“*Machinatuosity*”: Virtual Strings, Spectral Filters and Temperament Tools for *Esquisse*

Richard Dudas
Music Research Center, Hanyang University College of Music
dudas@hanyang.ac.kr

ABSTRACT

This paper describes the technological component underpinning the author’s composition Esquisse (in Memoriam J.-C. Risset), *for piano and computer, in the form of real-time signal processing and synthesis implemented in the* Max *visual programming language. The computer part, designed specifically for this piece, incorporates some novel techniques, including an extension of the Karplus-Strong algorithm that permits the synthesis of string harmonics via one simple control parameter, and a spectral-domain filtering system based on comb-like harmonic filters which also incorporate spectral-domain bandpass “windows” the can be calculated on either a linear frequency or octave scale. The practical result of this is the ability to create harmonic filters representing a single formant of any size or full-spectrum harmonic filters whose fundamental changes in different parts of the spectrum. The piece also incorporates some specially designed tuning tools to reconcile the equal-tempered tuning of the piano with the use of the harmonic series in the electronics as a prevalent compositional device used throughout the piece.*

1. INTRODUCTION

This year’s conference centers around the idea of the computer as virtuoso. This word conjures images in our minds of those grandiose, extrovert 19th century instrumental virtuosi such as Paganini or Liszt. It is therefore no surprise that the Latin root of this word is *vir* – man – which could be understood in a broader sense in our somewhat more inclusive and egalitarian era as “human”. (Let’s not forget the 19th century women who also were revered as virtuosi – Clara Schumann, for example.) Thus, *virtus* – initially denoting manliness and courage and later virtue, character and, above all, excellence (regardless of gender) – and its modern offspring, virtuoso, applies to *human beings* and their positive, constructive and alluring creative abilities. So, how could the idea of virtuosity be adapted to include a *machine*? Perhaps, where computers are concerned, we should coin a new word and instead call it *machinatuosity*.

Certainly, virtuosity comes in a broad spectrum of forms, and, indeed, its meaning encompasses not only great skill and technique at some artistic endeavor, but also a great capacity for experimentation within that endeavor. In a mixed computer music composition for combined instrumental and technological forces, the creative skill is likely to be in the hands of the person who experiments with computer music techniques and creates computer software to be used in concert. So, perhaps, after all, the computer itself is just there to express the virtuoso work of the computer music designer?

Then, what exactly is the rôle of the anthropomorphized *machinatuoso* machine in the arts, today? Does the computer need to conform to that 19th century idea of being a larger-than-life showman with flying fingers? There is probably not a single “one size fits all” answer to these questions, since different compositions and performance scenarios will obviously require their own unique solutions. But where the computer component of my composition, *Esquisse*, is concerned, the computer’s somewhat understated *machinatuosity* on the surface is supported by the creative and computational virtuosity of the machine’s underlying compositional and signal processing algorithms.

2. EXTENSIONS TO THE KARPLUS-STRONG ALGORITHM

Kevin Karplus and Alex Strong’s 1983 CMJ article discussing their efficient and “surprisingly rich and natural” sounding algorithm for the synthesis of plucked-string and drum timbres [1] itself includes several variations on the basic theme, a block diagram of which is shown in figure 1.

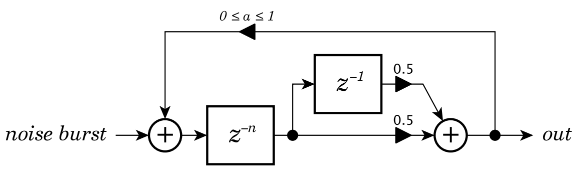

Figure 1. The basic Karplus-Strong algorithm, where n is the delay time and a is the feedback coefficient. A simple averaging filter provides the lowpass filtering for the feedback loop.

