Experiments with dynamic convolution techniques in live performance

Øyvind BRANDTSEGG
Music technology
NTNU - Department of Music
NO-7491 Trondheim, Norway
oyvind.brandtsegg@ntnu.no

Sigurd SAUE
Music technology
NTNU – Department of Music
NO-7491 Trondheim, Norway
sigurd.saue@ntnu.no

Abstract

This article discusses dynamic convolution techniques motivated by the musical exploration of interprocessing between performers in improvised electroacoustic music. After covering some basic challenges with convolution as live performance tool we present experimental work that enables dynamic updates of impulse responses and parametric control of the convolution process. An audio plugin implemented in the open source software Csound and Cabbage integrates the experimental work in a single convolver.

Keywords

Convolution, live processing, Csound, cross synthesis

1 Introduction

The Music Technology section at NTNU Department of Music has established the ensemble T-EMP (Trondheim ensemble for Electroacoustic Music Performance). The ensemble focuses on new modes of improvisation and music making, utilizing the possibilities inherent in contemporary electroacoustic instrumentation (Figure 1). A key objective is to blur the separation between the individual contributions from each musician and collectively develop tight-woven timbral gestures.

Through live sampling and processing the very generation of sonic material may grow out of a collaborative effort where acoustic sounds (voice, drums, etc.) are processed in real-time by digital instruments. Ultimately any sound produced could serve as source material for processing by another member of the ensemble. This concept of live interprocessing has a huge potential for timbral experimentation, but there are some very challenging issues regarding performance complexity.

Figure 1: T-EMP playing live with guests from Maynooth, Ireland.

We are experimenting with processing tools built upon the open source, platform-independent computer music software Csound [2], either written in the Csound language itself or implemented as opcodes[1] extending the language [3].

So far we have concentrated our attention on two different processing techniques: Granular synthesis and convolution. We have already presented our work on particle synthesis through the Hadron synthesizer, a digital instrument that unifies all known variants of time based granular synthesis [4, 5] . In the present paper we will focus on the musical uses of convolution, as an extension of previous work by our colleague,Trond Engum [6].

2 Convolution

Convolution is a well known signal processing technique, but the theory behind it remains unknown to most musicians [7]. The convolution of two finite sequences $x(n)$ and $h(n)$ of length N is defined as:

---

[1]An opcode is a basic Csound module that either generates or modifies signals.

$$x(n)*h(n)=\sum_{k=0}^{N-1} h(k)\,x(n-k)$$

A time-domain, direct-form implementation (similar to a FIR[2] filter) will require on the order of N^2 multiplications. We typically use segments of 2 seconds length, equivalent to 88200 sample points at 44,1 kHz sampling rate, which makes the computational complexity prohibiting. A far more efficient solution is *fast convolution* using FFT and simple multiplication in the frequency domain [8]. It does however introduce latency equal to the segment length, which is undesirable for real-time applications. Partitioned convolution reduces latency by breaking up the input signal into smaller partitions. Techniques combining partitioned and direct-form convolution can eliminate processing latency entirely [9].

There are many well-known applications of convolution, such as filtering, spatialization and reverberation. Common to them is that one of the inputs is a static impulse response (characterizing a filter, an acoustic space or similar), allocated and preprocessed prior to the convolution operation. Impulse responses are typically short and/or with a pronounced amplitude decay throughout its duration. The convolution process does not normally allow parametric real-time control.

We wanted to explore convolution as a creative sound morphing tool, using the spectral and temporal qualities of one sound to filter another. This is closely related to cross-filtering [7] or cross-synthesis, although in the latter case one usually extracts the spectral envelope of one of the signals prior to their multiplication in the frequency domain [10].

Trond Engum employed similar techniques in his artistic research project where he did real-time convolution of drums and guitars with industrial sounds such as trains and angle grinders. There are a few earlier references of related uses of convolution, starting with Barry Truax [10-13].

An important aspect of our approach is that both impulse response and input should be dynamically updated during performance. This adds significant amounts of complexity, both with respect to technical implementation and practical use. Without any real-time control of the convolution process, it can be very hard to master in live performance. Depending on the amount of overlap of spectral content between the two signals, the output amplitude may vary by several orders of magnitude. Also, when both input sounds are long, significant blurring may appear in the audio output as the spectrotemporal profiles are layered.

A possible workaround is to convolve only short fragments of the input sounds at a time, multiplying them frame by frame in the frequency domain. The drawback is that any musically significant temporal structure of the input signals will be lost in the convolution output. To capture the sound's evolution over time requires longer segments, with the possible artifact of time smearing as a byproduct. This seems to be a distinguishing factor in our approach to convolution and cross-synthesis.

