AUDIO-RATE CONTROL OF FFT-BASED PROCESSING USING FEW PARAMETERS

Zack Settel

McGill University
Music Faculty
555 rue Sherbrooke Ouest
Montreal, Quebec H3A 1E3
CANADA
zack@music.mcgill.ca

Cort Lippe

University at Buffalo
Department of Music
Hiller Computer Music Studios
222 Baird Hall
Buffalo, NY, USA 14260
lippe@buffalo.edu

ABSTRACT

Though the use of the Fast Fourier Transform (FFT) for signal processing in music applications has been widespread, applications in real-time systems for dynamic spectral transformation has been quite limited. The limitations have been largely due to amount of computation required for the operations. With faster machines, and with suitable implementations for frequency-domain processing, real-time dynamic control of high-quality spectral processing can be accomplished with great efficiency and simple approach. This paper will focus on dynamic real-time control of frequency-domain-based signal processing, and will describe the author's latest work (hi-resolution filtering and spatialization implementations) in this area. General background on the implementation and the development environment (Max Signal Processing, MSP) will also be provided.

1. INTRODUCTION

We employ the standard procedures commonly used when processing audio signals via the FFT, including: (1) windowing of the time-domain input signal, (2) transformation of the input signal into a frequency domain signal (spectrum) using the FFT, (3) various frequency-domain operations such as complex multiplication for convolution, (4) transformation of the frequency-domain signals back into the time domain using the IFFT, (5) and windowing of the time-domain output signal.

1.1 Development and Implementation

The development environment used by the authors, Max and Max Signal Processing (Msp) [1], has evolved from the Max software developed by Miller Puckette for the Ircam Signal Processing Workstation (ISPW) [2]. This environment facilitates the development of real-time general purpose audio applications. The FFT object provided in Msp is based on Miller Puckette's ISPW implementation [3] and stores time-domain signals as buffers of samples upon which the FFT analysis is done. The FFT object outputs each frame, bin-by-bin, using three sample streams running at the sampling rate. Thus, each bin is represented by three samples consisting of "real" and "imaginary" values, and the bin number (index). The IFFT is the complement of the FFT and expects as input, real and imaginary values in the same format as the FFT output.

As seen in Figure 1, the index values provide a synchronization phasor, making it possible to identify bins within a frame, and recognize frame boundaries. The index values can be used to access bin-specific data for various operations, such as attenuation or spatialization, and to read lookup tables for windowing.

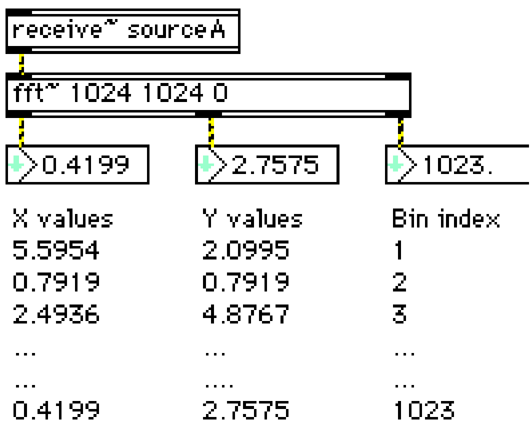


Figure 1. *sample-by-sample output of the FFT object*

1.2 Audio-Rate Control of FFT-Based Processing

The Max/Msp environment has two run-time schedulers: the Max "control" scheduler, which is timed on the basis of milliseconds, and the Msp "signal" scheduler, which is timed at the audio sampling rate [4]. In FFT-based processing applications, where changes to the resulting spectrum are infrequent, Msp's control objects may be used to provide control parameters for the processing. This is both precise and economical, but has bandwidth limitations. Significant and continuous modification of a spectrum, as in the case of a sweeping band-pass filter, is not possible using Msp's control objects, since they can not keep up with the task of providing 1024 parameter changes at the FFT frame rate of 43 times a second (using FFT buffers of size 1024 at the audio sampling rate of 44,100 samples per second). A more dynamic approach to filtering is to update lookup tables containing filter functions at the signal rate (the audio sampling rate.) The term "Spectral Processing Function" (SPF) will be used frequently in this text and refers to a lookup table-based function (actually a signal), whose length is that of the FFT. For each window of input signal (FFT data) which we receive in real-time, we generate a corresponding SPF with which the input may be convolved. Dynamically, the SPF can describe a particular sequence of forms (or spectral envelopes), which determine the time-varying intensity of spectral processing by individual frequency components via convolution; a "form" describes the action of the spectral processor. Thus, our approach to dynamic processing spectra focuses on efficiently generating forms with a potentially high degree of detail using simple and intuitive descriptions [5]. In this paper, we will explore two methods for generating SPFs: the first involves the generation of forms using the spectral envelope of a signal (via FFT), while the second makes use of table-based waveform generation techniques. In both cases, low-dimensional parametric control of complex forms is achieved, and a time-varying SPF is generated for, and convolved with, each window of input signal to perform filtering, band-limited panning, or "spectral mixing" of two or more signals.

