PROCEEDINGS of ISMA 2019
International Symposium on Music Acoustics

13 - 17 September 2019 in Detmold, Germany

A real-time physical model to simulate player control in woodwind instruments

Vasileios Chatziioannou[(1)]*, Alex Hofmann[(1)], Sebastian Schmutzhard[(2)]

[(1)]Department of Music Acoustics, University of Music and Performing Arts Vienna, Austria
[(2)]Acoustics Research Institute, Austrian Academy of Sciences, Vienna, Austria

Abstract

The interaction between woodwind players and their instruments is a key aspect of expressive music performance. Therefore, recent studies have focused on understanding how the actions of the players affect sound generation. Based on experimental results obtained using both human performers and an artificial blowing machine, this paper presents a numerical model that, taking the players' actions into account, may synthesise expressive woodwind instrument sounds. Implemented in C++ the model allows real-time performance on a standard desktop computer, thus enabling the user to modify the model parameters in a live performance scenario. Apart from varying parameters related to the embouchure of the player and the effective length of the resonator (i.e. mimicking modifications that could take place in real playing) this model also allows virtual modifications that would not be possible to realize in the physical world, such as a cone gradually morphing into a cylinder and vice versa. Besides parameters related to the excitation mechanism and the geometry of the instrument, the user is able to modify the properties of the air inside the instrument and the magnitude of the viscothermal losses.

Keywords: Physical Modelling, Woodwinds, Sound Synthesis

1 INTRODUCTION

Using physical modelling techniques, it is possible to numerically reproduce sounds of musical instruments that are directly related to the physical phenomena that take place [1, 10]. Advances in numerical analysis and computer science allow the formulation of efficient physical modelling algorithms that are even suitable for real-time performance. In order for the produced sound to be perceived as realistic by the listener, it is of paramount importance to consider the way the oscillations of the instrument are excited. This player-instrument interaction has attracted a lot of attention in the last decade [2, 11]. Several studies have been conducted in order to analyse the performance of experienced players and understand how this affects the sound generation. Focusing on single-reed woodwind instruments, this work presents a model that attempts to capture the interaction between the player and the instrument that takes place at the excitation mechanism. The formulation of the model is presented in the next section. Section 3 gives some details on the implementation and efficiency of the algorithm, section 4 presents some results obtained with the proposed model and section 5 discusses the findings of this work.

2 PHYSICAL MODELLING OF SINGLE-REED WOODWIND INSTRUMENTS

Wave propagation in an axisymmetric tube of length L and cross-sectional area $S(x)$, x indicating the position along the length of the tube, can be modelled using the following equations for the acoustic pressure p and the particle velocity v

$$\frac{\partial p(x,t)}{\partial x} + \rho \frac{\partial v(x,t)}{\partial t} + z_v * v(x,t) = 0 \tag{1a}$$

$$\frac{\partial \big(S(x)v(x,t)\big)}{\partial x} + \frac{S(x)}{\rho c}\frac{\partial p(x,t)}{\partial t} + S(x)y_\theta * p(x,t) = 0, \tag{1b}$$

*chatziioannou@mdw.ac.at

where $*$ denotes convolution with respect to time, ρ is the air density, c the speed of sound and z_v and y_θ are respectively related to the viscous and thermal losses within the tube (see [9] for more details). The boundary condition at the open end of the tube may be given as

$$p(t,L) = S(L)z_r * v(t,L), \tag{2}$$

z_r being a time-domain radiation impedance. At the excitation end, where the player-instrument interaction takes place, the above wave-propagation model needs to be coupled with a model of the reed oscillation, which takes into account the actions of the player. This is achieved by introducing an additional nonlinear term in the equation of motion of the reed, which, in previous studies, involved only two nonlinear components, namely the Bernoulli flow through the reed channel, and the collision of the reed with the mouthpiece lay [6]. This additional term describes the interaction of the player's tongue with the vibrating reed. Thus the equation of motion of the reed takes the following form

