Locating and Utilising Inherent Qualities in an Expanded Sound Palette for Solo Flute

Alice Bennett
Academic Studies Faculty, Australian Institute of Music, Australia
alice@alicebennett.net
http://www.alicebennett.net

Vincent Giles
Composition Faculty, Australian Institute of Music, Australia
PhD candidate, Victorian College of the Arts, University of Melbourne, Australia
vin@vgiles.net
http://www.vgiles.net

Proceedings of Korean Electro-Acoustic Music Society's 2015 Annual Conference (KEAMSAC2015)
Seoul, Korea, 2-3 October 2015

In the search for an idiosyncratic improvisatory language of solo flute performance, it is sometimes necessary to move beyond the scope of technique covered by traditional and extended techniques into the world of instrument-extension through computers. To this end, Bennett's creative work has led to exploration of new electroacoustic techniques, searching for ways to expand the available sonic palette.

This presentation will demonstrate an electronically-extended flute performance utilising Giles' Spectral Domain Microsound Amplification Software (SDMAS) in real-time. The SDMAS amplifies soft sounds relative to loud, by real-time input, shifting the partials by amplitude around an amplitude-based pivot point. The result is that these otherwise-inaudible partials are brought up to be audible alongside the higher amplitude partials, which drastically alters the perceived timbre of the instrument or instruments being treated. This allows the performer/composer to not only discover, but exploit a greater range of timbres than would be available by an acoustic instrument. Some examples relating to flute are the amplification of soft, closed-hole and whistle techniques, as well as expanding the soft partials that form integral but otherwise imperceptible parts in loud techniques such as jet-whistles and traditional technique.

The authors will discuss the sonically enriched results and how it has led to the development of new performative works and an idiosyncratic improvisatory language based on this hyper-instrument configuration.

Uncovering additional sonic worlds that are inherent in acoustic instrument sound production has been the goal of composers and performers for decades now, possibly as far back as Schoenberg's *klangfarbenmelodie* (Schoenberg 1978) and certainly being well explored and documented by Lachenmann with his *musique concrete instrumentale* (Lachenmann 2004). It has been a feature in the jazz and improvisational world too for quite some time, particularly coming out of the tradition of free jazz, with composer-performers like Ornette Coleman (Coleman et al. 1998) and Derek Bailey (Bailey 1993) exploring the expanded sound palettes offered through extended technique on their instruments.

In parallel to these activities, computer music has facilitated the synthesis, analysis, and resynthesis of sounds that, 120 years ago, were barely dreamed about. Composers such as Stockhausen (Stockhausen and Stockhausen 1968), Berio (Berio 1971), Boulez (Boulez 2010), and many others have sought to capture the idiosyncrasies of electronics and electronic processing, at times augmenting an acoustic instrument through electronic means, or a whole ensemble through spatialization.

Now in the popular musical world, effects pedals (electronic augmentation) are a common feature, and digital audio workstations (DAWs) allow for sound manipulation that once took massive effort when working with tape to be done quickly and with far less effort. So we come to somewhat of an impasse: the extraordinary expansion of acoustic sound production using instruments throughout much of the latter half of the 20th century and into the 21st century, and computing power and sonic exploration that continues to uncover new methods. The realm of microsound explored by Roads (Roads 2001) and others continues to be of interest to many composers and performers. Roads' microsound is expanded to include the amplitude domain (Giles 2013-2016). The goal of this paper is to explain and explore a novel approach to microsound and instrument augmentation using instruments, merging the practice of composer Vincent Giles and flutist/composer Alice Bennett. Giles' practice encompasses computer music, notated instrumental music, and installation work. His programming has come about by necessity, rather than from a background in programming, and so gives rise to idiosyncrasies in technique and realisation that a coder may not come across. Bennett is a classically trained flutist who specialises in contemporary repertoire, including an improvisational practice based on the electronic augmentation of the bass and concert flutes.

Spectral Domain Microsound Amplification Software (SDMAS)

The purpose of this software is to allow for the FFT-driven amplification of amplitude-based microsound in both real-time (RT) and non-realtime (NRT) situations, resulting in the ability to 'peer within' a sound's spectral content for those qualities of sound that exist at the edges of our perception, but nonetheless have a profound impact on our understanding of sound.