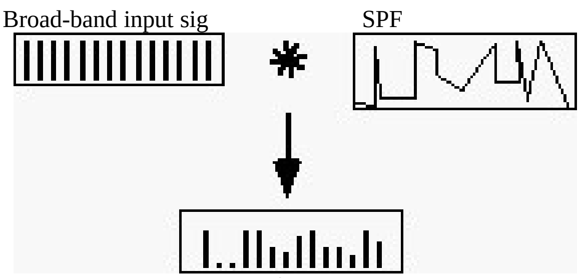


Figure 2. *a signal is filtered (via convolution) using an SPF*

2. GENERATING COMPLEX FORMS

When searching for a suitable method for generating complex forms (or spectral envelopes) using simple techniques, the classic FM pair algorithm (generating complex spectra using frequency modulation), with its simple structure and inherent potential for generating highly detailed and complex spectra, leaps to mind. The SPF is simply *the spectral envelope* of the signal generated via FM, and it provides a rich source of possible forms, whose shapes and complexity are determined using the FM algorithm's few parameters (carrier frequency, carrier-modulator ratio, modulation index). The intuitive mapping of the FM parameter values to the resulting spectral form make this method extremely easy to use. In our implementation, a particular set of static or time-varying parameters is specified for the FM algorithm; the parameters determine the shape of the resulting SPF.

Other sources for generating complex spectra, such as amplitude modulation, additive synthesis, or waveshaping, may also be used in the implementation above. However, the clear advantage of using FM lies in the simple control of the highly complex spectra it offers.

2.1 Waveform Generators

The use of wave tables and basic table lookup operations provides a general and flexible approach to the well-known waveform generation and synthesis techniques used to generate our Spectral Processing Functions (SPF). Techniques such as FM, AM, waveshaping, phase modulation, and pulse-width modulation all have the potential to provide complex, evolving waveforms, which can be used as SPFs to provide a high level of flexibility and detail for spectral processing techniques such as filtering, spectral panning or spectral mixing. Table lookup operations such as inversion, scaling, offsetting, wrapping, and nonlinear distortion (waveshaping), provide powerful means for dynamically modifying these SPFs. For example, nonlinear indexing of lookup tables can be employed to provide dynamic control for a constant-Q bandpass filter, where control on a nonlinear frequency scale is required. Most important, the parameters of these waveform-based techniques are few, well-known, and easy to understand. As discussed above, the FFT used in our implementation provides us with a phasor that we use as an index for table lookup. Note that the index is mirrored around the frequency at half the sampling rate, following from the symmetrical (real) spectrum output of the FFT.

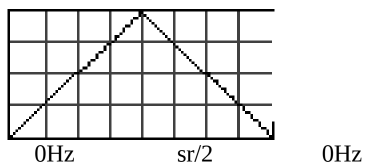


Figure 3. *phasor from the FFT's output*

The following sequence of operations is performed on the phasor; note how its shape (form) is changed by each operation. (Note that steps 3, 6, and 7 above entail loss of information; their placement in the sequence of operations can not be arbitrary.)