$$m\frac{\mathrm{d}^2y}{\mathrm{d}t^2} + m\gamma\frac{\mathrm{d}y}{\mathrm{d}t} + ky - k_{\mathrm{tg}}\lfloor y_{\mathrm{tg}} - y\rfloor^{\alpha_{\mathrm{tg}}}\left(1 + r_{\mathrm{tg}}\frac{\mathrm{d}(y_{\mathrm{tg}} - y)}{\mathrm{d}t}\right) + k_{\mathrm{lay}}\lfloor y - y_{\mathrm{lay}}\rfloor^{\alpha_{\mathrm{lay}}}\left(1 + r_{\mathrm{lay}}\frac{\mathrm{d}y}{\mathrm{d}t}\right) = S_{\mathrm{r}}p_{\Delta}, \tag{3}$$

where y is the displacement of the reed from its equilibrium position, γ is the reed damping, α_{tg} and α_{lay} are collision exponents, r_{tg} and r_{lay} contact damping, y_{tg} is the tongue displacement, S_{r} the effective reed surface and p_Δ the pressure difference across the reed. $\lfloor y\rfloor^{\alpha} = \theta(y)y^{\alpha}$, where $\theta(y)$ denotes the Heaviside step function. Casting this in a Hamiltonian formulation and discretising using the finite-difference method, one may arrive at the solution of a nonlinear equation

$$F_{\mathrm{NL}}(y) = 0 \tag{4}$$

at each time step, whence the reed displacement and subsequently pressure and flow at the mouthpiece may be calculated (see [5] for full details). A key feature of this approach is that numerical stability is ensured via the conservation of the system energy. Figure 1 (left) shows the simulation of a clarinet staccato note, along with the error in the conservation of energy, which is of the order of machine precision. The virtual tongue has been used in order to release the reed (start the note) and stop the reed (end the note), in an attempt to replicate signals generated by professional clarinet players ([8]). Figure 1 (right) shows a similar plot for portato articulation; in that case the tongue is used to separate the two notes. The ability of the model to resynthesise sounds produced by an actual instrument has been validated using measurements with real players [4] and an artificial blowing machine [5].

3 IMPLEMENTATION

The numerical model was implemented in C++ within the *Csound Plugin Opcode Framework* [7]. This provides the physical model as an opcode to the sound synthesis environment of Csound. In order to experiment with varying model parameters, a simple graphical user interface has been designed using FLTK Widgets[1], including an oscilloscope that visualises the mouthpiece pressure (see Figure 2). The model runs faster than real time on a standard desktop computer (Intel® Xeon(R) CPU E5-1650; 16Gb RAM). Nevertheless, computational efficiency should still be considered, in case several such models need to run in parallel.

The main drawback in order to assess the efficiency of the model is the need to use an iterative solver for the resulting nonlinear equation (4). In the presented case, this is solved using the Newton-Raphson method. This method has the ability to converge very fast given a starting point close to the root of the nonlinear function F_{NL}. Indeed, even though a maximum number of 8 Newton iterations has been specified in the code, this number is never reached for any case of wind instrument simulation. The slow varying motion of the reed, in comparison to the sampling rate that is used (44100 Hz) ensures that by using the solution at the previous time step as starting point, the nonlinear solver converges to machine precision $\varepsilon \approx 10^{-16}$ usually after 2 iterations.