The result of this amplification process can be analytical, as was first intended, or as a device for real-time augmentation of an acoustic instrument that allows those spectral sounds that are otherwise inaudible in the performance of an instrument to become audible. Spectral-domain processing has an important distinction when compared to time-domain processes (such as compression and expansion) that may or may not achieve similar results, and that is that time-domain processing is frequency-agnostic, dealing only with the sum of frequencies and amplitudes at any given sample point[1]. This frequency-agnosticism makes time-domain processing apply only an increase or decrease in amplitude of the summed frequency and amplitude data, rather than specific frequency or amplitude-based manipulation, which is only possible in the spectral domain.

The remainder of this section of the paper will describe each process within the SDMAS, detailing the algorithms and processing that are employed, and finally considerations for future development and refinement.

Stages of Signal Processing

Figure 1 (below) shows the flow of signal and derived data throughout the patch. This can be expressed in text in the following way:

Audio: Input (RT or NRT) --- Filter 1 --- FFT --- Filter 2 --- Output Stage
--- Output Stage

Data: Input --- Frequency and Amplitude Approximation --- Filters
--- FFT Control

Figure 1: Data-flow text

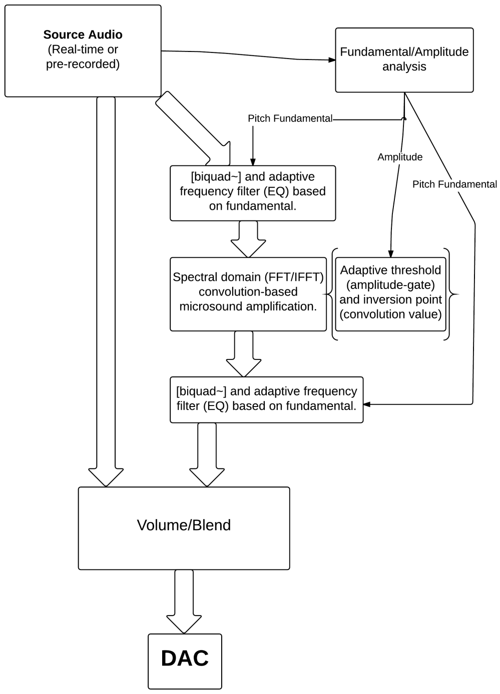

Figure 2: Signal and data-flow diagram of SDMAS.

The flow of signal and data corresponds in part to the layout of the interface for the patch. What follows is a description of the role of each interface component, with sub-sections for technical information, focusing on the algorithms inside the FFT sub-patch. The inner workings of the SDMAS are rudimentary, and the mechanism for amplifying microsound is equally simplistic. The most complex component is actually restraining the amplification system to avoid all of the background white noise inherent in any audio signal, whether RT or NRT.

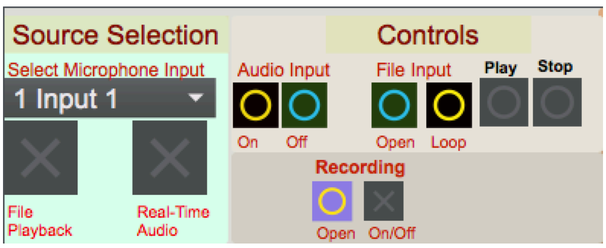

Figure 3: Source selection and playback controls

Figure 3 (above) correlates to the top left box in Figure 1, and has the following functions.

Source Selection

Within this sub-section a user can select the input from available inputs on their audio interface, and route either the file playback or incoming audio through the SDMAS system.

Controls

The audio input buttons switch the [adc~] (hardware input) object on and off, while the file input options include open: open an audio file; loop: loop it; play & stop. As of version 0.9, this is rudimentary playback from disk, not from a buffer. The recording stage just allows for the recording of SDMAS output to disk for later use.

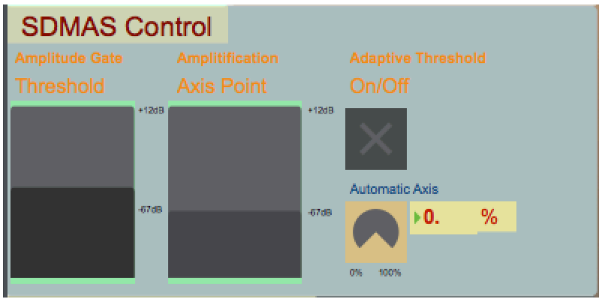

Figure 4: SDMAS controls

The primary user input controls for the FFT-based component of the software consist of a manually controllable amplitude-based noise gate, an axis point (both in decibels ranging from -90dB to ~+12dB), and control for automated threshold and axis point generation based on the incoming signal.

