

# The BRASS Project, From Physical Models to Virtual Musical Instruments : Playability Issues

Christophe Vergez<sup>1,2</sup> and Patrice Tisserand<sup>1</sup>

<sup>1</sup> IRCAM, 1 Pl. Igor Stravinsky, 75004 Paris, France

<sup>2</sup> LMA-CNRS, 31 Ch. Joseph Aiguier, 13402 Marseille Cedex 20, France

[vergez@lma.cnrs-mrs.fr](mailto:vergez@lma.cnrs-mrs.fr)

[tisserand@ircam.fr](mailto:tisserand@ircam.fr)

<http://www.ircam.fr> <http://www.lma.cnrs-mrs.fr>

**Abstract.** The BRASS project aims to deliver software virtual musical instruments (trumpet, trombone, tenor saxophone) based on physical modelling. This requires to work on some aspects of the playability of the models so that they can be played in real time through a simple keyboard : better control of the attacks, automatic tuning, humanization.

## 1 Introduction

The goal of the BRASS project is to propose software virtual musical instruments (based on physical models) playable in real-time and controlled through a simple keyboard. Target instruments included in the project are the trumpet, the trombone, and the tenor saxophone<sup>3</sup>. This project has been done at IRCAM<sup>4</sup> and ARTURIA<sup>5</sup> and is supported by the RIAM<sup>6</sup> network. This paper is not a review of the whole project, but focuses more specifically on the strategies implemented to improve the playability of the models.

For a sound synthesis software to achieve the characteristics of a virtual instrument, two key features are needed :

- The first one is obviously the sound synthesis technique, which has to be flexible enough to provide natural variations of the sound when input parameters (controlled by the player) are altered. To do this, physical modeling has been chosen. The models are briefly discussed in section 2.
- The second key feature, which is more highlighted in this paper, is the playability of the software application. Indeed, significant effort has to be provided in order to transform a physical model (even if it has the intrinsic ability to produce typical sound effects of a given instrument) into an easily playable

---

<sup>3</sup> which is obviously not however a *brass* instrument

<sup>4</sup> IRCAM people involved sorted by name : A. Almeida, R. Caussé, X. Rodet, N. Schnell, P. Tisserand, C. Vergez

<sup>5</sup> ARTURIA people involved sorted by name : F. Bourgeois, Y. Bonnefoy, N. Bronnec, J. Germond, X. Oudin, F. Paumier, N. Pianfetti, S. Simmermacher

<sup>6</sup> <http://www.riam.org/riam/>

virtual instrument. In fact, unlike in a musical acoustics study, where squeaks and slangs are welcome since they highlight the ability of the model to reproduce typical features of the instrument, they have to be avoided here. Indeed, the main goal of the project is to immediately make the keyboardist feel like if he knew how to play well the instrument. Moreover, the player is supposed to have no other MIDI controller than those provided by a standard mid-range synthesizer. Therefore, significant efforts were put on the fine tuning of the model, so that each key pressed on the keyboard makes the model play the desired note (see section 3). This is far from being a straightforward task with a physical model. Finally, in order to overcome limitations generated by the control of a model through a keyboard, an additional layer of (higher level) control has been added between the keyboard output and the model input (see section 4).

## 2 Physical modelling

### 2.1 General principles

The physical models used in the BRASS project rely on a formulation of the physical functioning principles in term of nonlinear delay differential equations (popularized for sound synthesis of self-sustained musical instruments by [1]). The trumpet and the trombone models have been mainly developed during the Phd thesis of Christophe Vergez (see [2], [3] for a general description, and [4], [5] for precise aspects of the models). The saxophone model has been developed during the ongoing Phd of André Almeida on reed instruments (similar modelling principles applied to the oboe can be found in [6]). However, this paper is more specifically devoted to the trumpet and the trombone.

### 2.2 Modified model for the lips

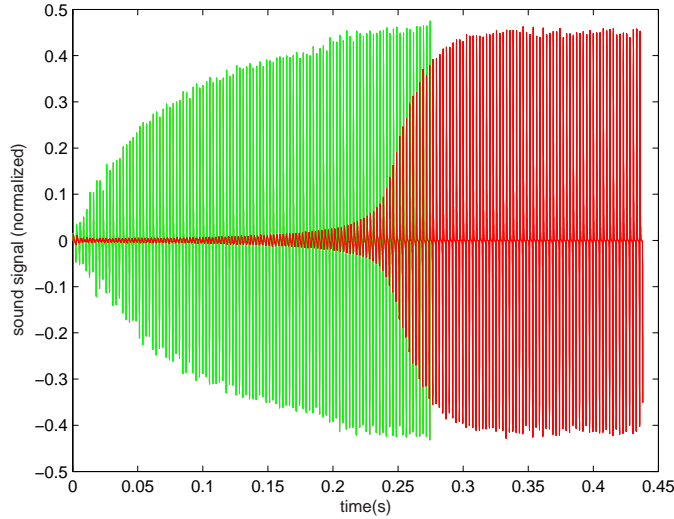
The classical single-mass lips model for the trumpet and the trombone has been slightly modified in order to obtain (and control) more typical brass sounds during the attack transient. Since the computational cost had to be kept as low as possible, the additional complexity of a two-mass model ([7], [8]) or even a single-mass model with two degrees of freedom ([9], [10], [11]) could not be afforded. Moreover, since the steady-state behavior of the model was satisfying, the modified model differs only in the first milliseconds of the sound, during the transient.

