VIRTUAL BASS-CLARINET IN MODALYS

Hans Peter Stubbe Teglbjærg
"Composer in Research" (IRCAM)
stubbe@post.nordit.dk

Thomas Goepfer
"Computer Music Designer" (IRCAM)
thomas.goepfer@ircam.fr

KEYWORDS

Physical modelling, Realtime, Bass-clarinet, Multiphonics.

ABSTRACT

This article describe how the authors and the Modalys-team (IRCAM) has collaborated to develop and implement a realtime model of a complete bass-clarinet to be used in the first authors composition "Rippled Reeds" for bass-clarinet and electronics, that explore the relation between an acoustic woodwind instrument and a similar virtual one.

The intend was to create a model that allowed fingerings to be specified generating predictable output, in order to overcome a number of practical shortcomings when chaining complicated forked fingerings producing multiphonics. The model should run in realtime so that it could be driven from the actual acoustical playing.

After a brief introduction, the first section will reflect about the choice of synthesis method and motivate its compositional use.

Second section describes how a model was created from measurements of a true bass-clarinet complete with reed, mouthpiece, tube, bell and holes, sufficiently efficient to run in realtime on one Intel processor.

Third section describe the interface and how it raised questions of preset interpolation and multithreading to obtain polyphony. The conclusion draw the contours of what was achieved and outlines possible further studies.

1. INTRODUCTION

1.1 The Composer Voice – briefly !

Does a new music require new instruments? Ask Partch, Nancarrow, Haba ... or Schöenberg - if we could! [13].

Digital artists has since long left the question behind for instrumental composers to try to make a new music with traditional chromatic instruments. Woodwind-instruments were for the most part perfected about 150 years ago. New quartertone instruments (fx. flute, clarinet, piano) continues to rest outside current musical practice because of a strongly living traditional repertoire. Schöenberg foresaw that the emerging microtonalism would not find into a central position of current musical practice without adapted instruments.

Concerning multiphonics the situation is roughly the same : current instruments are not particularly adapted.

In practice multiphonics pose a certain number of problems for both the interpreter and the composer. Every multiple sounds demand a different embouchure, their dynamic profiles are limited, their chaining are difficult due to the forked fingerings. They are difficult to sequence with other modes of playing.

It would seem that other means should be searched for; like a virtual replica, that can be played virtuoso by the computer. The first author therefore wished to develop a realtime model of multiphonics on a virtual bass-clarinet. A virtual model allows not only to even out these difficulties, but also, thanks to the rigourous control possible with the computer, to force the model of a multiphonic towards other non-standard vibratory behaviours [7,8]. Thus leaving what is very difficult, even barely possible, in real life to an extra-human device?

1.2 Why Physical Models ?

After more than 20 years of practical application physical models (PhM) still hold a great potential, esp. when more advanced models arrive and substantial knowledge about how to control them is developed [4]. Among its main features must be counted : attractive richness and realism in transients, stimulating complexity in interaction between excitation mechanism and resonator, its obvious instrumentality and possible expansion hereof, the many vibratory behaviours possible from a single model. In the authors view PhM offers a synthesis method which is very well adapted and coherent with current compositional practices.

Modal synthesis [5] was chosen because of its modularity and user-programability, and because it currently is the most advanced on interaction with an extensive control in realtime. Templates allow for quick prototyping, while a set of basic connections allow the user to defined customised interactions. Any parameter can be either directly modified in realtime or pass via a user-defined functional expression to their destination parameter. Such high degree of modularity is to the authors knowledge not available in other PhM environments except at the coding-level itself.

It deserves mentioning that recent advances of wind-instruments, such a brass and saxophone [6], has stimulated the modelling of a new air-mouthpiece interaction, that convincingly captures the non-linear coupling between air and reed. It effectivly replaces Modalys' rudimentary reed-connection by a more elaborate normalised-valve connection [5].