This paper presents some experiments that try to overcome some of the issues above. Our aim has been to:

- create dynamic parametric control over the convolution process in order to increase playability
- investigate methods to avoid or control dense and smeared output
- provide the ability to update/change the impulse responses in real-time without glitches
- provide the ability to use two live input sounds to a continuous, real-time convolution process

The motivation behind the experiments is the artistic research within the ensemble T-EMP and specific musical questions posed within that context.

3 Experiments

The experimental work has produced various digital convolution instruments, for simplicity called *convolvers,* using Csound and Cabbage[3].

The experiments can be grouped under two main headings: Dynamic updates of the impulse response, and parametric control of the convolution process.

3.1 Real-time convolution with dynamic impulse response

From our point of view processing with a static impulse response does not fully exploit the potential of convolution in live performance. We therefore wanted to investigate strategies for dynamically updating the impulse response.

[2]FIR: Finite Impulse Response

[3] Cabbage is a toolkit for making platform-independent Csound-based audio plugins [1]. See also http://www.thecabbagefoundation.org/

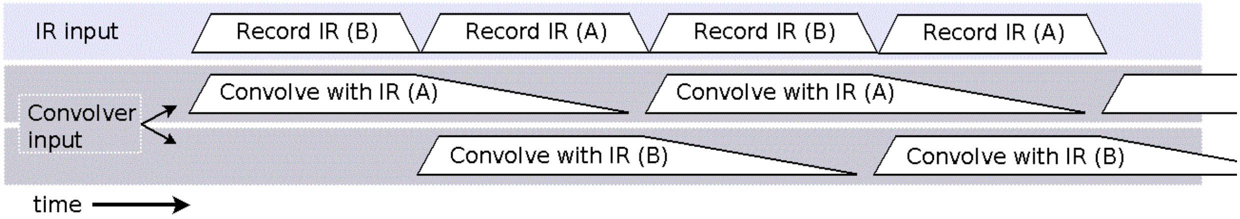

Figure 2: The stepwise updated IR buffer

A note on terminology: The term *impulse response* relates explicitly to the mathematical theory behind FIR filters or to the acoustic measurements of reverberation. Still we find it convenient to use the term (or its abbreviation IR) to signify the static input of a convolver, even when the signal no longer is the response to any impulse, strictly speaking.

3.1.1 The live sampling convolver

As a first attempt, we implemented a convolver effect where the impulse response could be recorded and replaced in real-time. This was intended for use in improvised music performance, similar to traditional live sampling, but using the live recorded audio segment as an impulse response for convolution instead. No care was taken to avoid glitches when replacing the IR in this case, but the instrument can be used as an experimental tool to explore some possibilities. Input level controls were used to manually shape the overall amplitude envelope of the sampled IR, by fading in and out of the continuous external signal. This proved to be a simple, but valuable method for controlling the timbral output.

In our use of this instrument we felt that the result was a bit too static to provide a promising basis for an improvised instrumental practice. Still, with some enhancements in the user interface, such as allowing the user to store, select and re-enable recorded impulse responses, it could be a valuable musical tool in its own right.

3.1.2 The stepwise updated IR buffer

The next step was to dynamically update the impulse response during convolution. A possible application could be to tune a reverberation IR during real-time performance. A straightforward method to accomplish this without audible artifacts is to use two concurrent convolution processes and crossfade between them when the IR needs to be modified.

When combined with live sampling convolution, the crossfade technique renders it possible to do real-time, stepwise updates of the IR, using a live signal as input. In this manner the IR is updated and replaced without glitches, always representing a recent image of the input sound.

Figure 2 illustrates the concept: Impulse responses are recorded in alternating buffers A and B from one of the inputs. Typical buffer length is between 0.5 and 4 seconds with 2 seconds as the most common. An envelope function is applied to the recorded segments for smoother convolution. The second input is routed to two parallel processing threads where it is convolved with buffer A and B respectively. The convolution with buffer A fades in as soon as that buffer is done recording. Simultaneously the tail of convolution with buffer B is faded out and that buffer starts to record.

This allows us to use two live input signals to the convolution process. There is however an inherent delay given by the buffer length. Future research will explore partitioned IR buffer updates to reduce the delay to the length of a single FFT frame.

3.2 Parametric control of the convolution process

Convolution can be very hard to control even for a knowledgeable and experienced user [7]. A fundamental goal for our experiments has been to open up convolution by providing enhanced parametric control for real-time exploration of the technique.

