

AUDIO SIGNAL DRIVEN SOUND SYNTHESIS

Cornelius Poepel

Academy of Media Arts
Peter-Welter-Platz 2
D-50676 Koeln
cp@khm.de

Roger B. Dannenberg

Carnegie Mellon University
School of Computer Science
Pittsburgh, PA 15213 USA
rbd@cs.cmu.edu

ABSTRACT

A new approach to computer music instruments is described. Rather than sense control parameters from acoustic instruments (or non-acoustic instrument controllers), the sound of an acoustic instrument is used directly by a synthesis algorithm, usually replacing an oscillator. Parameters such as amplitude and pitch can control other aspects of the synthesis. This approach gives the player more control over details of the sound due to the use of the rich acoustic signal. Latency in sensing parameters, particularly pitch, is less of a problem because pitch information is carried directly by the acoustically generated signal. Several examples are described and the results of a subjective evaluation by musicians are presented.

1. INTRODUCTION

Researchers often begin with a new synthesis method and try to build an appropriate control interface. This has the drawback that musicians must learn a new instrument in order to obtain new sounds. Alternatively, research can begin with a conventional instrument and focus on sensors that enable the instrument to control a synthesis method in real time. This has the drawback that sensors, including pitch detectors, are often slow, make errors, and transmit only a subset of the musician's actions that are meaningful in sound production. The control information obtained from an acoustic instrument may not be suitable for any synthesis algorithm. A contrasting approach, described by Poepel [7], is based on the idea of modifying existing synthesis methods to be playable with a traditional musical instrument. This approach uses the instrument itself as the "oscillator" and eliminates the need for very low-latency sensing. It is capable of timbral nuances and articulations that are not tracked by a sensing system.

This new approach can be used to adapt or modify various synthesis methods. The main principle is that the central oscillator in the synthesis method is replaced by the direct audio signal of the acoustic instrument. The raw and unanalyzed audio signal provides the primary signal for the synthesis engine. Additionally this signal is analyzed to obtain parameters such as pitch (fundamental frequency) and envelope. The results are mapped to modify the synthesized sound indirectly. Figure 1

illustrates the configuration for what we call "Audio Signal Driven Sound Synthesis."

After describing related work, we describe two simple algorithms based on FM and subtractive synthesis. This leads to an oscillator-free (and pitch-estimation free) version of FM we call *self-modulation*. We then present findings from string players who tested our implementation. In Section 6, we compare and contrast our approach with more traditional computer music instrument building.

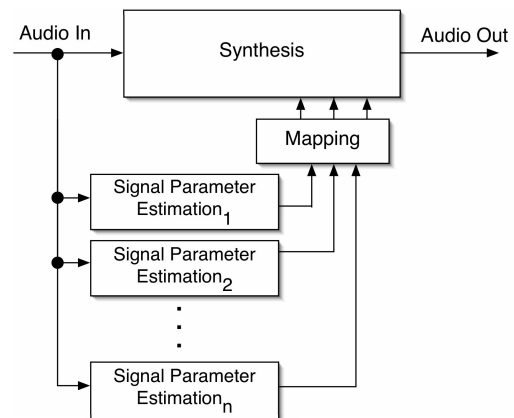


Figure 1. Schematic of audio driven sound synthesis. The audio input is used directly within the synthesis process. Additional parameters may be obtained from the signal and used to provide additional and indirect control.

2. RELATED WORK

Audio signal driven sound synthesis is not without precedents. The Mutron [3], an effect box introduced to the market in 1972, is an early example of the approach that we are advocating here. The Mutron combines an envelope follower with a variable resonant filter. Thus, all the performance nuances incorporated into the input audio are more-or-less transferred to the output, but in addition, parameters are extracted from the audio (in this case, just the envelope) and used to control some synthesis processing (in this case, just a filter), giving the performer a new form of control over partially electronic, synthetic sound production.

Similar to the Mutron, a version of the electronic violin of Max Mathews used the envelope follower to control filters [5, 6]. While first versions of the instrument worked with 17 to 37 fixed resonant filters [4], a later version used envelope follower modulated

lowpass filters. The more intensive the audio signal was, the bigger was the bandwidth of the filters. The resulting sound had some of the character of brass instruments. Another version used a bandpass filter. The center frequency was moved up and down according to the amplitude of the signal. This version gave sound results somewhat like a human voice.