1.3 Why Real Time ?

The first author has made a number of pieces based on PhM mostly non-realtime [9,10]. Though PhM in realtime (RT) has been exploited for some time its complexity and processing demands has prevented extensive usage. With increasing processing power more advanced models can run in RT and appropriate live control-systems be investigated.

RT control is important when exploring the "playability" of a model either in the studio or on stage. It allows one to "sense" the interactions of the instrument and immediately evaluate its output. The authors wished to control individual keyholes of the virtual bass-clarinet in close co-ordination with fingering sequences specified for the live player. Each hole interact dynamically with the tube via a hole-connection that can be gradually opened or closed.

Realtime control of the model was used both in the studio and on stage. It proved a very effective device in searching a parameter-space. Modal synthesis tend to have many interacting parameters, certain changes may cause multiple effects, such fx. as lowering number of modes may increase overall amplitude, due to less modes that interact. These can be quickly estimated through aural feedback.

The parameter-space of modal synthesis is not linear in the sense that linear interpolation of physical parameters would create an even-spaced progression from one timbre to another. Interpolating presets had to be carefully considered. Interpolation presets does not necessarily create an "interesting" transition between otherwise fascinating states. RT feedback proved again the quickest way of finding "zones of interest".

We wanted to put the models live capabilities to a test, making full use of its many controls in RT. By pushing the limits we wanted to address problems thus provoked and device solutions, some of which will be discussed under implementation in section 2.

1.4 Multiphonics – an extended timbre-study

Many has noted the "electronic" sound-quality of multiphonics [1]. True, there is a bit of a ring-modulator gone wild about them. But they are more than just interesting new timbres. The way they physically work and manifest themselves as "multiple sounds" goes beyond the mere coloristic. For the first author multiphonics constitute not only a wish for a large sound palette, but a quest for composing at the principles of forked fingerings in an integral way. A search for a link between a physical mechanism and its manifestation in sound where multiple components participate in the co-ordinated production of more than one layer of sound.

Understanding the physical-acoustical conditions necessary for the production of multiphonics in woodwinds has been described by some authors [3]. Analysis of multiphonics naturally asks for a notational reduction for it to be practical. Main criteria for listing them found in methods is lowest (heard) pitch, usually with some additional information about dynamic range, speed of response, timbre, ease of playability, possible alternations, etc.

Multiphonics are seldom those fixed entities they appear in the methods [1,2]. They have emerging qualities that largely determine the order of appearance and salience of each component, many of which the player can control. Clarinettists are fx. known to be able to produce different multiphonics from the same fingering only by changing aspects of the embouchure, reed-position and air-pressure. This domain is rarely systematically described. The first author consequently attempted a physical-acoustical categorisation of the multiphonics based on 3 criteria :

1) **perceptual**, concerns how a multiphonic is perceived. An empiric, but useful number of etiquettes is proposed :

- *homogene / heterogene*
- *pure / electronic*
- *fine / coarse*
- *fused / split*
- *stable / rolling*
- *harmonic / inharmonic*

A main division is if the multiphonic provokes a single fused percept or multiple simultaneous percepts.

2) **physical**, concern how a multiphonic is obtained (esp. with regard to type of fingering). They where labeled :

- *natural*
- *register-key (proper or fake)*
- *perforated*
- *trill-keys*
- *forked*
- *resistant (altissimo-fingerings underblown)*

Each type of fingering describe a specific relation between air-column and nodes.

3) **embouchure**, concern how the multiphonic is excited. This rather intricate category consists of :

- *reed-position* of lips at tip, mid or base of reed allowing more or less of the reed to vibrate.

- *lip-pressure* on the reed and how much air that slips through the sides of the mouth.

- *air-pressure* determines together with lip-pressure the speed of air passing the reed / mouthpiece.

- *reed-type* either **soft** (elastic) helping the production of multiphonics or **hard** (stiff) enforcing fundamental and a strident timbre.

- *oral cavity* is known to play an important role, but is relatively little described in the literature.