Their basic algorithm can be represented by the difference equation

$$y_n = x_n + (y_{n-N} + y_{n-(N+1)}) / 2. \qquad (1)$$

In their conclusions and discussion of ideas for future research, Karplus and Strong explored possible modifications and additions to the algorithm, especially the idea of "more complicated modifiers in the feedback loop" than had been described in the paper, and the idea of "cross-coupling" two delay lines tuned to different pitches. Some of the ideas they touched upon were further elaborated on in the subsequent article in the same issue by Jaffe and Smith [2]. One such elaboration included the use of a comb filter applied to the initial noise burst in order to attenuate or suppress certain harmonics for the simulation of the pick location along the string.

2.1 Comb-filtered feedback

In my own experiments with the Karplus-Strong algorithm, which I had initially only intended to use for the purpose of artificially – though hopefully also convincingly – extending the duration of the piano's low string resonance (something which was a compositional necessity in the first sketches for *Esquisse*), I wondered what would happen if I replaced the one-sample averaging delay in the feedback loop with an averaging delay greater than one sample – i.e., a simple comb filter. For my initial experiments, I used a small delay-time difference, and noticed this produced a dampening effect, such as one would get when muting piano strings near the end of the string. Further tinkering with the algorithm led to implementing a double tap on the delay line – i.e., more or less the above-mentioned cross-coupling suggested by Karplus and Strong, albeit with an averaging filter on each tap – at equal and opposite distances from a theoretical "central" delay time to define the fundamental pitch of the string. This modification, shown in figure 2, like Smith's comb-filtered noise burst, allows the virtual string itself to be "touched" at various nodes in order to suppress different harmonics as it resonates.

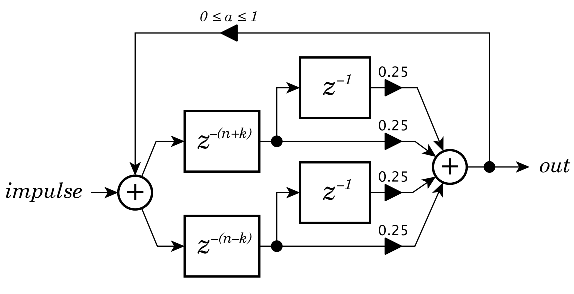

Figure 2. The modified Karplus-String algorithm used for *Esquisse*, incorporating comb-filtering on the output of the (double) delay line. One delay is k samples less than delay time n and the other is k samples greater. The effect of this is the creation of regular notches in the otherwise harmonic spectrum, which allows certain harmonic partials to be accentuated or attenuated. (Note that the two multi-sample delay lines could share sample memory.)

The difference equation for this modification is

$$y_n = x_n + (y_{n-(N+k)} + y_{n-((N+k)+1)} + y_{n-(N-k)} + y_{n-((N-k)+1)}) / 4, \quad (2)$$

where k is some fraction of the total delay time N. For example, a half-string harmonic can be obtained where k is $N/2$. Some harmonics have alternate fractions which can be used to obtain them. For example, the k value for the eighth harmonic (three octaves above the fundamental) can be obtained by $N/8$ or $3N/8$ – each of which provides the same harmonic, but with a slightly different timbre (particularly noticeable in the attack portion of the note).

As mentioned earlier, if k is very small – in the range of $N/80$ to $N/120$ – this creates a notch in the upper end of the spectrum and results in the string sounding audibly dampened, thereby resulting in a timbre similar to that created by physically dampening the piano strings near the end of the with the fingers.

2.2 Choice of Impulse

To get a piano-like hammer attack, instead of using a filtered noise burst, which can impart a metallic, artificial quality to the resulting sound, I experimented both with hand-drawn multi-sample impulses and synthesized impulses created from multiple low-frequency waveforms. Through trial and error, I found the hand-drawn impulses to be audibly richer and more convincing in terms of their "instrumental" sound quality than the synthesized ones. Some of my hand-drawn impulses were based on research by Migneco and Kim into synthesizing excitations for plucked guitar synthesis, [3][4] and adapted empirically and intuitively to try to imitate something more akin to a piano hammer, than a pluck. A set of piano-hammer-like impulses was created and, upon subjective listening, the impulses were arranged on a scale from bright to dark, so the timbral quality of the impulse could be an eventual musical parameter used in the piece, as necessary. An example of one of the impulses is given in figure 3.

