# DEVELOPING REAL-TIME SYSTEMS FOR CONCERT PERFORMANCE

*Richard Dudas*

Hanyang University School of Music
Seoul, South Korea

## ABSTRACT

This artist talk describes the compositional and technical aspects of several of the author's compositional projects using the Max/MSP graphical programming environment. Musical choices and audio signal processing technologies used in the context of a real-time performance system often have a tightly interwoven relationship: sometimes technological developments are motivated by musical necessity, and conversely the construction of some of musical material is a consequence of technological considerations. This will be elucidated through examples from the author's series of preludes for flute, clarinet, and alto flute. In addition to discussing the signal processing effects used in these pieces, an important emphasis will also be made on the pitch tracking and score following techniques used.

## 1. INTRODUCTION

The idea behind a series of short preludes for solo instrument with real-time computer interaction was partially inspired by Mario Davidovsky's *Synchronisms* series [2]. As this collective title suggests, these pieces explore synchronization in addition to extending the timbral and gestural possibilities of the instruments used; this seemed like a useful general model for a series of short interactive pieces. In contrast to many recent compositions for instrument(s) and real-time computer interaction which sometimes overstay their welcome with audiences, the Davidovsky pieces are not particularly lengthy (a couple of them are indeed quite brief), and this mix of brevity and conciseness was also an influential factor as a musical model for the author's preludes.

## 2. IMPORTANT DEVELOPMENTS

The pieces which will be discussed in this talk all owe their existence to an important development: the gizmo~ object for real-time transposition of audio signals in the Max/MSP environment [4]. The gizmo~ object was developed by the author specifically with the first of these compositions in mind, and the subsequent compositions in the series also rely on it. Although by 2002 the signal processing capabilities of the Max/MSP program had been vastly expanded beyond the basic toolkit of objects that was provided with the first release of MSP [10], there were nonetheless a few necessary musical tools still missing from the environment. Some of these gaps in the tool set were addressed by diligent third-party developers who made their work available to the Max/MSP user community (for example Mille Puckette's pitch tracking objects, such as fiddle~ [8]), yet other tools were absent entirely, or of an unsatisfactory musical quality. One important musical item lacking in the environment was a way to transpose sounds in a natural, realistic manner, as the granular time-domain transposition unit provided (based on earlier analog models [1]) as an example patch with the program left a lot to be desired. Therefore the author embarked on the creation of a spectral-domain transposition object based upon published research into the phase vocoder and spectral domain effects by Laroche and Dolson [5]. The result was the development of the gizmo~ object as a standard part of the Max/MSP object set.

## 3. COMPOSITIONAL WORKFLOW

### 3.1. The Instrumental Score

The instrumental score of each of the preludes for instrument and computer was generally composed first, keeping in mind the electronic processing that would be used in different sections of the pieces. The realization of the computer part, in the form of a Max/MSP patch, was designed afterward according to indications in the instrumental score, at which point the score was often modified both for musical reasons and in order to accommodate technical limitations, such as unreliable pitch tracking. The melodic and harmonic aspects of the composition are freely composed, using no pre-determined compositional system for note choice. Generally, however, the melodic aspect of all the compositions follows diatonic sets of pitches that modulate to different pitch sets via common tones.

### 3.2. Audio Processing Design

Apart from some preliminary tests to try out sound processing ideas at the early stages of composition, the signal processing for these pieces was constructed based on the musical ideas and processing requirements indicated in the instrumental score. The processing was designed using Max/MSP, although the signal processing flow can be described in non-software-specific terms which could be used to re-create the real-time computer part of any of these pieces in the future using other audio signal processing software, if necessary. An example would be the block diagram for the *Prelude for Flute and Computer* (see figure 1). Each

 ISSN 1448-7780 

of the preludes has its own distinct signal processing data flow, although the pieces do share some common processing algorithms, such as transposition and delay of the live input sound, recording into audio buffers and granular or algorithmic playback of the recorded sounds.