For fingerings whose output is ambiguous and difficult to notate exactly an action type notation combined with finger-tabulature was adopted. A graphical notation of the players embouchure was invented and is described in the score [11].

**1.5 Embouchure** – a link to bifurcation ?

Certain embouchure and fingering combinations seem to point towards a bifurcative behaviour that has been theorised about by some authors [7,8]. Such behaviours has never been reported in connection with multiphonics. The authors believes certain embouchure techniques may constitute the missing link. Both in real and simulated attempts has spectra been achieved that divide in a bifurcative fashion. This domain needs a more systematic study in order to advice precise practical playing techniques. In simulation however a precise physical model has been realised that can produce the full route from normal to bifurcative to chaotic behaviour. This instrument is controlled in realtime by mapping from the onstage instruments playing position between two microphones to a "strength" parameter that controls degree of bifurcation.

2.   MAKING OF AN INSTRUMENT

2.1 Implementation – a true collaboration !

We aimed at a modelisation that would closely match the properties of a real bass-clarinet, complete with reed, mouthpiece, tube, bell and 32 holes (last hole being at the bell). A true bass-clarinet from the company Buffet-Crampon was taken apart and its geometric dimensions measured. With a set of MatLab scripts these data where used to calculate a mesh using the finite-element method of representing 3D objects [5]. This actually modelled the whole interior air-column of the bass-clarinet as one single composite object (see figure 1). The mesh was then turned into modal data (mode-frequencies, mode-losses and corresponding mode-shapes) which Modalys can read directly from a file. This data now represents one big resonant object, whose impedance-curve match that of the measured true bass-clarinet with all keys closed. This work was undertaken by the team for acoustics of musical instruments (IRCAM).

Modalys exist in several versions (MatLab, LispWorks, OpenMusic, MaxMSP) that can co-communicate. The model was read from file into a Lisp-script in order to create all necessary accesses, additional objects (reed, keys), connections (normalised-valve, hole-connections) and controls (dynamic controllers for all parameters). Apart from being non-RT the Modalys Lisp-version is a perfect user entry. Once executed this lisp-code produced a final textual script that can be run in realtime from MaxMSP (using the extern modalys~) applying value-changes to all predefined controls via messages. No further definitions takes place, one "plays" ones "instrument". Special care was taken to create a flexible reed on a detachable mouthpiece, and to make the hole-interaction as simple as possible. The final instrument has about 40 interactions, approx. 50 primary physical parameters and 90 secondary parameters to be controlled in realtime.
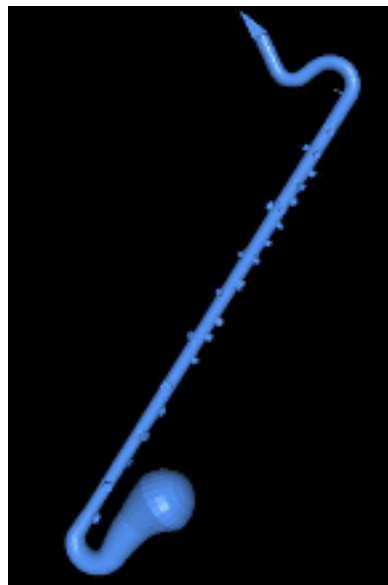
Figure 1.
A mesh of the interior air-column of a bass clarinet.

2.2 How to blow the instrument ?

The instrument is blown by augmenting an air-pressure parameter (called gamma) to a critical value. Air-pressure is really a time-dependent parameter one has to vary in a smooth fashion. Abrupt changes are likely not to work and high air-pressure values may likewise block the reed from vibration. Each register demanded air-pressure and embouchure to be adopted differently. Multiphonics had usually their own settings. The time spend in establishing which blowing would work for which fingering tells how delicate blowing really is - as in real life!