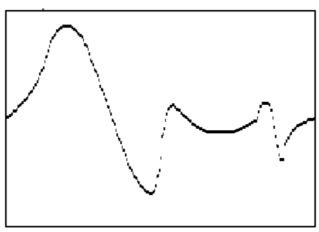

Figure 3. An example of a hand-drawn time-domain piano-hammer-like excitation impulse created to be used with the modified Karlus-Strong algorithm.

In order to create sustained piano string tones (as this was the compositional impetus for the project in the first place), the option to use filtered pink noise as a continuous excitation source alongside or in place of the pseudo-hammer impulse was added to the algorithm. (It was both simple as well as musically effective.)

3. SPECTRAL FILTERING

The Max 8 software [5] includes a couple of new audio-processing extensions [6] (or "objects", as they are called in Max terminology) designed to make it easier to convert lists of values representing amplitude spectra into audio-rate "spectral frames" used inside a *pfft~* subpatcher, which itself implements a Short-Time Fourier Transform (STFT). [7] These two new objects are *frame~* and *framesnap~*, both originally designed and implemented by the author, and now incorporated into the canonic Max software distribution. The former converts a list of values representing a frequency-domain amplitude spectrum into a static "frame" which is repeatedly output as an audio signal within in the pfft~ subpatcher, while the latter does the inverse: capturing a single frame of spectral-domain audio data within the pfft~ subpatch and outputting it as a list of values representing a "snapshot" of the spectrum at that point in time. (These two objects are analogous to *sig~* and *snapshot~*, respectively, in the time-domain set of Max objects.) The *frame~* object, in particular, greatly simplifies the creation and application of intricate spectral filtering functions to frequency-domain signals.

For *Esquisse*, a spectral-domain harmonic filter was designed to be able to filter individual harmonic partials, or bands of partials, from a harmonic spectrum, and be able to dynamically transition between these without resorting to the use of glissandi, which could potentially highlight frequencies present within in the residual noise outside the spectrum being filtered. This filter also allows a harmonic sound to be filtered with a harmonic series based on a fundamental frequency other than its own, if desired.

The control parameters for the filter are *fundamental frequency* in Hz, *central partial* (with 1 being the fundamental, 2 being the 2nd partial, etc.), and *spread* (in number of partials – i.e. this is a "bandwidth" around the central partial, expressed in number of partials). Figure 4 shows some examples.

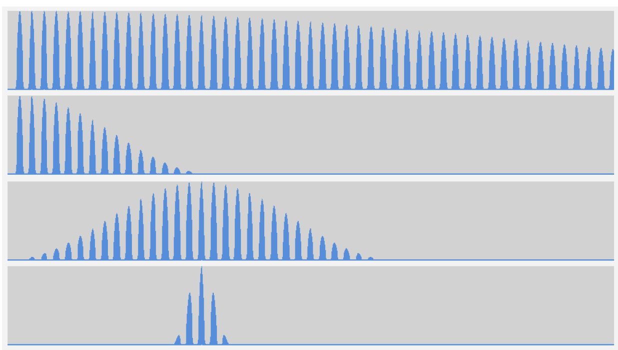

Figure 4. Spectral filters graphed from 0 Hz to SR/4. All have a fundamental of 220 Hz. The top two filters have a *central partial* of 1 (the leftmost partial) and the bottom two filters have a *central partial* of 16 (4 octaves above the fundamental). The *spread* values, from top to bottom, are: 200, 32, 32 and 5.