The amplitude gate threshold slider sends the output to the FFT sub-patch as an amplitude value between 0 and 1, all sound below that amplitude will be discarded from the signal path. The axis point needs to be set somewhere above the threshold, because there will be nothing to act on if it is below the value set by the threshold. To make this easier, the automatic axis dial allows the user to set a percentage above the threshold for the axis point to be automatically set to. The adaptive threshold toggle simple takes the peak amplitude of the incoming signal and uses that to modulate the threshold slider.

Inside the FFT

Threshold and axis point functions within the FFT can be described with the following pseudo-code statements:

IF (amplitude[a] < threshold[t]) { discard } **ELSE** { pass } **THEN FOR** (a > t) { 1/a }

This process is simply a means of removing excess sonic information (background noise) and inverting the frequency and amplitude information around a given point, using *1/x*. The result is that around any given axis point, all amplitudes will be inverted with soft becoming loud, loud becoming soft. However, the frequencies to which those amplitudes belong *do not change*, meaning that the frequencies themselves become more audible.

Finally, this inverted signal is sent to a convolution[2] system to reinforce dominant frequencies. Figure 5 (**Error! Reference source not found.**) shows the patching for this process within a [pfft~] sub-patch, running at 4096 window size, using a hamming windowing function, with an overlap factor of 8.

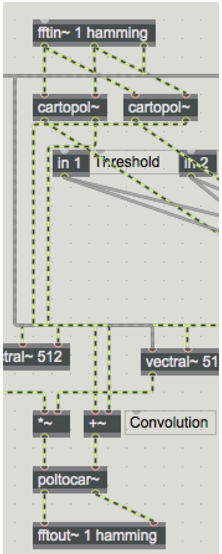

Figure 5: FFT-based convolution

Filtering

Filtering occurs in two stages: before the [pfft~] sub-patch, and after it. It is the same filter on both sides, and can be controlled by fundamental frequency estimation automatically, or controlled manually. The default filter is a bandpass, which results in removing trimming the excess sound going into and out of the FFT. Wider bandwidth (Q) allow for greater upper partials, however, narrow bandwidth allows for a tightly controlled emphasis on certain frequencies that may be of interest. This is particularly useful when the user wishes to enhance subtones or very high frequency overtones, and having the filter before and after the FFT cleans the sound up substantially. Figure 6 (below) shows the filter graph and other controls for this component of the SDMAS.

In the top-right corner of Figure 6 is the threshold control for the [fzero~] object which estimates the fundamental and peak amplitude of a given analysis window; the threshold control allows the user to specify the lowest amplitude above which new pitches and amplitudes will be reported. The second option in the top-right corner allows a user to change the type of filter being employed.

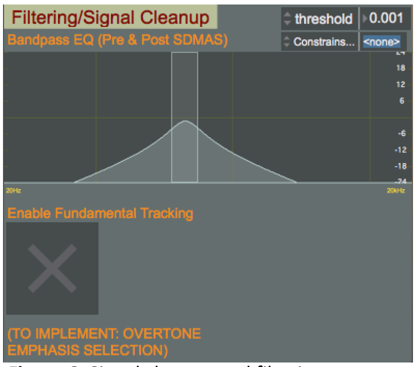

Figure 6: Signal cleanup and filtering

Output and real-time analysis

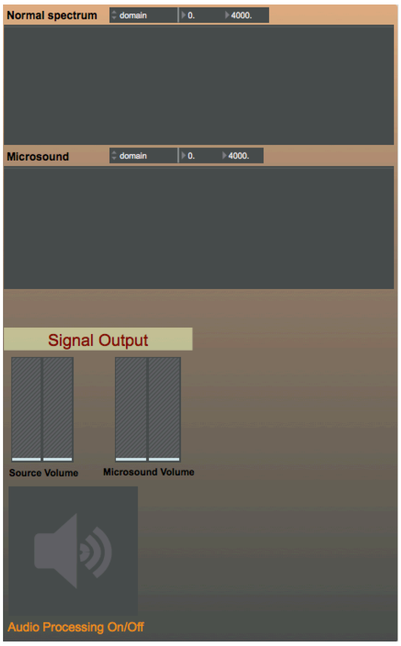

Figure 7: Spectral analysis and signal mixing

Figure 7 (**Error! Reference source not found.**) gives the user a real-time spectral analysis in the form of a spectrogram of the source audio and the microsonic components that have been enhanced through the use of the SDMAS patch. The final stage is the mixing between source audio and microsound, allowing the user to blend the two signals to their needs.