As mentioned earlier, many have derived control information from acoustic instruments' audio output and used this to control synthesis algorithms, for example Furukawa and Dutilleux [2], and others have built controllers to sense physical gestures, for example Solvi and Thierry [11], but our approach differs in that it incorporates the acoustic audio signal directly. Our work can be considered a subset of the class of adaptive effects [12], which are digital audio effects that are controlled by parameters extracted from the input signal.

3. EXAMPLES FROM PREVIOUS WORK

Two implementations of this method were presented by Poepel [7]. The first one is based on a modified simple FM synthesis and the second one is based on a modified subtractive synthesis.

3.1. Modified FM Synthesis

To modify simple FM synthesis, the modulator oscillator is replaced by the input audio signal. To control the volume of the carrier oscillator, the envelope follower signal is mapped to control the carrier amplitude. To avoid unnatural brightness in the resulting sound on strong input, the signal of the envelope follower is mapped to control the index. The Carrier : Modulator frequency ratio is maintained using a pitch tracker on the input audio signal. Figure 2 shows the resulting algorithm.

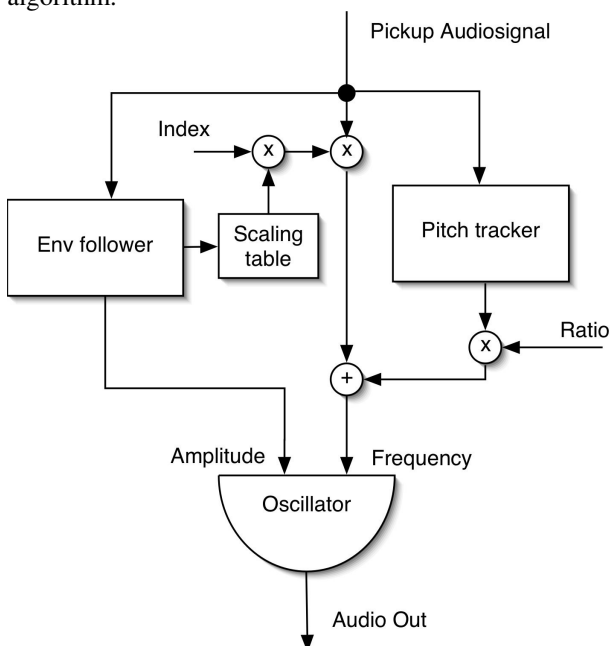


Figure 2. Schematic of audio signal driven FM synthesis.

3.2. Modified Subtractive Synthesis

In subtractive synthesis, the basic idea is to modify the spectrum by filtering a signal. The usual oscillator placed before the filters is replaced by the input audio signal. Bandpass filters are used which can be assigned to any of the partials of a tone. The pitch tracker controls the center frequencies of these filters. Bandwidth and volume of each filtered harmonic can be set. The complete flowchart is shown in Figure 3.

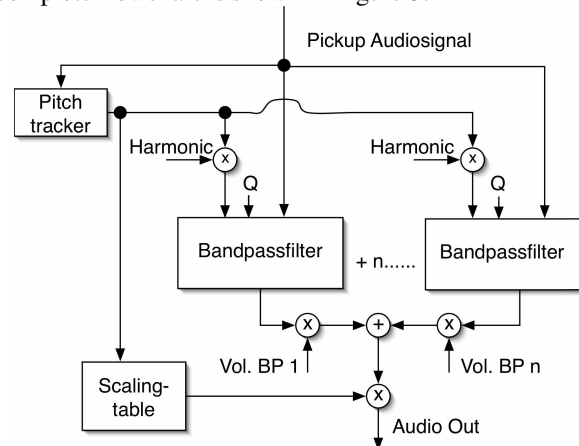


Figure 3. Schematic of audio signal driven subtractive synthesis

The advantages and drawbacks can be summarized as follows:

- Nuances in sound applied by the player at the input of the system are represented in the output signal even if they are not tracked by the envelope follower and pitch detector. Therefore, the method offers a higher instrument-specific playability than do common synthesis methods.
- Latency is reduced because sound is already perceived if an input signal is applied. The latency of the pitch tracker will lead to a time-varying timbre in the output. However, this is a minor problem in comparison with having no sound at all until the pitch detector sends data.
- The big drawback is the non-compatibility with common interfaces and existing synthesis methods. MIDI synthesizers or OSC-synthesizers (Open Sound Control) cannot be driven using this new approach. It is necessary to modify existing synthesis methods in order to use them as described here.

This paper builds upon this earlier work. We suggest a new "self-modulation" synthesis method and describe experience and new implementations using audio signal driven sound synthesis.