Just as the *spread* parameter represents a notion of bandwidth, the idea of a *central partial* was also adapted by the author from the idea of a central frequency as a defining parameter of a bandpass filter. The shape of the harmonic filtering function underneath the larger filtering band defined by these parameters, was originally simply sinusoidal. However, this allowed too much noise from the excitation to pass through the filter. After experimenting with several variations of a sinusoidal function, a final filtering form, shown in figure 5, was obtained. It was created by applying a square root to the absolute value of a cosine function, reapplying the original sign, scaling and translating it to be unipolar and then squaring the result. This was done in order to maximize the attenuation between the peaks of the harmonics, while also allowing the peak itself to be somewhat wide at the top:

$$y = (\mathrm{sign}(\cos(2\pi x f)) \cdot \mathrm{sqrt}(|\cos(2\pi x f)|) \cdot 0.5 + 0.5)^2 \quad (3)$$

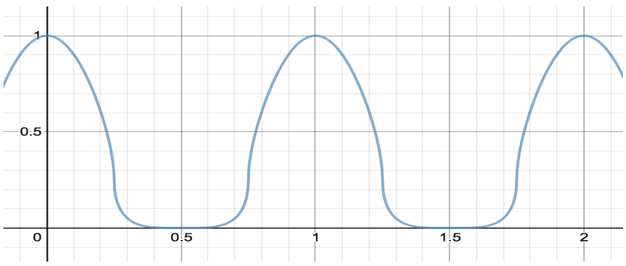

Figure 5. A close-up view of a graph of the shape of the partials in the partial filtering function, with f=1.

3.1 Octaviation

The spectral filter is also provided with an *octaviation* parameter similar to that implemented by Michael Clarke in the CSound FOF unit generator, [8] but with an upward change in octave (as opposed to a downward one). As this parameter moves from 0 to 1, odd partials in the spectral filter are attenuated so the resulting perceptual pitch of the filter subtly transitions an octave upward; values between 1 and 2 perform the same operation for the next upward octave transition, and so forth. This is shown in figure 6.

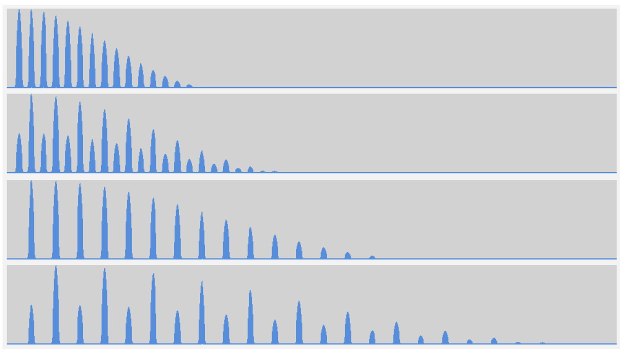

Figure 6. Spectral filters graphed from 0 Hz to SR/4. All have a fundamental of 220 Hz, a *central partial* of 1 (the leftmost partial) and a *spread* value of 16. The *octaviation* parameters, from top to bottom, are 0, 0.5, 1 and 1.5.

3.2 Pitch-scale filters

A variation on the spectral filter created for *Esquisse* is a harmonic filter with two fundamental frequencies which alternate every octave (or any other user-defined interval). This was achieved by creating a sinusoidal-shaped spectral filtering function whose peaks and troughs are equally spaced on the pitch-scale (but which get larger on the frequency scale as frequency increases). The peaks and troughs can be set to any interval. This filtering function and its inverse are then multiplied by harmonic filters with different fundamental frequencies, as shown in figure 7. In the context of the piece, this is used to interleave the partials of two harmonic series with fundamentals a minor third apart, in order to make a direct allusion to the minor-third-based pitch material used in the piano part.

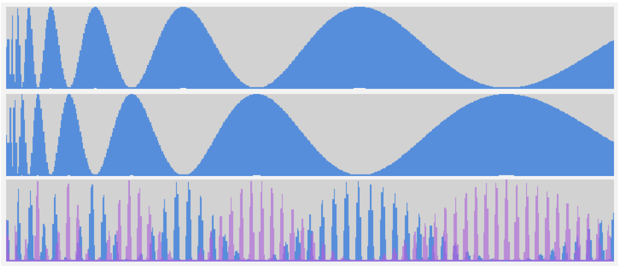

Figure 7. Octave-based pitch-scale filters, graphed from 0 Hz to SR/4. The upper graphs are inverse of each other and have peaks and troughs every octave, centered on the pitch A. The lower graph shows 2 harmonic series filters – tuned to A 110 Hz and F# 92.5 Hz – multiplied by these. The combined result is a spectral filter whose harmonic series alternates each octave.