Holes are opened by setting the appropriate hole-weight to non-zero thus activating the hole-connection. When hole-weight is set to zero the hole is closed and the hole-connection will be deactivated (freezed) in order to save CPU. All holes had the sasme initial parameters tuneable from the interface. Each hole radiates a certain amount of energy that will be added to the final sound when the hole is open. The two register-keys where however treated with the cheaper speed-connections as their radiation do not contribute to the sound. A key-to-hole mechanism was not included in the model. Instead a scheme for all conventional fingerings for corresponding opening and closing of holes was saved as presets for instant recall. Forked fingerings found in methods where usually created as variant of regular ones and saved as presets.

2.3 Completing the instrument

A special feature allows the mouthpiece to be detached from the rest of the tube - while playing and still sounding ! This procedure would not be possible in real playing and thus shows an example of how the virtual

instrument can extend the real one. The idea of gradual coupling / decoupling has required the implementation of a weight parameter in all connections. The weight parameter can grade the connection from fully connected to fully deconnected. This feature is also used when opening and closing holes.

Another important implantation is the very efficient expression-controller that allows user-specified functional expressions to be evaluated in real time. It is used for a multitude of purposes like mapping a single dynamic controller to multiple actions, to make tests of states like if a connection has weight equal zero then freeze it, or to perform tasks in the background. It has a rich syntax including most arithmetic operators, conditional tests, and special operators for retrieving and storing information in objects, controllers or connections. Expression controllers are fx. used to operate on modal data in realtime. Retrieval and storing of information can also be executed from MaxMSP using the getinfo and setinfo message to modalys~. This is handy when the state of an object is unknown (or exploded!) and needs to be reset to a known state.

Volume-control of a dynamic system like Modal synthesis is a topic by itself. Modalys in non-RT normalises by default, which is handy for the generation of soundfiles. No such feature exist for RT use. One must either scale the audio output from modalys~ or build scaling functions into the script. One way to proceed is to run the script in non-RT with the auto-normalisation flag off and the then look for the maximum output sample. One may find that the value range of Modalys' 64 bit calculations is so small that it will not display in MaxMSP's floatboxes. Using an expression controller one may create a function that dynamically adapt the overall output volume according to the a running average of the maximum sample so far. For the practical use all instruments had to have approx. the same output volume.

2.4 Optimisations – stairway to heaven !

A good number of codelevel optimisations where carried out by Nicholas Ellis (IRCAM) in order to achieve a full model running without clicks on one Core 2 Duo intel processor. This included making Modalys benefit from the processors 64 bit calculation possible from Leopard 10.5.7. A double gain was achieved; in terms of speed and in terms of resolution. Physical modelling requires a large value range to be able to represent movements of very large and very small masses. 64 bit resolution is effectively well adapted for this representation.

In the Lisp-script optimisations where mainly achieved using the expression controller. Fx. send in real time of certain basic physical parameters would cause the entire model to reinitialise and actual values to be lost. Instead specialised expression controllers where use to affect only desired parameters without causing re-initialisation. A somewhat counterintuitive "feature" is that the more open holes (:the more interactions) the more demanding the model in CPU.

An expression controller was also used to freeze hole-connections when weight equals zero, thus saving in CPU. In the final version of the model the CPU raised upto about 80% when all holes where open.

2.5 Code developments – a summary

These elements where added to the Modalys application for the realisation of this project :

- *weight* parameter in all connections
  for grading degree of interaction
- *expression controller*
  allows functional expression to evaluate in RT
- *spring connection*
  a twosided stiffness-controlled connection
- *normalised-valve connection*
  replacing the older reed-connection
- *auto-normalisation* on/off
  flag for non-RT use
- *get-info* and *set-info*
  more complete set of operators

3.   COMPOSING FOR THE INSTRUMENT

3.1 Interface – for multiple simultaneous instruments

Alltogether 9 virtual instruments where developed, 2 of them complete virtual bass-clarinets. The 7 remaining ones are derived from singling out various aspects of the complete model. They may be characterised as follows:

- *soloist instruments* are complete virtual
  bass-clarinets, one with a detachable mouthpiece.
- *modulator instruments* are either modulated
  from the patch or by the real player.
- *resonator instruments* takes players sound-input
  and filters it through modal objects.