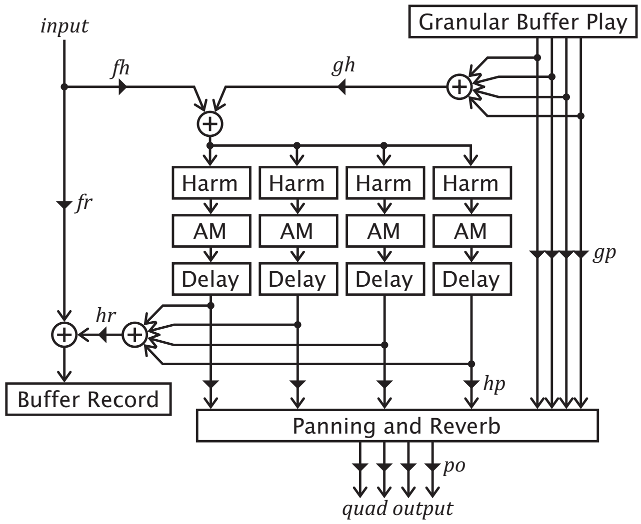


**Figure 1.** *Prelude for Flute and Computer* block diagram.

### 3.3. Compositional Choices and Necessities

A fundamental conceptual and compositional choice for the series of preludes is that the computer part begins with an "empty slate" – there is no pre-recorded material and all sounds originate from the onstage performer in concert. Some sounds from the live instrument are recorded for use later in the piece, while other sounds from the live instrument are delayed using multiple delay lines in the patch. The delay times used in the pieces are generally set using a tempo derived from the live performer.

One additional but unfortunate compositional necessity imposed on the score during the realization of the Max patches was the addition of notes in passages which had one or more repeated notes. Although human performers and accompanists are adept at recognizing discrete note events, even when the same note is repeated multiple times in a musical passage, the current computer-based score following systems are less adroit. Therefore additional notes, not harmonically related to the repeated notes, were added in some places throughout the score in order to aid the computer pitch tracking system. Although future pitch trackers may be more capable of dealing with a wider variety of musical input, these extra notes have, nonetheless, become part of the final composition.

## 4. EXISTING TECHNOLOGIES

There are two important pre-existing techniques without which the preludes could not exist: pitch tracking and score following.

### 4.1. Pitch Tracking

The first two pieces in the series, the *Prelude for Flute and Computer* and the *Prelude for Clarinet and Computer,* use a modified version of an older pitch tracker by Miller Puckette called pt~, which was originally designed for Max/FTS on the Ircam Signal Processing Workstation (ISPW) [7]. This object has proven to be much more responsive than newer pitch tracking objects for MSP such as Puckette's fiddle~ and sigmund~. Although with the flute's timbre, the pt~ object will often report erroneous pitches one octave from the actual fundamental, the incorrect octave can be corrected at the score-following stage For the clarinet, the pt~ object is rarely incorrect in its pitch and octave estimation, due to the clarinet's odd-harmonic spectrum, thereby allowing for very accurate and fast on-the-fly transpositions of individual note events.

For the *Prelude for Clarinet and Computer*, the original pt~ object needed to be modified in order to allow larger window sizes for the object's internal FFT, thereby accommodating more accurate pitch tracking of the lower notes in the clarinet's chalumeau register. Another modified version of the object, named ptg~, was also compiled specifically for this composition. It can be used inside a standard MSP pfft~ subpatch, and therefore contains only the pitch tracking code, not any code specific to the Short Term Fourier Transform, which is handled by the pfft~ object. Having the pitch tracker inside the pfft~ subpatch is useful for two reasons: 1) if a the pfft~ object is already being used in the patch for spectral-domain sound processing, a second (computationally expensive) Fourier Transform is not necessary when adding pitch tracking to the patch, and 2) it facilitates synchronization of the reported pitch estimation with any spectral-domain effects that need to use that information.

### 4.2. Score Following

An updated version of the score following system designed by Miller Puckette and Cort Lippe on the ISPW was used for the flute and clarinet preludes [9]. For the moment it works quite robustly with these two pieces, and makes use of objects in the standard Max/MSP distribution. The system uses the afore-mentioned pitch tracker in conjunction with the Max detonate and qlist objects. Both objects have been a regular part of the Max distribution since the mid-1990s. The three-part system works thus: the pitch estimation from the pitch tracker is sent to the detonate object, which contains a piano-roll MIDI representation of the entire score [6], in addition to cues which are output as the follower advances through the score in response to the pitch-tracked input. These cues are then sent to the cue list in the qlist object, and the events of each cue are then remotely sent to control different parameters in the patch.