4. TUNING AND TEMPERAMENT

One of the major hurdles was dealing with the inherent out-of-tune-ness of the equally-tempered piano juxtaposed with synthesis and sound processing based on the harmonic series. Since the piano cannot re-tune itself, it is the computer which must make any necessary adjustments. The first musical gesture at the very beginning of *Esquisse* provides a good case study for the kind of tuning reconciliation which needs to be performed throughout the piece.

The effect of this musical gesture is that the low Bb on the piano – 29.135 Hz or MIDI note 22 – makes an upward harmonic glissando to the 11^{th} partial, E – 1318.51 Hz or MIDI note 88. However, the 11^{th} partial of Bb, with a two-octave upward adjustment, is actually 1281.95 Hz – noticeably out of tune with the E on the piano. Conversely, the E on the piano is the 11^{th} partial of a low Bb whose pitch (with a two-octave downward adjustment) is 29.966 Hz. The solution to this in *Esquisse* is a smooth series of constant and subtle adjustments to the fundamental frequency of the pitches being synthesized by the computer. For this first gesture, this means that the low Bb fundamental gradually raises in pitch while the spectral filters create a harmonic glissando to E. When the piano enters on that E, its pitch magically seems to match that of the Bb harmonic series.

The patch therefore includes some basic calculation tools that allow any equal-tempered pitch to be defined as a given harmonic partial of a fundamental frequency defined as a fractional MIDI pitch.

5. CONCLUSIONS

The tools created for *Esquisse*, although used subtly in the context of the composition itself, are virtuosic – or rather *machinatuosic* – by their very nature, since they allow the computer to excel at doing what it does best: to perform complex calculations at lightning-speed so that they can be applied creatively and artistically in the context of a musical composition. The modifications to the Karplus-Strong algorithm used for the piece, while they may have been hinted at in the conclusions of the initial paper itself, show that there is still room for extending these old ideas and using them in new and creative ways. Future improvements to the spectral filtering system created for the piece include re-implementing it in coded form for efficiency. Finally, the thinking about tuning and temperament has a broader application to future compositional work.

6. REFERENCES

[1] K. Karplus and A. Strong, "Digital Synthesis of Plucked String and Drum Timbres," *Computer Music Journal*, Vol. 7, No. 2, pp. 43–55, 1983.

[2] D. Jaffe and J. Smith, "Extensions of the Karplus-Strong Plucked-String Algorithm," *Computer Music Journal*, Vol. 7, No. 2, pp. 56–69, 1983.

[3] R. V. Migneco, and Y. E. Kim, "Modeling plucked guitar tones via joint source-filter estimation," *Proceedings of the 2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, Sedona, AZ, USA, IEEE, 2011.

[4] R. V. Migneco, and Y. E. Kim, "A Component-Based Approach for Modeling Plucked-Guitar Excitation Signals," *Proceedings of the 2012 NIME Conference,* Ann Arbor, MI, USA, 2012.

[5] D. Zicarelli et al., *Max 8*, Cycling '74, Inc., https://cycling74.com/products/max

[6] D. Zicarelli, "An Extensible Real-time Signal Processing Environment for Max," *Proceedings of the International Computer Music Conference*, Ann Arbor, MI, USA, pp. 463-466, 1998.

[7] J. B. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, Signal Processing,* Vol. ASSP-25, pp. 235-238, June 1977.

[8] M. Clarke, "FOF and FOG Synthesis in Csound," *The Csound Book*, ed. R. Boulanger, The MIT Press, Cambridge, MA, USA, pp. 293–306, 2000.