[1] https://www.fltk.org/doc-2.0/html/index.html

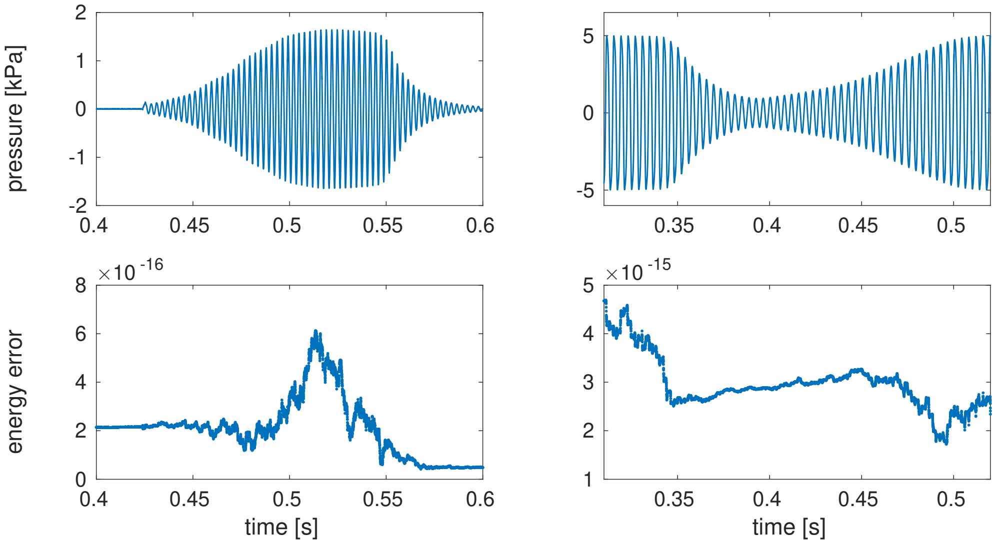

Figure 1. Simulated mouthpiece pressure during staccato (left) and portato (right) articulation. Model parameters are taken from [5]. The bottom plots show the error in the conservation of energy.

However, the fact that an unknown number of iterations is, in principle, required for convergence, introduces an uncertainty to such nonlinear solvers. For specific cases it is possible to pre-calculate the maximum number of iterations required to find the root of a nonlinear function [3]. Alternatively, there are different strategies to ensure computational efficiency. For instance, using a linear approximation of F_{NL} allows a direct solution subject to some loss of accuracy. A simple way to arrive at such an approximation is to perform only one Newton iteration per time step. This effectively acts as a local linearisation of F_{NL}. Given the slow-varying state of the system, in comparison to a high sampling rate, the accuracy loss is kept low. Such a strategy is particularly suited when oversampling is required (e.g. due to numerical dispersion or for perceptual considerations). Performing only one Newton iteration per time step, i.e. using a direct solution for the nonlinear equation, in the case of the examples shown in Figure 1, results to an energy error of 10^{-8} for staccato and 10^{-6} for portato articulation. Since a collision nonlinearity is modelled, which poses a rather strongly nonlinear element, this shows the feasibility of such an approach to increase computational efficiency. Using a higher sampling rate may reduce the approximation error when only one iteration is used, as summarised in Table 1. In practice, a compromise between computational efficiency and numerical accuracy can be made, e.g. enforce two Newton iterations per time step with a sampling rate of 44100 Hz.

Table 1. Order of magnitude of the error in the conservation of energy for different configurations of the number of Newton iterations per time step (Niter) and the chosen sampling rate (fs).

	portato		staccato	
	Niter < 8	Niter = 1	Niter < 8	Niter = 1
fs = 44100 Hz	10^{-15}	10^{-6}	10^{-16}	10^{-8}
fs = 200000 Hz	10^{-15}	10^{-8}	10^{-16}	10^{-10}

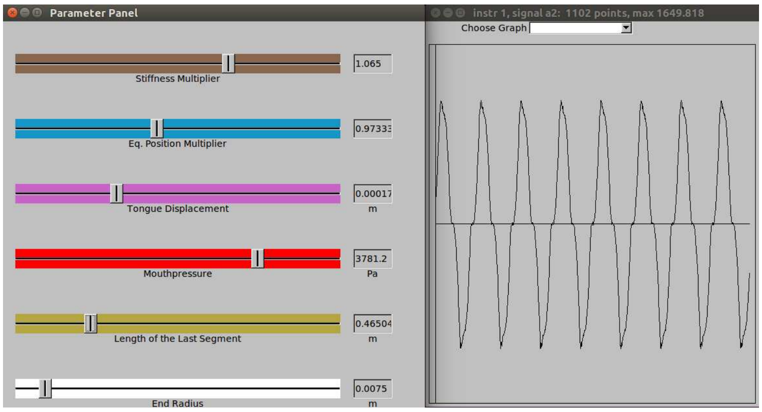

Figure 2. A graphical user interface to tune model parameters during the simulation. On the right a real-time plot of the mouthpiece pressure is generated.

4 NUMERICAL RESULTS

The presented model may be used to generate diverse woodwind sounds, while modulating the physical model parameters. One interesting case, that may be only virtually realised, is to gradually morph the shape of an instrument from a cylinder to a cone. The radiated sound of such a numerical experiment is visualised in Figure 3, in the form of a spectrogram. Besides the bore geometry, all other physical model parameters are kept constant. For the first two seconds of the simulation the bore has a cylindrical shape (with radius $r_{\text{in}} = r_{\text{out}} = 0.0075$ m) and in the next ten seconds the output radius linearly increases up to $r_{\text{out}} = 0.015$ m, while the input radius remains constant.