### 4.3. Anticipatory Score Following

The *Prelude and Fantasy for Alto Flute and Computer* uses a newer pitch tracking and score following system:

Arshia Cont's antescofo~ object [3] for score following and event triggering. Although this third party object, available from IRCAM, is still in development, it is already stable enough to have been used successfully in musical productions in and outside IRCAM. The advantages of antescofo~ over the older pitch tracking, score-following and message-sending system is that it is particularly robust with fast musical material, has a built-in tempo estimation which can be used for tighter rhythmic interaction, and from a compositional point of view, merges the (text-based) score representation with the event and message-sending, allowing both the musical score and signal processing parameters to be viewed together in a single document.

## 5. COMPUTING TECHNIQUES

In addition to the techniques described above, the preludes share several novel signal-processing techniques. Some of these techniques are extensions of existing technologies, whereas others were designed specifically for use in the pieces.

### 5.1. Rhythm and Tempo Detection

A series of cues are used at several occasions in the pieces for clarinet and alto flute in order to set the tempo of various computer-processed events. In the *Prelude for Clarinet and Computer* the first of these occurs at the outset of the piece, where a series of cues sets the tempo for a sequence of transpositions of the live note. Additionally, the opening notes of the B section of that piece are used not only as a bridge from the musical material at the end of the A section, but also to allow the clarinetist to define the tempo for the following section. In the fast fantasy section of the *Prelude and Fantasy for Alto Flute and Computer* timed note sequences are adjusted according to the tempo of the regular quintuplets played by the live flautist. In both these pieces, a collection of delta-times are obtained from a series of cues, the values are sorted and the highest and lowest values are thrown out, since they could potentially be false readings, or inaccurate rhythms on the performer's part. The remaining timing values are averaged, producing an accurate tempo estimation. This tempo is used to calculate the timing and rhythm of musical events executed by the computer, including accurately setting the canon entries in the clarinet prelude, or inserting notes during the performer's rests in the alto flute piece. In all cases this allows the performer to set a comfortable tempo for the computer interaction in a natural, and musical manner.

### 5.2. Timbral Modification

In the clarinet and alto flute pieces some special timbral modification techniques are introduced. Since the clarinet's instrumental acoustics generate odd harmonic partials only, a principal effect used in the clarinet prelude is to fill-in the even-harmonic gaps in the clarinet's spectrum. This has the effect of transforming the clarinet sound into that resembling an alto saxophone. The effect is achieved simply by pitch-tracking the clarinet sound and using the fundamental frequency as a Hertz value to frequency-shift the audio. When the frequency-shifted audio is mixed with the original signal, it fills in the even harmonics in the clarinet spectrum. The frequency-shifted audio may be added to the original sound at any arbitrary volume level, thereby allowing smooth changes in timbre along the continuum between clarinet and saxophone. The alto flute piece also makes use of a special transposition unit which allows individual partials of the sound to be moved to alternate harmonic or inharmonic locations in the spectrum. This is used particularly to transform the spectrum of the alto flute into a bell-like spectrum.

### 5.3. Spectral Delay with Transposition

Although all three preludes are structured with four transpositions of the original instrumental sound which can each be delayed an arbitrary amount of time, two sections of the clarinet piece in particular required a special recursive transposition and delay effect: the end of the A section and the end of the coda. For these places an alternate spectral-domain transposition patch was made. In it, each of the four gizmo~-based transpositions is delayed a certain number of FFT frames in the spectral domain sent to the subsequent gizmo~ object. The fourth transposition is sent in a feedback loop to the input of the first. By using carefully-adjusted parameters for the transposition and volume attenuation of these transpositions, a "hazy" or "out of focus" decaying echo sound is produced, quite different from the effect that would be made if the effect were designed in the time-domain.

