HAL open science

# Applications for Real-time FFT-based Resynthesis

Zack Settel, Cort Lippe

▶ **To cite this version:**

Zack Settel, Cort Lippe. Applications for Real-time FFT-based Resynthesis. Journées d'informatique musicale, Mar 1994, Bordeaux, France. pp.61-67. hal-03347045

**HAL Id: hal-03347045**

**https://hal.science/hal-03347045**

Submitted on 17 Sep 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

# Applications for Real-time FFT-based Resynthesis

Zack Settel & Cort Lippe
*IRCAM, 1 place Stravinsky, Paris, 75004, France*
email: settel@ircam.fr & lippe@ircam.fr

## Introduction

The Fast Fourier Transform (FFT) is a powerful general-purpose algorithm widely used in signal analysis. FFTs are useful when the spectral information of a signal is needed, such as in pitch tracking or vocoding algorithms. The FFT can be combined with the Inverse Fast Fourier Transform (IFFT) in order to resynthesize signals based on its analyses.This application of the FFT/IFFT is of particular interest in electro-acoustic music because it allows for a high degree of control of a given signal's spectral information (an important aspect of timbre) allowing for flexible, and efficient implementation of signal processing algorithms.

This paper presents real-time musical applications using the IRCAM Signal Processing Workstation (ISPW) [Lindemann, Starkier, and Dechelle 1991] which make use of FFT/IFFT-based resynthesis for timbral transformation in a compositional context. Taking a pragmatic approach, the authors have developed a user interface in the Max programming environment [Puckette, 1988] for the prototyping and development of signal processing applications intended for use by musicians. Techniques for filtering, cross-synthesis, noise reduction, and dynamic spectral shaping have been explored, as well as control structures derived from real-time signal analyses via pitch-tracking and envelope following [Lippe & Puckette 1991]. These real-time musical applications offer composers an intuitive approach to timbral transformation in electro-acoustic music, and new possibilities in the domain of live signal processing that promise to be of general interest to musicians.

## The FFT in Real Time

Traditionally the FFT/IFFT has been widely used outside of real-time for various signal analysis/re-synthesis applications that modify the durations and spectra of pre-recorded sound [Haddad & Parsons 1991]. With the ability to use the FFT/IFFT in real-time, live signal-processing in the spectral domain becomes possible, offering attractive alternatives to standard time-domain signal processing techniques. Some of these alternatives offer a great deal of power, run-time economy, and flexibility, as compared with standard time-domain techniques [Gordon & Strawn 1987]. In addition, the FFT offers both a high degree of precision in the spectral domain, and straightforward means for exploitation of this information. Finally, since real-time use of the FFT has been prohibitive for musicians in the past due to computational limitations of computer music systems, this research offers some relatively new possibilities in the domain of real time.

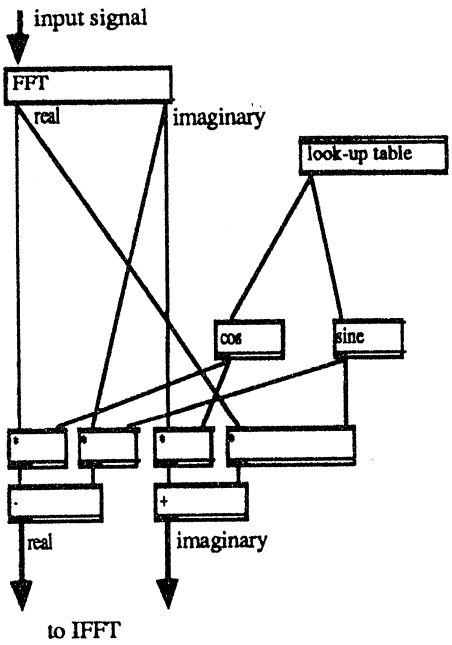
## Programming Environment

Our work up to this time has been focused on real-time signal processing applications involving the spectral modification of sounds. (We hope to attack the problem of time-stretching at a later date.) Since we are constructing our signal processing configurations in Max using a modular patching approach that includes both time-domain and frequency-domain modules, we are able to develop hybrids, discussed below, that combine standard modules of both types. Development in the Max programming environment [Puckette, 1991] tends to be simple and quite rapid: digital signal processing (DSP) programming in Max requires no compilation; control and DSP objects run on the same processor, and the DSP library provides a wide range of unit generators, including the FFT and IFFT modules.