Considerations and Future Development

Real-time processing, as described elsewhere in this paper, can be problematic even though the results are extraordinarily sonically interesting. Nonetheless, the system works to highlight aspects of the amplitude-domain in an incoming signal that are otherwise inaudible, contributing to a sense of a hyper-instrument that is not augmented through additional registers or new synthetic sounds, but takes advantage of the inherent sound features of the instrument and combines those with the conventional sound, creating the hyper-instrument.

Non real-time processing does not have the control problems that plague the RT context, so the system is being deployed as part of an analysis methodology to uncover microsonic qualities within works already composed for acoustic instruments, along with acousmatic sounds. The system provides a novel approach to generating new material in an acousmatic setting while also providing a functional tool for analysis and performance.

As of the end of August 2015, the project files have been placed on GitHub[3] and Giles invites contributions to the code. In the future, the project could be re-written using SuperCollider or similar text-based programming environment to allow more precise control over the algorithms and more efficient use of processing power.

SDMAS and acoustic instruments: why?

This section of the paper will deal with the use of the SDMAS, particularly in relation to the Bennett's artistic practice: improvising and composing with solo bass and concert flutes. In particular it examines how the SDMAS can be used from a creative perspective to augment the inherent sounds produced by these instruments in an improvisatory language and in the composition *Wilf's Lament* (Bennett 2014).

Amplification of flutes, especially softer techniques such as whistle tones and closed-hole technique, is widely used in contemporary performance practice, in notated art-music, jazz, and other contemporary improvisation (post-classical). Simple amplification can be thought of as using a microphone and loudspeaker, analogous to an electric guitar or singer, and allows these techniques to be heard more easily in situations such as: concert hall ensemble performance, open-air outdoor venues, and performances with larger audiences which without amplification would only be audible in intimate concert settings with solo instruments. Examples of this type of amplification include: Paul Méfano's *Traits Suspendus* for amplified contrabass flute (Méfano 1980), Matthias Ziegler's album *Uakti* (Ziegler 1999) featuring amplified low flutes, and Mary Finsterer's *Ether* for solo flute (Finsterer 2001) featuring amplified whistle tones. Jean Penny (2009) discusses the effect of amplification on flute techniques such as whistle tones and breath noises, which she refers to as microsounds, and how these microsounds extend the creative possibilities of the flute. The sounds under discussion in this paper are quieter than those of whistle tones, and are intrinsically linked to the production of sound on a flute, for example: partials, mechanical sounds, and mouth noises that usually would only be audible by the performer. Therefore the purpose of the SDMAS in this performance and compositional context is to go further than simple amplification by exposing this microscopic sound world inherent in the physical action of producing sound with a flute.

Discovering the SDMAS and its uses for augmentation

From a performance and improvisation perspective, it is important to understand how a system (or instrument) works in order to use and exploit it. The SDMAS takes an incoming signal and inverts the amplitude (see above for a detailed description of this process). The settings and their functions within the SDMAS are discussed in the section above, but to understand the idiosyncrasies of this augmented performance system, it is important to understand that these variables have a profound effect on the outcome and must be taken into account before performance. What follows, therefore, is an account of a performer's interaction with the software.

When learning a new software interface it seems prudent to just experiment. A good place to start with the SDMAS is to turn the device on and observe the decibel and frequency level of the background noise in the room, providing a baseline from which to approximate settings that may be useful. Next, it is best with the SDMAS to test the various techniques you would like to use, in order to again establish a baseline from which to operate. While it is possible to do this in real-time, it is helpful to instead use pre-recorded examples to avoid having to constantly stop playing to make notes and try new settings. Instead, the settings can be manipulated in real-time with an immediate sonic result. This process of exploration allows, with some practice, for a performer to intimately understand the system and attempt to counteract and work with idiosyncratic behaviour of the software itself, resulting in an idiosyncratic language that exploits the inherent qualities of the flute.