### 5.4. Glissandi

Although not a novel technique per-se, the use of an S-curve for portamento is more musically effective than a linear glissando. The standard curve~ object for MSP implements a signal-rate quasi-exponential or quasi-logarithmic curve. Two curves can be triggered successively in order to create an S-curve, which approximates well the effect of a human glissando, such as that of a finger on a violin string: starting slowly, accelerating through the center portion of the glissando and slowly easing into the destination pitch.

### 5.5. Pitch Grid

The most important development for the *Prelude for Clarinet and Computer* takes a small inspiration from the world of algorithmic composition, albeit performs it in the context of a digital signal processing effect. The author designed a pitch-class corrector as an event-level Max patch, whereby all incoming (MIDI) pitches are modified to fit to a pre-defined pitch grid. Each input note is compared to a given pitch-class set, and then transposed to the nearest note in the set. When used in conjunction with the robust ptg~ pitch tracker and the gizmo~ object, each note played by the clarinet may be adjusted to a different fundamental frequency in real-time. In some ways this is the antithesis of the ubiquitous auto-tune effect in popular music, as it does not correct the fine-tuning of the note, but rather moves the entire note event to a different pitch-class level, preserving vibrato and microtonal nuances inherent in the original sound.

This effect is used heavily in the canonic B section of the composition. In this section the computer-delayed canons of the live clarinet sound are transposed to different key centers, but the individual notes of the canons are modified in order to match the current harmonic context implied by the live soloist. The effect enters subtly at first, but becomes more obvious as the piece progresses. By bar 31, the transposed canons are no longer following the original diatonic material, but rather are transposed to fit to a triadic harmonic grid.

The *Prelude and Fantasy for Alto Flute and Computer* also makes use of a similar on-the-fly transposition technique for the algorithmic playback of the short notes recorded in the opening section of the piece.

## 6. CONCLUSION

The technology and musical ideas behind the author's preludes for instrument and computer are at once a culmination of previous musical and technological work as well as a springboard for future compositions in the series. Especially, the algorithmic nature of the processing in the latter two of these pieces lends itself to the concept of hybridizing digital signal processing techniques with ideas taken from algorithmic composition.

## 7. ACKNOWLEDGEMENTS

The author would like to acknowledge composers Erik Oña and Andreas Breitscheid, who, in 2005 and 2006, respectively, organized the workshops and concerts that made the creation and premiere of the first two preludes possible.

## 8. REFERENCES

[1] Bode, H., “History of electronic sound modification,” *Journal of the Audio Engineering Society*, Vol. 32, No. 10, New York, NY, USA, 1984.

[2] Chasalow, E., “Mario Davidovsky: An Introduction,” *AGNI 50*, Boston University Press, 1999.

[3] Cont, A., "Antescofo: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music," *Proceedings of the International Computer Music Conference*, Belfast, Northern Ireland, 2008.

[4] Dudas, R., "Spectral Envelope Correction for Real-Time Transposition: Proposal of a 'Floating Formant' Method," *Proceedings of the International Computer Music Conference*, Göteborg, Sweden, 2002, pp. 126-129.

[5] Laroche, J. and Dolson, M., “New Phase-Vocoder Techniques for Real-Time Pitch-Shifting, Chorusing, Harmonizing and Other Exotic Audio Effects,” *Journal of the Audio Engineering Society*, Vol.47 No.11, Nov 1999.

[6] Puckette, M., "Explode: A User Interface for Sequencing and Score Following," *Proceedings of the International Computer Music Conference*, Glasgow, Scotland, 1990, pp. 259-261.

[7] Puckette, M., "FTS: A Real-time Monitor for Multiprocessor Music Synthesis," *Computer Music Journal* 15(3), MIT Press, Cambridge, Massachusetts, USA, 1991, pp. 58-67.

[8] Puckette, M., Apel, T., Zicarelli, D., "Real-Time Audio Analysis Tools for Pd and MSP," *Proceedings of the International Computer Music Conference*, Thessaloniki, Hellas, 1998, pp. 109-112.

[9] Puckette, M. and Lippe, C., "Score Following in Practice," *Proceedings of the International Computer Music Conference*, San Francisco, California, USA, 1992, pp. 182-185.

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