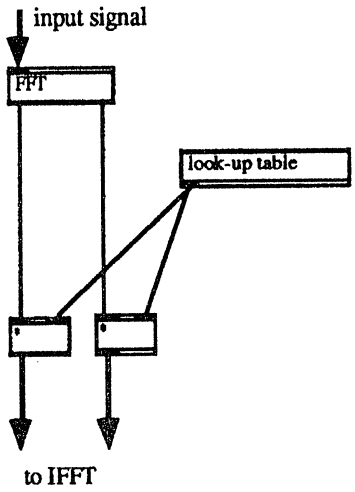
## Algorithms and basic operations

All of the signal processing applications discussed in this paper modify incoming signals and are based on the same general DSP configuration. Using an overlap-add technique, the DSP configuration includes the following steps: (1), windowing of the input signals, (2) transformation of the input signals into the spectral domain using the FFT, (3) operations on the signals' spectra, (4) resynthesis of the modified spectra using the IFFT, (5) and windowing of the output signal. Operations in the spectral domain include applying functions (often stored in tables), convolution (complex multiplication), addition, and taking the square root (used in obtaining an amplitude spectrum); data in this domain are in the form of rectangular coordinates. Due to the inherent delay introduced by the FFT/IFFT process, we use 512 point FFTs for live signal processing when responsiveness is important. Differences in the choice of spectral domain operations, kinds of input signals used, and signal routing determine the nature of a given application: small changes to the topology of the DSP configuration can result in significant changes to its functionality. Thus, we are able to reuse much of our code in diverse applications. For example, though functionally dissimilar, the following two applications differ only slightly in terms of their implementation.*see figure1.*

**phase rotation**

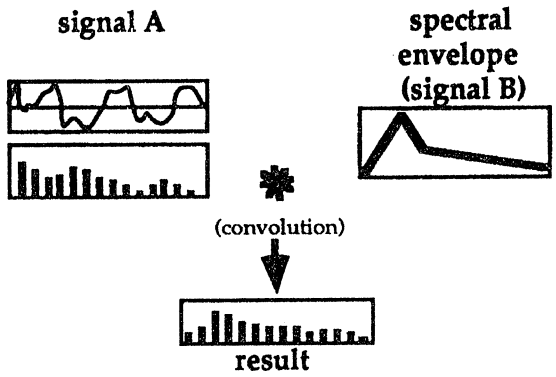



**filtering**

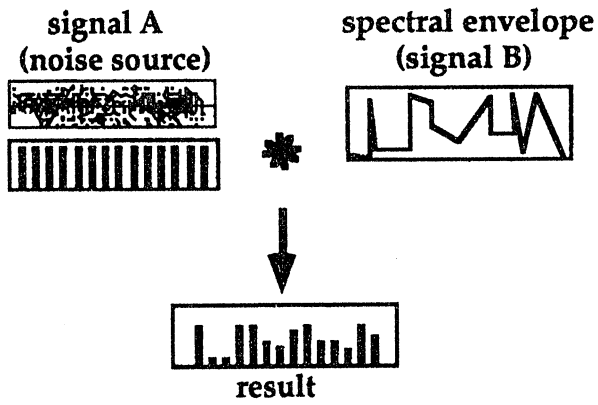


## Applications

*High-resolution filtering*
Highly detailed time varying spectral envelopes can be produced and controlled by relatively simple means. A look-up table can be used to describe a spectral envelope in the implementation of a graphic EQ of up to 512 bands. The spectrum of the input signal is convolved, point by point, with the data in the look-up table, producing a filtered signal. Because we are able to alter the spectral envelope in real time at the control rate (up to 1kHz), we may modify our spectral envelope graphically or algorithmicaly, hearing the results immediately. *see figure 2.*



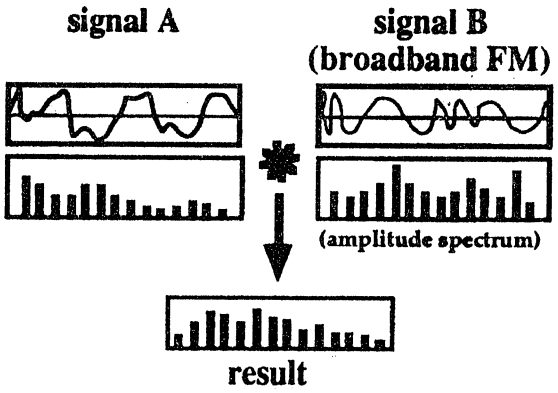

Using a noise source as the input signal, it is also possible to do *subtractive synthesis* efficiently. *see figure 3.*