The new model proposes to take into account the influence of the tongue at the attack onset. In any brass instrument, the note starts when the tongue stops to stick to the lips and let the path free for the air flow. In spite of poor agreement between brass players on the precise tongue movement, its critical influence on the transient characteristics is well acknowledged. In this study, while the tongue itself has not been modelled, its influence on the lips dynamics is considered :

the quick removal of the tongue is supposed to generate a disturbance force on the mass of the form :

$$F = F'_l e^{-\alpha'_l t} \sin \omega'_l t \quad (1)$$

Equation (1) is applied until time<sup>7</sup>  $t^* \triangleq \frac{-1}{\alpha'_l} \ln \frac{10^{-3}}{F'_l}$ . The particular expression of  $F$  may obviously be thought as the impulse response of a second order damped system. However, equation (1) should be seen as an additional way of controlling the attack characteristics (depending on values of  $F'_l$ ,  $\omega'_l$  and  $\alpha'_l$ ), rather than a refined physical model deriving from a rigorous analysis. However, from the point of view of perception, the flexibility added might be associated with the different types of attack transients a brass player can produce using various tongue techniques (from a soft attack without any use of the tongue, to very pronounced attacks). In figure 1, two attack transients (note G4) are synchronized to highlight the influence of  $F$  (equation (1)) both on the length of the attack and on the shape of the envelope. Note that the additional force (equation (1)) is only taken into account until  $t^* = 22ms$  in figure 1.



**Fig. 1.** Comparison of two transients (note G4) : green one corresponds to  $F'_l = 0$  in (1), red one corresponds to  $F'_l = 3$ ,  $\omega'_l = \omega_l$ ,  $\alpha'_l = 350$ .

<sup>7</sup>  $t^*$  corresponds to the time where the amplitude of the force  $F$  as decreased by a factor 1000 until  $t = 0$ .

### 2.3 Control parameters of the model

Finally, the input parameters for the trumpet and the trombone models are given in table 1. They are divided into two groups : the tuning parameters and the control parameters. Values of the parameters inside the first group are automatically chosen (see section 3). On the contrary, parameters inside the second group can be mapped by the player to various controllers.

**Table 1.** Input parameters of the model for the trumpet/trombone

Tuning parameters	
Name	Description
$\omega_l$	Resonance frequency of the lips
$\alpha_l$	Damping of the lips
$l_b$	Length of the bore (i.e. valves position for the trumpet, and length of the slide for the trombone)
$\{A^i, \Omega^i\}_{i=1,2}$	Amplitude and Frequency of the additional modes introduced to compensate for the truncation of the reflection function (see [2] for details)
Control parameters	
Name	Description
$p_m$	Mouth (or blowig) pressure
$\lambda$	Amount of losses within the bore of the instrument
$\epsilon$	Amount of randomness injected in the air flow
$\Delta\omega_l, \Delta\alpha_l$	Variations around $\omega_l$ and $\alpha_l$