The second author made an interface for each instrument with a coherent style of GUIs and use of presets [12]. Due to a rather large number of primary and secondary parameters for the complete model we chose to split that interface into two parts: one covering everything concerning the holes and one concerning blowing, reed, tube and modal data. Each with its own storing and retrieval system. The interface groups the models many parameters according to their functionallity. In other words we kept fingerings apart from the instrument allowing us to test various blowings against any series of fingerings. We were conceptually at ease with this distinction.

Each instrument interfaces includes appropriate value-limits for all parameters to avoid clicks and leaks from excessive control values, and a limiter to act a volume safety-gaurd (Modalys can be very sensitive!).
The concert-patch is essentially an event-player that can trigger sequences of presets and affect specific parameters

via messages, including interpolated presets for fingerings, blowings and tube design for various types of instruments.

The intention was to control the compositions 9 modalys instruments in realtime, with a maximum of 3 running simultaneously. However, to achieve a satisfactory richness in sounding result much modulation was necessary. The CPU strain became very high and a reduction necessary. Instruments that was not modulated from either the event-player or the soloist where therefore bounced in realtime in the studio and instead triggered as premixed 6-channel soundfiles including a volume-control. For the realtime control in the studio two BCF2000 where connected to MaxMSP and mapped to modalys parameters. This proved a very fruitful studio-use of Modalys in realtime.

From a mixing point of view the sonic result is deliberately kept close to Modalys. No sound appear which is not either from Modalys (whether bounced or in realtime) or from the real player, fx. no external samples where used. However due to the realtime demand of the instruments a lowering of number of modes where often necessary, resulting in fewer high partials being present in many of the sounds. To compensate for this high-cut characteristic bounced sounds where fed through a harmonic exciter to regain some of the lost higher partials. Various other post-production treatments where used sparingly only to enhance the sonic projection of the otherwise pure direct output from Modalys.

No realtime treatment was involved except for some reverb from the main mixer to fit the hall-acoustics.

3.2 Preset interpolation – or why does it work ?

Special considerations had to be given to automatic value interpolation. As described earlier the parameter-space of modal synthesis is not linear. Because of the dynamic nature of modal synthesis random switching or interpolation of presets would not garuantee to bring back a known vibratory state. We deviced a way of forcing the system to reset and learned to pay attention to the exact order and timing of switching presets. Interpolation of embouchure settings also proved very sensitive and these where mainly used in a switching fashion.

In addition we noted how the CPU strain during interpolating varies according to parameter type. To begin with we led Modalys interpolate everything by default but realised it caused click for certain basic physical parameters. We carefully examined the CPU strain of each parameter and left the critical ones uninterpolated by Modalys. For MaxMSP we adopted a system such that a preset could be recalled with a global interpolation-time and followingly any desired parameter modulated locally. Only values that has changed where actually sended. It was thus necessary to distinguish between the interpolation in Modalys which is done on samplebasis (more precise but costly) and in MaxMSP that perform it at vectorbasis (less precise but cheaper).

3.3 Polyphony

To be able to run several instruments simultaneously in concert, we looked into what can be achieved from the OSX multithreading capabilities on multi-core Macs.

Modalys can multithread its instruments provided no cross-references are made. It means that a single model or parts of it cannot be multithreaded. However all keywords for controllers must have individual names as they exist in a global namespace. The same holds for MaxMSP where multithreading can be achieved from poly~ but not generally for MaxMSP itself. Modalys~ inside a poly~ can be multithreaded. MaxMSP's namespace is also global, meaning local variables are not possible outside poly~ in multithread mode.

Many of the instruments have uniform names for controllers performing similar functions. This makes instrument definition and interface development easier and we decided to keep this convention. We therefore made each instrument an individual runtime application controlled from a central patch via OSC. Multiple runtime applications will automatically benefit from multithreading on a multi-core Mac. Each instrument is thus an independent application that easily could be integrated into other environments using OSC.