3.2.1 Signal preprocessing

As we have noted, the convolution process relates all samples of the IR to all samples of the input sound. This can easily result in a densely layered, muddy and spectrally unbalanced output. Various forms for preprocessing of the input signals has been proposed, such as high-pass filtering [14], compression/expansion and square-root scaling [10].

We furnished our convolver with user-controlled filtering (high-pass and low-pass) on both convolution inputs, as this can reduce the problem of dense, muddy output considerably.

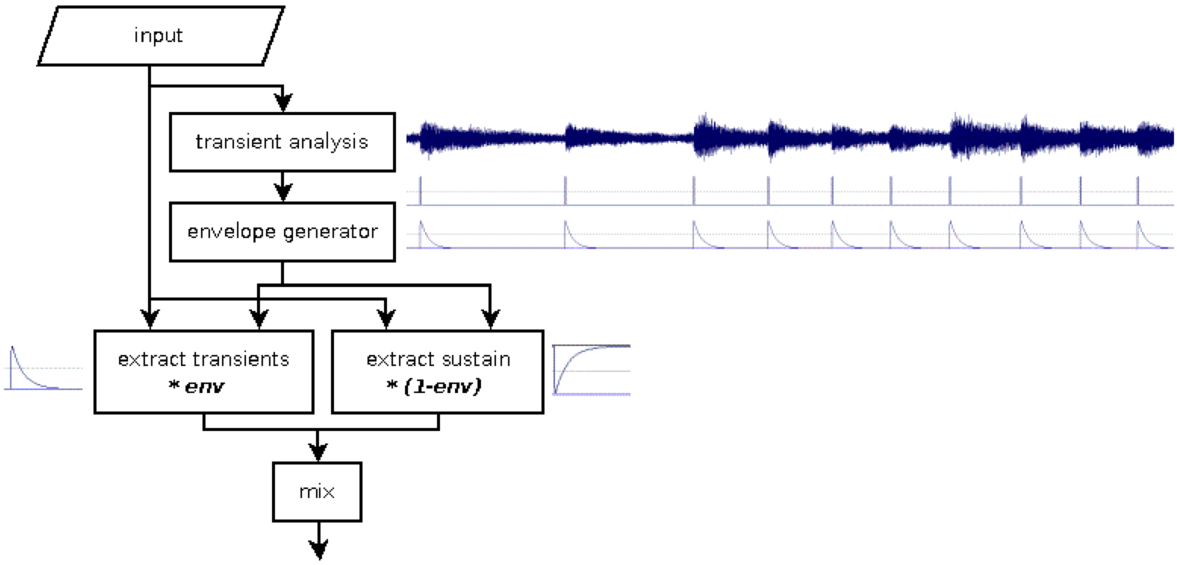

Figure 3: Splitting transient and sustained parts of an input signal

The point is to provide dynamic control over the “degree of spectral intersection” [10].

3.2.2 The transient convolver

As another strategy for controlling the spectral density of the output, while still keeping with the basic premise that we want to preserve the temporal characteristics of the IR sound, we split both the IR and the input sound into transient and sustained parts. Convolving with an IR that contains only transients will produce a considerably less dense result, while still preserving the large-scale spectral and temporal evolution of the IR.

The splitting of the transient and sustained parts was done in the time domain by detecting transients and generating a transient-triggered envelope (see Figure 3). The transient analysis can be tuned by a number of user-controlled parameters. The sustained part was extracted by using the inverted transient envelope. Hence the sum of the transient and sustained parts are equal to the original input.

Transient splitting enables parametric control over the density of the convolution output, allowing the user to mix in as much sustained material as needed. It is also possible to convolve with an IR that has all transients removed, providing a very lush and broad variant.

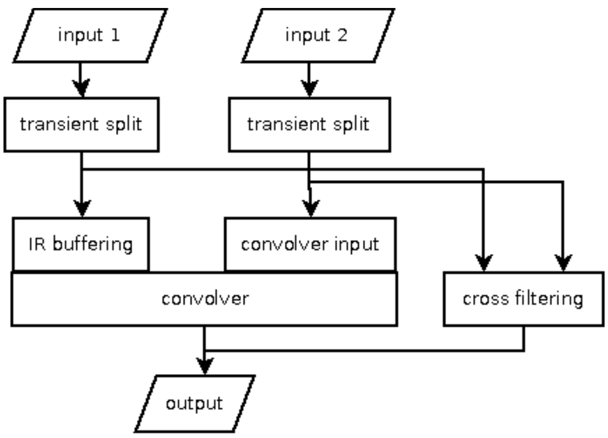

Figure 4: Convolver with transient split and stepwise update IR buffer. Straightforward cross-filtering is added as reference.