## 3 Automatic tuning of the physical models

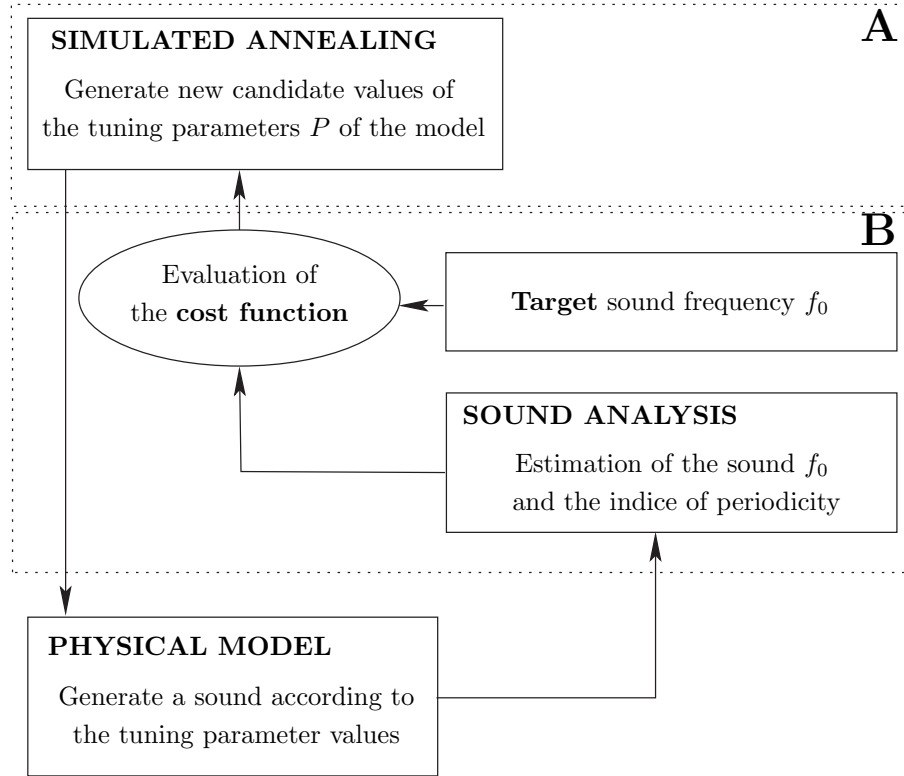
### 3.1 Problem statement

Physical models discussed in section 2 have proved to be well adapted for sound synthesis. However, the trumpet and the trombone models are particularly difficult to tune. Indeed, as it is well known since [12], the lips modal characteristics ( $\omega_l, \alpha_l$ ) influences significantly the frequency of the auto-oscillation.

Therefore, in order to avoid tedious adjustments (by hand) of the tuning parameters (listed<sup>8</sup> in the upper part of table 1), the tuning is considered as an optimization procedure : we are looking for the parameter values which make the model produce the sound whose fundamental frequency  $f_0$  matches at best

<sup>8</sup> For certain notes, the length of the bore is determined by the tuning process in order to allow fine adjustments of the playing frequency, just as a trumpet player does with the tuning slide

a target frequency  $\hat{f}_0$ . This comes to minimize a cost function (see sections 3.2 and 3.3). A general flowchart of the tuning method is presented in figure 2.



**Fig. 2.** Basic principle of the automatic tuning procedure by numerical optimization. The dotted squares marked **A** and **B** correspond to section 3.3 and 3.2 respectively.

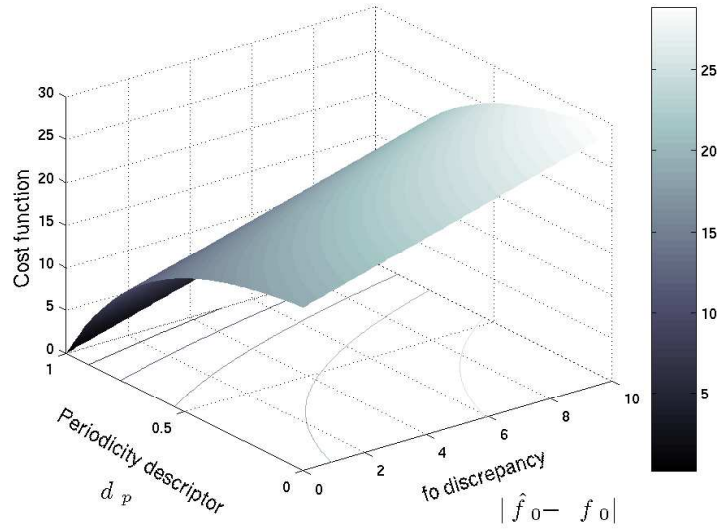
### 3.2 Cost function

The cost function  $\mathcal{C}$  is a measure of the discrepancy between the target response  $\hat{f}_0$  and the response  $f_0$  of the model to candidate parameter values  $P$  proposed by the optimisation process (detailed later in section 3.3). Therefore, the cost function  $\mathcal{C}$  may have many different expressions. The most intuitive expression of  $\mathcal{C}$  depending only on the square or the absolute value of  $(f_0 - \hat{f}_0)$  could not be retained. As a matter of fact, among all possible parameter values, only few of them make the model produce periodic sounds. Therefore, the cost function has to be penalized by a periodicity criteria. This is done by using a monophonic version of the fundamental frequency extractor recently developed at IRCAM ([13]) which also calculates a confidence rating of the proposed  $f_0$  (between 0 and 1), further used in this paper as a periodicity descriptor  $d_p$  (the more  $d_p$  is close to 0, the more we penalize the cost function). Finally the cost function defined by

(2) proved to be efficient to match the target  $\hat{f}_0$ , while avoiding quasi-periodic and non harmonic sounds (see figure 3 for a graphical representation).

$$\mathcal{C}(\hat{f}_0, f_0) = e^3 \left(1 - e^{3(d_p-1)}\right) + |\hat{f}_0 - f_0| \quad (2)$$

A probabilistic technique has been chosen to minimize the above cost function



**Fig. 3.** Graph of the cost function defined by equation (2) and contour plot highlighting the exponential dependance of the cost function on the periodicity descriptor  $d_p$ .