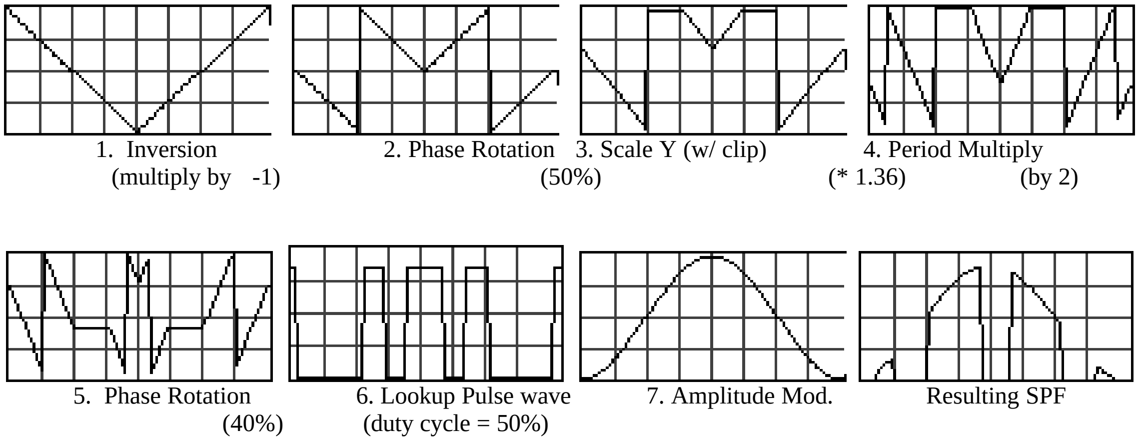
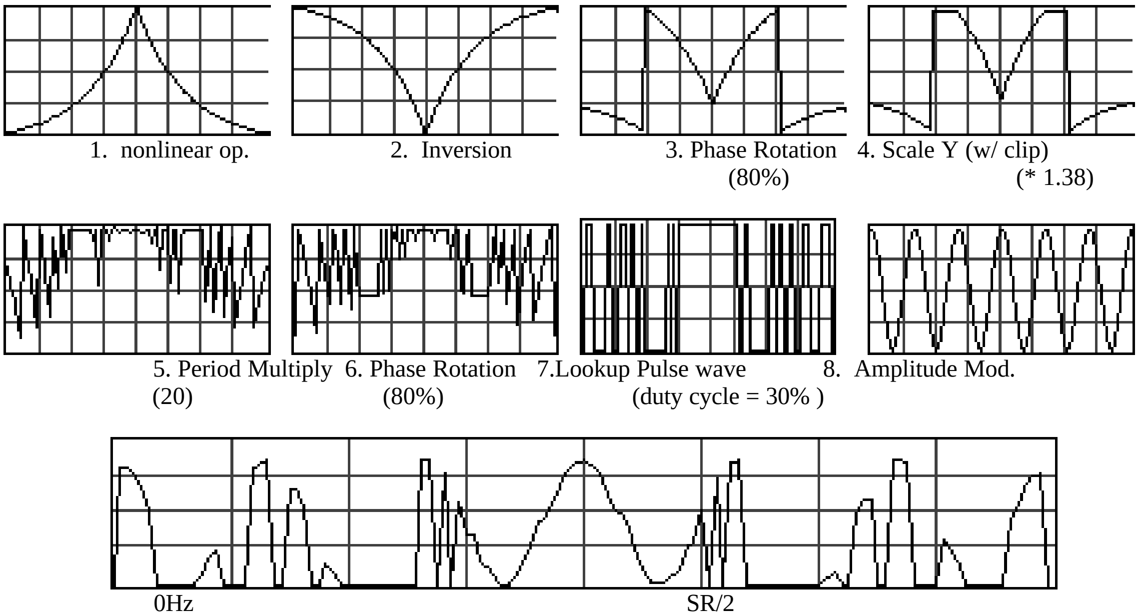


Figure 4. *lookup table operations*



0Hz

Resulting Spectral Processing Function (SPF)

Figure 5. *generating a complex SPF*

While the operations and parameter settings in Figure 4 will produce a static SPF, it is important to note that the "phase rotation" operation also accepts a frequency parameter (LFO); a non-zero value will cause the phase rotation value to change constantly. Thus, the resulting SPF will vary periodically due to the constant change of its phase rotation parameter. This "LFO-based" technique provides control for periodic spectral modulation, and is useful in implementations such as swept band-pass filters (or spectral panners), comb filters, or "phase shifters". To generate a more complex SPF, we start the same phasor above, then apply the operations shown in Figure 5.

