI faders and a PowerGlove interfaced through the STEIM SensorLab.

References


This research has been funded by the Région Wallonne, in Belgium.