Output routing for each instrument-application was set to individual output channels, assembled and redirected in RMEs Fireface 800 matrix-mixer to the main PA mixer for live broadcast. The concert setup used for the first performance is described by the second author [12].

4. SUMMING UP

4.1 Parts of a conclusion

The model was not entirely in tune when compared to the tempered scale, thus predictability for forked fingerings found in referenced methods was only partially achieved. The deviation was as much as a semitone. Possible cause of deviation could be the measured dimension of the tube and holes, the hole of the register key that descend into the instrument like a small tube, and tuning of hole resonances, which maybe must be done on a individual basis. As a consequence precise pitch-contend had to be measured afterwards in order to be notated and correlated with multiphonics used in the composition.

However the various principles for forked fingerings produced in many cases spectral results strikingly close to the behaviour of real-life multiphonics. It is expected that better tuning of the model, fx. of hole resonances, combined with proper embouchure setting would raise predictability. As in real playing proper embouchure control plays a crucial role in obtaining desired multiphonics. Interpolation of embouchure proved esp. difficult for which further study is necessary.

Driving the model from the acoustic playing is but one solution. The model is sufficiently rich and flexible to be played on its own fx. by a wind controller. The implementation as a standalone applications allows easy integration with other styles and ideas. Appropriate strategies for realtime control of model in concert whose parametrical behaviour is complicated remain however a concern, as much for the composer as the developer.

The approach herein described yield an attractive structural match between how the composition was "thought" and how the synthesismodel was "controlled", as both could be organised according to a similar set of "fingerings". Compositional workflow was considerably enhanced through this unified representation. The instrumental part and the electronics does not exist in two parallel universes, but unfolds along a common line of thought. Multiphonics as a form of controlled distortion seem very promising, and may constitute a new aesthetic in the making ?

4.2 Acknowledgement

5. REFERENCES

Methods with fingerings for bass-clarinet multiphonics
[1]   Ph. Rehfeldt, "New Directions for Clarinet", *University of California Press* (1977).
[2]   H. Bok, "New Techniques for the Bass Clarinet", *special limited edition by H. Bok* (1989).

On the acoustics of multiphonics
[3]   J.Backus, "Multiphonic tones in the woodwind instruments", *JASA, 63, 591 (1978).*

On the physical modelling of clarinet tones
[4]   R.T. Schumacher, "Ab initio calculations of the oscillations of a clarinet", *Acoustica 48, 71 (1981).*

Modalys reference
[5]   Modalys, IRCAMs Physical modelling application, *http://support.ircam.fr/doc-modalys*.
[6]   Arturia Brass, 3 PhM Brass software instruments, *http://www.arturia.com/evolution/en/products/brass*.

References on bifurcative behaviour
[7]   F. Laloï et al., "Bifurcation, period doublings and chaos in clarinetlike systems", *Europhysics Letters 1 (6), pp. 295-302 (1986).*
[8]   Vincent Gibiat, "Chaos and musical instruments", *proceedings of ICA, Madrid, Spain (2007).*

First authors compositions based on physical modelling
[9]   H.P.S. Teglbjærg, "Regime_S" for 8.2 channel tape, *performance material by the composer (2007).*
[10]   H.P.S. Teglbjærg, "Kobar Qunar II" for clarinet and electronics, *score by the composer (2008).*
[11]   H.P.S. Teglbjærg, "Rippled Reeds" for bass-clarinet and electronics, *score by the composer (2009).*

Description of concert patch and technical setup for "Rippled Reeds"
[12]   Th. Goepfer, "Rippled Reeds", *IRCAM technical report No. 145 (2009), http://brahms.ircam.fr/sidney*.

On the evolution of new instruments
[13]   A. Schöenberg, "The future of orchestral instruments", *in "Style and Idea" Belmont Music Publishers (1975).*