3. TECHNIQUES

The above techniques for dynamic spectral processing work particularly well in applications such as high-resolution dynamic filtering, and spectral panning. The authors have implemented such applications, and have experimented with spectral resolution, making use of 1024 to 4096 point FFTs, twice or four times overlapped. Given the implicit tradeoff between time and frequency resolution, and considerations of computational cost, the use of 1024 point FFTs is preferable for real-time performance situations when audio signal latency (delay from input to output) is an issue. However, the authors found that a longer FFT of 4096 points is generally preferable, despite the relatively low spectral update rate of 10Hz (10 FFT frames per second), since a long window allows for a (correspondingly long) highly detailed spectral processing function (SPF.) For example, the "period multiply" operation shown above requires a long SPF when a greater number of periods is specified. In the case of high resolution filtering applications, the SPF specifies the degree of attenuation to be applied to each component (band-limited region or bin) of the input spectrum. The input signal's spectrum is convolved with the SPF and filtering is accomplished. The nature of the filter is determined by the form of the SPF. In the case of an application for spectral panning (band-limited panning), the SPF is used to specify the degree of phase rotation (via convolution with a complex sinusoid) to be applied to the components of the input spectrum. The operation results in a change to a given component's energy distribution in its real and imaginary parts. The real and imaginary outputs of the IFFT are mapped to a corresponding stereo output. Dynamic spectral panning occurs when the phase of the input signal's spectral components is rotated by a changing amount. And, in the case of spectral mixing of two or more signals, the SPF is used to specify the degree to which a given frequency bin will be present for a given signal.

Compressor/limiters are examples of signal processors that analyze the input signal to determine how it is in turn to be modified. A signal passing through such a processor can be thought of as a self-modifying signal. When an analysis stage (such as an envelope or pitch follower) is added to our spectral processing implementation, the resulting input analysis information may be used to dynamically control one or more parameters of the spectral processing. Since our interface for spectral processing requires only a small number of parameters, and since each parameter can significantly alter the shape of the SPF (offering a wide potential range of transformational possibilities), the mapping of few (even just one) input-derived control streams to processing parameters can provide a very high degree of signal self-modification. Musically, the ability to control the degree and quality of a signal's spectral transformation via aspects of that signal's dynamics or inflection can be compelling; in live performance applications, this mapping of musical gesture to resulting timbre can be quite tangible and inspiring to the performer.

A particularly effective example of this sort of mapping can be produced using the waveform generation technique discussed earlier, where the control derived from the input intensity is mapped to the "period multiply" parameter of the processing. Thus, the more intense the input signal, the greater the number of peaks and notches in the SPF. In a filtering application, the effect of increasing the input dynamic level translates into increasing separation (as opposed to fusion) of spectral components into discretely perceived sinusoids. In a spectral panning application, increasing dynamic level translates into increasing stereo separation of neighboring components, enhancing the perception of sound source location. Alternatively, a similar result can be achieved using the spectral generation technique, discussed earlier in this paper. In this case, an FM pair is used to generate the SPF, and the input intensity is mapped to the "modulation index" parameter of the processing.

There are many other possible mappings of input features to spectral processing parameters. In each case, a given aspect of the input signal will potentially cause the spectrum of that same signal to be modified in a significant way. The challenge is to recognize the practical/musical sense of certain choices (mappings), which in turn, can serve a musical purpose in performance and/or composition. The approach of playing an instrument while processing its sound (as described above) has proven to be a very effective way to evaluate the mappings and discover their particular musical tendencies. As with dynamics processors, the choices of input signal, analysis mapping and processing technique produce results that cover a wide range of timbral possibilities. Future work will be concerned with discovering and establishing musically effective combinations of these choices.

4. CONCLUSIONS

The use of spectral generation and waveform-based techniques for low-dimensional audio-rate control of FFT-based processing has great potential: the parameters are few, familiar and easy to control, and direct mappings of real-time audio input from musicians to the control of FFT-based DSP is made possible. While the applications in this paper has been limited to filtering, spectral panning, and spectral mixing; audio-rate control of FFT-based processing applies equally well to any FFT-based applications where a high degree of processing control is required at the frame rate. The authors have also implemented the techniques mentioned above in applications for denoising and dynamics processing. We believe that these techniques hold great promise for control-intensive FFT-based applications, especially for live-performance DSP, where a new level of interactivity for a relatively new set of frequency-domain DSP techniques, is now possible.

5. ACKNOWLEDGMENTS

The authors would like to thank Miller Puckette, David Zicarelli and i Fratelli Convoluzione.

6. REFERENCES

[1] Zicarelli D., 1997. "Cycling74 Website", www.cycling74.com

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

[3] Puckette, M., 1991. "Combining Event and Signal Processing in the Max Graphical Programming Environment", Computer Music Journal 15(3), pp. 68-77.

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

[5] Settel, Z., and Lippe, C., 1995. "Real-time Musical Applications using Frequency-domain Signal Processing", IEEE ASSP Workshop Proceedings, Mohonk, New York.