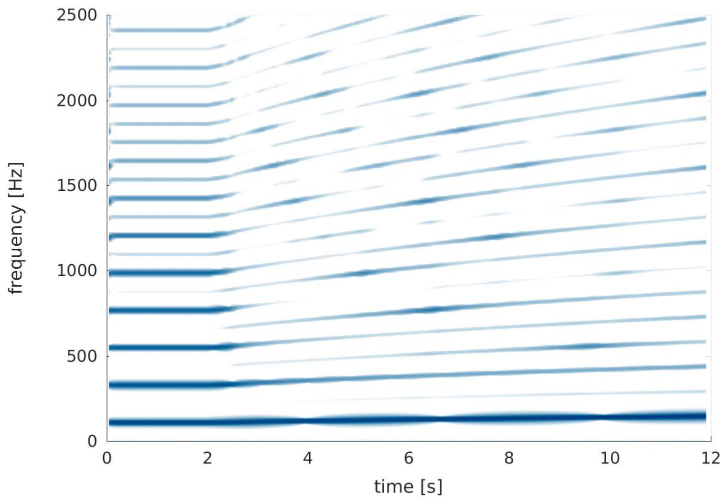

Figure 3. Spectrogram of the radiated sound of a single-reed woodwind instrument, the bore of which gradually morphs from a cylinder into a cone.

It can be observed that the odd harmonics are not present at the beginning of the tone, when the bore is cylindrical. They gradually appear when the instrument takes a conical shape. During that time all harmonics increase with increasing cone angle.

5 CONCLUSIONS

This work presents the formulation of a physical model, that may simulate embouchure control in single-reed woodwind instruments and is suitable for real-time performance. Numerical stability is ensured using an energy-based method. Implemented in C++, the model allows various approaches to control the model parameters during performance. The fact that all parameters have a direct physical interpretation allows an intuitive control over the system, whereas the virtual nature of the instrument gives infinite possibilities in terms of shaping the instrument geometry and varying model parameters related to the sound excitation mechanism.

ACKNOWLEDGEMENTS

This research is supported by the Austrian Science Fund (FWF): P28655-N32.

REFERENCES

[1] S. Bilbao. *Numerical Sound Synthesis*. Wiley & Sons, Chichester, UK, 2009.

[2] V. Chatziioannou and A. Hofmann. Physics-based analysis of articulatory player actions in single-reed woodwind instruments. *Acta Acustica united with Acustica*, 101(2):292–299, 2015.

[3] V. Chatziioannou, S. Schmutzhard, and S. Bilbao. On iterative solutions for numerical collision models. In *Proc. International Conference on Digital Audio Effects (DAFx-17)*, pages 72–79, Edinburgh, 2017.

[4] V. Chatziioannou, S. Schmutzhard, A. Hofmann, and M. Pàmies-Vilà. Inverse modelling of clarinet performance. In *Proc. 26th International Congress on Sound and Vibration (ICSV26)*, Montreal, 2019.

[5] V. Chatziioannou, S. Schmutzhard, M. Pàmies-Vilà, and A. Hofmann. Investigating clarinet articulation using a physical model and an artificial blowing machine. *Acta Acustica united with Acustica*, 2019.

[6] B. Fabre, J. Gilbert, and A. Hirschberg. Modeling of wind instruments. In R. Bader, editor, *Springer Handbook of Systematic Musicology*, pages 121–139. Berlin, Heidelberg, 2018.

[7] V. Lazzarini. The csound plugin opcode framework. In *Proc. 14th Sound and Music Computing Conference*, pages 267–274, 2017.

[8] M. Pàmies-Vilà, A. Hofmann, and V. Chatziioannou. Analysis of tonguing and blowing actions during clarinet performance. *Frontiers in Psychology*, 9:617, 2018.

[9] S. Schmutzhard, V. Chatziioannou, and A. Hofmann. Parameter optimisation of a viscothermal time-domain model for wind instruments. In *Proc. International Symposium on Musical Acoustics (ISMA17)*, pages 27–30, Montreal, 2017.

[10] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen. Discrete-time modelling of musical instruments. *Reports on progress in physics*, 69(1):1–78, 2006.

[11] J. Wolfe, N. Fletcher, and J. Smith. The interactions between wind instruments and their players. *Acta Acustica united with Acustica*, 101(2):211–223, 2015.