Describing the relationship between flute technique and SDMAS augmentation

Each technique on the flute produces different results with the SDMAS and may require changes in settings. This section discusses Bennett's explorations of flute technique and how the SDMAS modifies that signal, and what settings within the patch were most functional.

Whistle tones: the SDMAS clearly enhances the fundamental frequency and the overtones of this technique along with any breath or mechanical sounds that are incidentally produced. The effect is similar to the simple amplification of whistle tones discussed above, but stronger and richer, with a greater range of frequencies added to the sound through amplification. The most effective settings for whistle tones are:

- Threshold: -24dB;
- *axis point*: 25% above the threshold;
- wide frequency *bandpass* at 1000Hz and unity amplitude, and;
- *fundamental tracking* switched on for more precision and less partials.

Closed-hole techniques: the SDMAS amplifies lots of partials creating a sound that is indistinct. This technique results in a wide variety of microsounds that are dependent upon what physical point of the instrument the microphone is attached or pointing at. Amplifying the body of the flute result in more high-end frequencies being amplified, whereas amplifying the head joint results in lower frequencies. The addition of fundamental tracking results in a more distinct sound, and mixing both the source and microsound signals into the output produces the most detail. Other settings are the same as with the whistle tones mentioned above.

Aeolian tone: for this technique it is best to have a wide bandpass without fundamental tracking due to the wide range of frequencies produced. The produced effect is not as complimentary to the aeolian tone as it is to other techniques as the SDMAS produces what could be described as filtered white noise, rather than augmenting already existing timbral qualities. As above, all other settings are the same.

Key clicks: the SDMAS amplifies the pitch of the key clicks and accentuates difference tones produced by trilling the clicks. Placing a microphone at the head joint gives more pitched material than amplifying at the keys. Tapping the different sections of the flute with fingernails is also very effective as the SDMAS amplifies the sounds resonating from the head joint, like a percussion instrument. The most effective settings are:

- threshold -24dB;
- axis point -12dB, and;
- with the bandpass filter set to a narrow band centred on 1000Hz at unity.

Other settings are the same as above.

Soft sounds (*ppp* to *mp*): because these 'soft' sounds are by comparison loud, the best settings are:

- threshold -12dB;
- axis point -8dB, and;
- narrow bandpass filter at 1000Hz and unity amplitude.

The effect on these pure tones (non extended techniques) is the amplification of clear overtones with some breath and mechanical noises. It is both effective and highly expressive in real-time performance.

Louder sounds (*mf* to *fff*): due to the amount of sonic information contained above the amplitude threshold at this dynamic, the SDMAS amplifies a lot of partials, producing an effect that is very distorted. This sound changes and can be shaped with different tone qualities and embouchure shapes. For example, in the low register (C4 to F4), a warm, open tone will result in a less distorted microsound signal than a hard, edgy tone that has many more amplified partials. While this effect could facilitate an interesting study of tone colour, it is quite overpowering and its use is limited. The most effective settings are:

- threshold -24dB;
- axis point -10dB, and;
- a wide bandpass centred on 1000Hz at -6dB.

Multiphonics, and flute and voice: both techniques produce sonically similar to the above when used with the SDMAS. If the flute techniques can be played at a lower dynamic (*ppp-mp*) then the SDMAS is able to focus on only a few overtones producing usable and interesting results. However, at louder dynamic just like with the unaltered sounds above, it produces a distorted signal that is not very useful.

Exploiting the idiosyncrasies in creative practice

Having explored the possibilities of a new sound world with the SDMAS, the next step is to internalise these effects and develop control over them, both with the acoustic instrument (adapting the playing to the different effects of the software) and with the device (controlling the settings). There are two ways to go about that the latter: leaving the SDMAS settings in place or having interactive control over them. The first option is much simpler and can have very effective results, however the performer is limited to certain techniques, due to the incompatibility between different techniques within the SDMAS; such as whistle tones and conventional timbres played at *forte*. This clearly limits the expressive potential of the instrument and the SDMAS. *Wilf's Lament* (Bennett 2014) was composed with this in mind, exploring the quieter sounds of the flute and using the SDMAS to amplify the minute detail of whistle tones, harmonics and difference tones, rather than using louder, pure sounds which risk SDMAS distortion and loss of control.