4. SELF MODULATION

Figure 2 shows a synthesis method based on FM. One possible drawback of this method is that it requires pitch estimation in order to set the carrier frequency. An even simpler approach uses the input audio in place of both the carrier *and* the modulator oscillators, as shown in

Figure 4. As with ordinary FM, this might be more properly called *phase modulation*. The incoming signal enters a variable delay which is modulated by the signal itself. The output signal is given by:

$$y(t) = x(t - b - x(t)) \quad (1)$$

where b is a constant to bias the phase to be strictly less than t (because $x(t)$ may be negative). Otherwise, the system would not be causal.

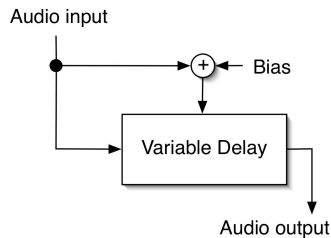


Figure 4. Schematic of audio signal driven FM using self modulation.

This method was implemented and tested using the second author’s trumpet as an input source. It was immediately apparent that the resulting sound tended to be very bright, as one might expect from FM when both the carrier and modulator are harmonically rich. To reduce the brightness, at least at lower amplitudes, a low-pass filter was added to the audio input. Also, an attenuation control for the modulator is essential. The resulting system is shown in Figure 5.

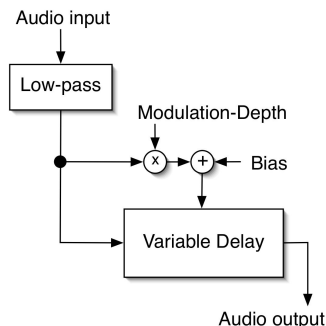


Figure 5. Complete self-modulating FM.

This system was implemented in Aura [1] to allow real-time, interactive experimentation. As one would expect, the resulting sounds convey the amplitude and pitch of the original trumpet audio quite transparently. On the other hand, the low-pass filter (12dB/octave) severely deadens the tone at low amplitudes, and the phase modulation generates FM-like spectral evolution as the trumpet gets louder. This gives the instrument a very familiar FM quality.

Various extensions to this synthesis method are possible. While the Carrier : Modulator frequency ratio is fixed at 1:1, non-linear processes could be used to change the modulator or carrier. For example, squaring a signal (either the carrier, modulator, or both) will double the frequency of a sinusoidal input. The effect is more complicated with real input that is non-sinusoidal.

Additional modulation sources can be added, including noise and fixed frequency oscillators. Noise

can broaden the spectrum, adding roughness to the sound, and fixed frequency oscillators can lead to inharmonic partials. Additional modulators can themselves be modulated by the amplitude envelope and even frequency of the audio input with the addition of an envelope follower and/or pitch detector.

5. APPLICATIONS

5.1. Synthesized Strings

The instrumental implementation described by Poepel [7] uses a Harms silent Viola and a Shadow SH 941 pickup. Figure 6 presents the hardware necessary for an application. Software is implemented using MaxMSP. The algorithms are used as described in Section 2. Pitch and amplitude tracking is done by using the MSP-object `fiddle~` by Miller Puckette [10]. To verify usability on different systems, a traditional viola equipped with a Zeta pickup and a Yamaha silent violin were tested.



Figure 6. Hardware used in the application.

5.2. Sound Examples and Players Experiences

Sound examples may be found on the following website: <http://www.khm.de/~cp>.

To get a broader view of the playability of instruments using these methods, the instrument was given to string players for testing. Traditional simple FM Synthesis was implemented and players were asked to compare traditional FM Synthesis with modified FM Synthesis [8]. The new method was estimated to be more accurate in dynamics, in vibrato, and in articulation, presumably because the new method can react to slight differences in timbre. Players also felt the new method feels more like a string instrument, but did not feel the new method sounds better. It could be that players judged the sound based on expectations of a clean electronic sound or other preconceptions of synthesized music.

6. DISCUSSION AND FUTURE WORK

6.1. Measurement and Players Focus

The traditional approach to controlling sound synthesis

may be seen as related to formal Artificial Life approaches that have been criticized for the semantic gap between simulation and measurement [9]. Prem states that measurement is a process in which two dynamical systems interact with each other. The interaction between a musician and the instrument can be interpreted as a mapping of a complex situation to a sound. However this view often disregards the dynamic nature of the measuring process itself.

Where do we find this problem in practical interface use and development? In the case of string instruments, common interfaces use measuring systems to observe at least pitch and amplitude. The problem is that such a system shifts the player's focus from the string to the tracking system. In order to get the notes the player wants to get, he or she has to create an input to the interface, enabling the measurement system to send the data necessary to perform the sound the player wants to perform. This often causes the player to feel he or she is not playing a string instrument anymore, but instead playing a tracking system via the string-instrument-based interface.