### Low dimensional control of complex spectral envelopes

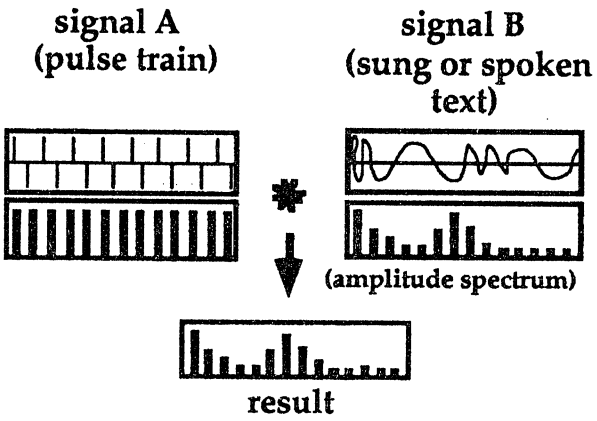
The spectral envelope used in the above filtering application can also be obtained through signal analysis, in which case a second input signal, signal B, is needed. Signal B is analyzed for its spectral envelope, or amplitude spectrum, that describes how signal A will be filtered. Obtaining a spectral envelope from an audio signal offers several interesting possibilities: spectral envelopes can be "collected" instead of being specified, and can change at a very fast rate (audio rate), offering a powerful method of dynamic filtering. Also, audio signals produced by standard signal processing modules such as a frequency modulation (FM) pair (one oscillator modulating the frequency of another) are of particular interest because they can produce rich, easily modified, smoothly and nonlinearly varying spectra [Chowning 1973] which can yield complex time varying spectral envelopes. These spectral envelopes can be modified using only 3

(FM) parameters: carrier frequency, modulator frequency, and modulation index. Likewise, other standard signal processing modules such as an amplitude modulation (AM) signal generator, an additive synthesis instrument, or a band-pass filter bank offer rich varying spectral information using relatively simple means with few control parameters. One of the advantages of using standard modules is that electronic musicians are familiar with them, and have a certain degree of control and understanding of their spectra. *see figure 4.*



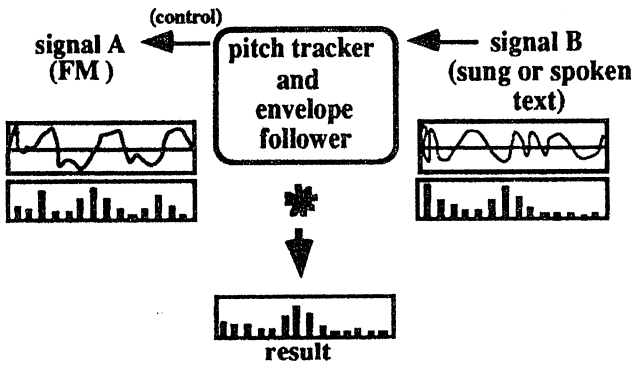

## Cross synthesis

In this application two input signals are required: signal A's spectrum is convolved with the amplitude spectrum of signal B. Thus, the pitch/phase information of signal A and the time varying spectral envelope of signal B are combined to form the output signal. Favorable results are produced when Signal A has a relatively constant energy level and broadband spectrum, and when signal B has a well defined time varying spectral envelope. For example, when wishing to transform spoken or sung text, we assign the text material to signal B while specifying a pulse train, noise source or some other constant-energy broadband signal to signal A. Since the frequency information (pitch, harmonicity, noise content, etc.) of signal A is retained in the output, unusual effects can be produced when frequency related changes occur in signal A. In the following example of a *vocoder*, text can be decoupled from the speaker or singer's "voice quality", allowing one to modify attributes of the voice such as noise content, inharmonicity, and inflection, independently of the text material. *see figure 5.*



## Mapping qualities of one signal to another

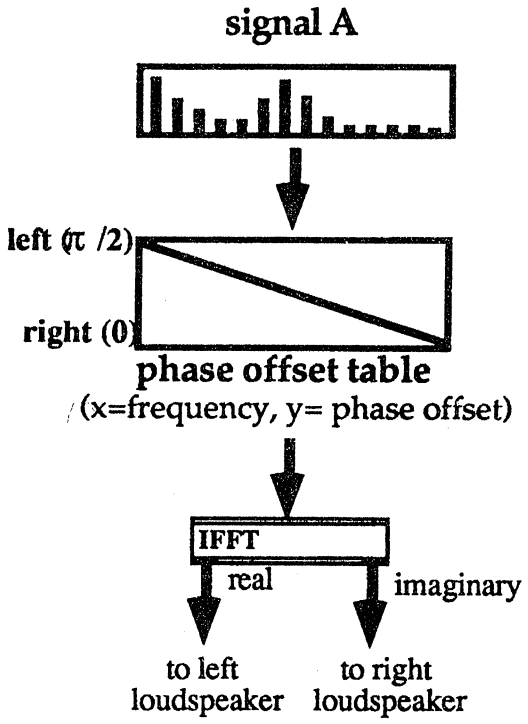
A simple FM pair may be used to provide an easily controlled, constant-energy broadband spectrum for use in cross synthesis as signal A. Musically, we have found that in some cases, the relationship between signal A and signal B can become much more unified if certain parameters of signal B are used to control signal A. In other words, real-time continuous control parameters can be derived from signal B and used to control signal A. For example, the pitch of signal B can be tracked and applied to signal A (FM) to control the two oscillators' frequencies. Envelope following of signal B can yield expressive information which can be used to control the intensity of frequency modulation (FM index) of signal A . In experiments incorporating the above, a mezzo soprano's voice was assigned to signal A, while her pitch and intensity were mapped onto signal B (FM), producing striking results akin to harmonization and frequency shifting.
*see figure 6.*