Figure 8: Excerpt from Wilf's Lament (Bennett 2014)

Interacting with the SDMAS directly and in real-time would be the most functional way to perform highly interactive pieces and utilise this newly uncovered sound palette. This could be achieved either by collaborating with a sound technologist during performance (such as in Time and Motion Study II (Ferneyhough 1978))or by the flutist using a foot pedal. As with any ensemble performance, communication is key and the performer must be very clear with the sound technologist about what kind of effects they are looking for, desired settings, and so on. This arrangement is suited to written music where the technologist is able to follow a score and make the necessary changes when instructed or cued, but may not be as useful in improvised performance.

In improvisation, a foot pedal can give complete control of the software to the performer, though this requires a considerable amount of preparation and coordination between hands and feet, which is not usually part of a flutist's performance practice. To facilitate this, a floor-based MIDI control pedal must be programmed into the Max patch, with the required pedals being mapped to pre-set functions that the performer can then employ as needed. An example might be having the settings most conducive to a whistle-tone being mapped to button number 1, while the settings best suited to key clicks could be mapped to button number 2, and so on. A foot pedal could be usefully mapped to control settings based on the pre-set, such as threshold, axis, EQ-width, and the mixer settings, enabling a performer to adapt the settings to suit the technique they are employing in real-time, and to exclude any unwanted background noise generated by the FFT or the room.

The performer must practice using the controls so that they are comfortable with them as they are with their acoustic instrument and able to move their feet without detracting from the performance.

Concluding Remarks

This paper has explored the technical workings of the SDMAS system, and an approach to utilising it in a live performance context. It uncovers some problematic areas and suggests future development to the system, along with methods of circumvention or adaptation to get the system to work in these situations. The processes described herein offer insight into the sonically rich worlds of *amplitude domain* microsound, as distinguished from the conventionally used *temporal domain* microsound, and offers a novel and flexible system for exploiting those microsounds through FFT in both a real-time and non-realtime contexts for instrument augmentation or analysis. Specifically, we examined how a solo flutist could develop an improvisational and compositional language based upon the idiosyncratic sounds that are inherent in flute performance, dramatically expanding the sonic palette of the instrument and building upon those expansions already mentioned in the introduction, through extended techniques.

References

Bailey, Derek. 1993. *Improvisation : its nature and practice in music*. New York: Da Capo Press.

Bennett, Alice. 2014. Wilf's Lament.

Berio, Luciano. 1971. Sequenza VII; per oboe solo. [London]: [Universal Edition].

Boulez, Pierre. 2010. RÈpons [1980-1986]. Belluno-Venezia: Colophon.

Coleman, Ornette, Eric Dolphy, Don Cherry, Freddie Hubbard, Scott LaFaro, Charlie Haden, Billy Higgins, and Ed Blackwell. 1998. *Free jazz [a collective improvisation]*. New York: Atlantic.

Ferneyhough, Brian. 1978. Time and motion study : 2, solo 'cello and electronics. London: Edition Peters.

Finsterer, Mary. 2001. Ether : for amplified solo flute. [Canberra]: Llumin.

Giles, Vincent. 2013-2016. Microsound, Spectra, and Objectivity: A meme's-eye view of comprehension in music (PhD research: incomplete). edited by University of Melbourne. Unfinished.

Lachenmann, Helmut. 2004. "On My Second String Quartet ('Reigen seliger Geister')1." *Contemporary Music Review* 23 (3-4):59-79. doi: 10.1080/0749445042000285681.

Méfano, Paul. 1980. Traits suspendus : pour flûte octobasse ou pour flûte basse en sol. Paris: Salabert.

Penny, Jean. 2009. "THE EXTENDED FLAUTIST: Techniques, technologies and performer perceptions." Queensland Conservatorium Griffith University.

Roads, Curtis. 2001. *Microsound*. Cambridge, Mass.: MIT Press.

Schoenberg, Arnold. 1978. *Theory of harmony*. Berkeley: University of California Press.

Stockhausen, Karlheinz, and Karlheinz Stockhausen. 1968. *Gesang der J¸nglinge Kontakte*. Germany: Deutsche Grammophon.

Ziegler, Matthias. 1999. *Uakti*. New York: New Albion.

[1] This problem of summed frequency and amplitude data over time is the subject for further research into microsound.

[2] This is the inverse of the complex convolution example in the Max tutorial system: https://docs.cycling74.com/max5/tutorials/msp-tut/mspchapter26.html

[3] https://github.com/vgiles/SDMAS