Since the present method enables an instrument builder to create interfaces using both the audio signal and measured parameter data to control the synthesis engine, the dynamic nature of the measuring process may influence the players focus in a different, probably less disturbing way.

6.2. Construction-Conditioned Features

It is obvious that the sound of an instrument using audio driven sound synthesis will be crucially influenced by the selected synthesis method. One can expect that specific synthesis methods will interfere with input more or less depending on the method of playing and the nature of the synthesis algorithm. A synthesis method producing noisy sounds will cover up input that depends on small amounts of noise such as subtle articulation effects. A synthesis method that generates new modulation partials may remove the player's control over brightness.

It is also obvious that the indirect modification of audio signal controlled sounds will have a bigger or smaller influence on the sound output, depending on how indirectly those parameters affect the sound result. According to our experiences, those disturbances do not depend only on the specific synthesis method used; they also depend on the musical structure that is performed. The method described in Section 5, based on a modified subtractive synthesis for example, shows different "plopping artifacts" depending on the size of intervals being played. While the center frequencies of filters are controlled by the pitch tracker, the plops get stronger with larger changes in frequency. For example, a trill (performed on one string) causes no problem at all, but a legato using octaves can hardly be performed without disturbing plops.

7. SUMMARY AND CONCLUSION

In summary, we have described a basic method that offers a new set of advantages and drawbacks. By incorporating acoustic instrument signals directly into the synthesis method, we can capture more of the performer's subtle control. The causal relationships in this method are different from what is found in common synthesis methods like FM synthesis or subtractive synthesis. Our initial investigations already show satisfying degrees of playability and sound quality. We hope this approach will inspire new synthesis and signal processing techniques that introduce new classes of sounds but retain much of the controlling acoustically generated signal.

8. REFERENCES

- [1] Dannenberg, R. B. "Combining Visual and Textual Representations for Flexible Interactive Audio Signal Processing", *Proceedings of ICMC 2004: The 30th Annual International Computer Music Conference*. Miami, USA, 2004, ICMA, San Francisco, 240-247.
- [2] Furukawa, K. and Dutilleux, P., Live-Electronics Algorithms in the Multimedia Work "Swim Swan", *Proceedings of the 5th International Conference on Digital Audio Effects (DAFx-02)*, Hamburg, Germany, 2002, 263-268.
- [3] Gill, C. "The Stomping Ground: Musitronics, Mu-Tron, and the Gizmotron", *Vintage Guitar Magazine* Sept. 1997, 11 (12), Publ. Alan & Cleo Greenwood, Bismarck, ND, 1997
- [4] Mathews M. V. and Kohut J. "Electronic simulation of violin resonances", *Journal of the Acoustical Society of America*, 53(6), 1973, 1620-1626.
- [5] Moog, R. Personal conversation, 2004.
- [6] Pierce, J. R. *Klang mit den Ohren der Physik* german translation of the book *The science of musical sound* Second Edition, Heidelberg, Spektrum-der-Wiss.-Verlagsgesellschaft, 1989.
- [7] Poepel, C. "Synthesized Strings for Stringplayers", *Proceedings of the 2004 Conference on New Interfaces for Musical Expression*, Hamamatsu, Japan, 2004, 150-153.
- [8] Poepel, C. "On Interface Expressivity: A Player-Based Study", *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, Vancouver, Canada, 2005, 228-231.
- [9] Prem, E. "Epistemic Autonomy in Models of Living Systems", Husbands P. and Harvey I.(eds.), *Fourth European Conference on Artificial Life (ECAL97)*, A Bradford Book, MIT Press, Cambridge, MA, 1997, 2-9.
- [10] Puckette, M., Apel, T. and Zicarelli, D. "Real-time audio analysis tools for Pd and MSP", *Proceedings of the 1998 International Computer Music Conference*, Ann Arbor, USA, 1998, ICMA, 109-112.
- [11] Solvi, Y. and Thierry, V. "Design of a Flute Interface to Control Synthesis Models", *Proceedings of the 1999 International Computer Music Conference*, Beijing, China, 1999, ICMA, 464-467.
- [12] Verfaillie, V. and Depalle, P., "Adaptive Effects Based on STFT, Using a Source-Filter Model", *Proceedings of the 7th International Conference on Digital Audio Effects (DAFx'04)*, Naples, Italy, 2004, 296-301.