Finally, it should be noted that interesting transformations can be produced by simply convolving signal A's spectrum with signal B's spectrum. In this case, the phase (frequency) and spectral envelope information from each signal figures in the output signal. Transformations of broadband sounds, akin to, but more pronounced than flanging, can be produced when convolved with the signal of a high index, inharmonically tuned FM pair, whose frequency parameters are controlled by the pitch of the first signal.

## Frequency dependent spatialization

In the spectral domain, the phases of a given signal's frequency components can be independently rotated in order to change the component's energy distribution in the real and imaginary part of the output signal. Since the real and imaginary parts of the IFFT's output can be assigned to separate output channels, which are in turn connected to different loud-speakers, it is possible to control a given frequency's energy level in each loud-speaker using phase rotation. The user interface of this application permits users to graphically or algorithmically specify the "panning" (phase offset) for up to 512 frequency components. *see figure 7.*




### Frequency dependent noise-gate

In the spectral domain, the energy of a given signal's frequency components can be independently modified. Our noise reduction algorithm is based on a technique [Moorer & Berger, 1984] that allows independent amplitude gating threshold levels to be specified for each frequency component in a given signal. With a user-defined transfer function, the energy of a given frequency component can be altered based on its intensity and particular threshold level. This technique, outside of being potentially useful for noise reduction, can be exaggerated in order to create unusual spectral transformations of input signals, resembling extreme chorusing effects.

## Future Directions

The authors are currently working on alternative methods of sampling that operate in the spectral domain. Many interesting techniques for sound manipulation in this domain are proposed by the phase vocoder [Dolson 1986; Nieberle & Warstat 1992]. Along with the possibility of modifying a sound's spectrum and duration independently, we would like to perform transposition independent of the spectral envelope (formant structure), thus allowing one to change the pitch of a sound without seriously altering its timbral quality.

## Conclusion

With the arrival of the real-time FFT/IFFT in flexible, relatively general, and easily programmable DSP/control environments such as Max, non-engineers may begin to explore new possibilities in signal processing. Though our work is still at an initial stage, we have gained some valuable practical experience in manipulating sounds in the spectral domain. Real-time convolution can be quite straightforward and is a powerful tool for transforming sounds. The flexibility with which spectral transformations can be done is appealing. Our DSP configuration is fairly simple, and changes to its topology and parameters can be made quickly. Control signals

resulting from detection and tracking of musical parameters offer composers and performers a rich palette of possibilities lending themselves equally well to studio and live performance applications.

## Acknowledgements

The authors would like to thank Miller Puckette, Stefan Bilbao, and Philippe Depalle for their invaluable technical and musical insights.

## References

Chowning, J. 1973. "The Synthesis of Complex Audio Spectra by means of Frequency Modulation" *Journal of the Acoustical Society of America* 21(7), 526-534.

Dolson, M. 1986. "The phase vocoder: a tutorial", *Computer Music Journal*, 10(4), 14-27

Gordon, J. and Strawn J. 1987. "An introduction to the phase vocoder", *Proceedings, CCRMA*, Department of Music, Stanford University, February 1987.

Haddad R, and Parsons,T. 1991. "Digital Signal Processing, Theory, Applications and Hardware", New York: Computer Science Press

Lindemann, E., Starkier, M., and Dechelle, F. 1991. "The Architecture of the IRCAM Music Workstation." Computer Music Journal 15(3), 41-49.

Lippe, C. and Puckette, M. 1991. "Musical Performance Using the IRCAM Workstation", *Proceedings of the 1991 International Computer Music Conference*. San Francisco: International Computer Music Association.

Moorer and Berger, 1984. "Linear-Phase Bandsplitting: Theory and Applications", Audio Engineering Society (preprint #2132), New York: 76th AES convention 1984.

Nieberle, R and Warstat, M 1992. "Implementation of an analysis/synthesis system on a DSP56001 for general purpose sound processing", *Proceedings of the 1992 International Computer Music Conference*. San Francisco: International Computer Music Association.

Puckette, M. 1988. "The Patcher." *Proceedings of the 1988 International Computer Music Conference*. San Francisco: International Computer Music Association.

Puckette, M., 1991. "FTS: A Real-time Monitor for Multiprocessor Music Synthesis." Music Conference, 420-429, San Francisco: International Computer Music Association.

A different version of this article will be published by Harwood Academic Publishers (Switzerland).