3.3 Combining the results

Finally, the various convolution experiments outlined above were combined into a single convolver[4]. It works with two live audio inputs: one is buffered as a stepwise updated IR and the other used as convolution input (see Figure 4).

The IR can be automatically updated at regular intervals. The sampling and replacement of the IR can also be relegated to manual control, as a way of “holding on” to material that the performer finds musically interesting or particularly effective.

Each of the two input signals can be split into transient and sustained parts, and simple low-pass and high-pass filters are provided as rudimentary methods for controlling spectral spread.

Straightforward cross-filtering, continuously multiplying the spectral profile of the two inputs frame by frame, was also added to enable direct comparison and further experimentation. As should be evident from Figure 5 the user interface

[4] The Csound code for this convolver is available at http://folk.ntnu.no/oyvinbra/LAC2013/, ready to be compiled into a VST using Cabbage.

Figure 5: The convolver user-interface

provides a great deal of parametric control of this convolver.

4 Conclusion and further work

We have implemented a number of variations of convolution as an attempt to overcome limitations inherent in convolution as a music processing technique. The context for our experimental work is musical objectives growing out of improvisational practice in electroacoustic music. Preliminary tests show that some of the limitations have been lifted by giving real-time parametric control over the convolution process and the density of its output, and by allowing real-time updates of the IR.

In practical use, the effect is still hard to control. This relates to the fact that, with real-time stepwise updates of the IR, the performer does not have detailed control over the IR buffer content. The IR may contain rhythmic material that are offset in time, creating offbeat effects or other irregular rhythmic behavior. With automatic IR updates the performer does not have direct and precise control over the timing of IR changes. Instead the sound of the instrument will change at regular intervals, not necessarily at musically relevant instants.

A possible way of controlling rhythmic consistency would be to update the IR in synchrony with the tempo of the input material, for instance so that the IR always consists of a whole measure or beat and that it is replaced only on beat boundaries. Another proposal would be to strip off non-transient material at the start of the IR, so that the IR would always start with a transient. This is ongoing work.

We hope to present our convolver live at the conference.

5 Acknowledgements

Our thanks goes to the performers of T-EMP who in addition to author Øyvind Brandtsegg are: Tone Åse, Ingrid Lode, Carl Haakon Waadeland, Bernt Isak Wærstad and in particular Trond Engum, who has advocated creative uses of convolution for many years now. We would also like to thank the Norwegian Artistic Research Programme for supporting T-EMP as a performing laboratory.

References

[1] R. Walsh. 2011. Audio Plugin development with Cabbage. Proceedings of Linux Audio Conference, pages 47-53.

[2] Csound. See http://www.csounds.com/.

[3] V. Lazzarini. 2005. Extensions to the Csound Language: from User-Defined to Plugin Opcodes and Beyond. LAC2005 Proceedings, pages 13.

[4] Ø. Brandtsegg, S. Saue, T. Johansen. 2011. Particle synthesis–a unified model for granular synthesis. Proceedings of the 2011 Linux Audio Conference(LAC'11).

[5] Ø. Brandtsegg, S. Saue. 2011. Performing and composing with the Hadron Particle Synthesizer. Forum Acusticum, Aalborg, Denmark.

[6] T. Engum. Real-time control and creative convolution. Proceedings of the International Conference on New Interfaces for Musical Expression}, pages 519-22.

[7] C. Roads. 1997. Sound transformation by convolution. In C. Roads, A. Piccialli, G. D. Poli, S. T. Pope, editors, *Musical signal processing*, pages 411-38. Swets & Zeitlinger.

[8] R. Boulanger, V. Lazzarini. 2010. *The Audio Programming Book*, The MIT Press.

[9] W. G. Gardner. 1995. Efficient Convolution without Input-Output Delay. *J Audio Eng Soc*, 43(3), pages 127.

[10] Z. Settel, C. Lippe. 1995. Real-time musical applications using frequency domain signal processing. Applications of Signal Processing to Audio and Acoustics, 1995, IEEE ASSP Workshop on, pages 230-3 IEEE.
[11] B. Truax. 2005. Music and science meet at the micro level: Time-frequency methods and granular synthesis. *Acoustical Society of America Journal*, 117pages 2415-6.
[12] R. Aimi. 2007. Percussion instruments using realtime convolution: Physical controllers. Proceedings of the 7th international conference on New interfaces for musical expression, pages 154-9 ACM.
[13] D. Merrill, H. Raffle, R. Aimi. 2008. The sound of touch: physical manipulation of digital sound. Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems, pages 739-42 ACM.
[14] T. Erbe. 1997. SoundHack: A Brief Overview. *Computer Music Journal*, 21(1), pages 35-8.