(see section 3.3), because it appeared to have many local minima.

### 3.3 Minimisation of the cost function through adaptative simulated annealing

**Generic Simulated Annealing :** simulated annealing technique (SA in the following) has been developed to statistically find the best global fit of a non-linear constrained non-convex cost function. This method is derived from the Metropolis method ([14]). Using notations introduced in section 3.2, the aim is to find the global minimum of  $\mathcal{C}(\hat{f}_0, f_0)$  defined by equation (2). The principle of SA is recalled below (see for example [15] p444 for details):

1. Choice (random or not) of an initial candidate  $P_0$  leading to a sound frequency  $f_{0_0}$ .

2. Choice of an initial temperature<sup>9</sup>  $T_0$
3. While temperature  $T_l > T_{\text{end}}$ ,
  - Do the following steps  $m$  times
    - Generate randomly a new candidate  $\tilde{P}_{k+1}$  (sound frequency  $\tilde{f}_{0_{k+1}}$ ), neighborhood of  $\tilde{P}_k$  (sound frequency  $f_{0_k}$ ) according to the probability density  $g_{T_k}$  (practically the neighborhood area decreases with  $T_k$ )
    - Calculate  $\Delta\mathcal{C}(\hat{f}_0, f_{0_k}, \tilde{f}_{0_k}) \triangleq \mathcal{C}(\hat{f}_0, \tilde{f}_{0_{k+1}}) - \mathcal{C}(\hat{f}_0, f_{0_k})$
    - If  $\Delta\mathcal{C}(\hat{f}_0, f_{0_k}, \tilde{f}_{0_k}) \leq 0$ ,  $\tilde{P}_{k+1}$  is automatically accepted as  $P_{k+1}$
    - If  $\Delta\mathcal{C}(\hat{f}_0, f_{0_k}, \tilde{f}_{0_k}) > 0$ ,  $\tilde{P}_{k+1}$  is accepted as  $P_{k+1}$  with the probability  $h_{T_k}(\Delta\mathcal{C}(\hat{f}_0, f_{0_k}, \tilde{f}_{0_k}))$ . Otherwise  $P_{k+1} = P_k$ .
    - $k \leftarrow k + 1$
  - Annealing schedule : decrease temperature ( $T_{l+1} < T_l$ ) according to the chosen cooling law
4. The last state  $P_{\text{end}}$  is statistically the best fit, i.e.  $\mathcal{C}(\hat{f}_0, f_{0_{\text{end}}})$  is the global minimum of function  $\mathcal{C}$ .

**Extension to an adaptative algorithm** Since computation time is often a limiting factor for probabilistic search of a global fit, we decided to take advantage of a refined SA algorithm developed by L. Ingberg and called VFR (Very Fast Re-annealing, [16]). This algorithm introduces adaptative (or re-annealing) capabilities to allow an automatic adaptation to changing sensitivities in the parameter space. Moreover it provides an annealing schedule for temperature  $T$  decreasing exponentially in annealing time  $k$ . As explained in [16], this is faster than more classical annealing schedules such as the Cauchy annealing (corresponding to the scheme  $T(k) = T_0/k$ ) and much faster than the Boltzmann annealing (where  $T(k) = T_0/\ln k$ ).

### 3.4 Results and discussion

*Optimization results* : the trumpet/trombone models have been tuned on three octaves. For a particular note, around thousand candidates have to be generated by the VFR process. The range of research for each parameter (i.e. the extrema values allowed) didn't appear to be of critical importance. Therefore, large ranges were generally used, so that they could be kept unchanged for many adjacent notes. The quality of the result given by the optimisation process is evaluated through real time playing of the model. If needed, the optimization process is launched again while advantaging the research against the rapidity of convergence. This is done by increasing the number of candidates generated at each temperature step (parameter  $m$  in the algorithm described in section 3.3).

<sup>9</sup> For historical reasons, the scheduling in SA is based on the analogy with temperature cooling

*Implementation / Computation time* : practically, the flowchart described in figure 2 implies code coupling, between the model itself, the  $f_0$  calculus and the VFR process. This can be very time consuming if the computation time is not kept in mind as a priority. Therefore the three standalone codes have been melt into a single code (function calls preferred instead of data files exchange). The tuning of a note lasts about 13 minutes (for 1000 iterations) on a low-end computer (Athlon XP 1800<sup>+</sup> running at 1533Mz with 768Mo of memory)

*Ease of use* : the optimization process presented above is far from being straightforward to use. In fact, several parameters of VFR have to be adjusted (initial temperature, number of parameters generated at each temperature, parameters of the cooling law . . .), which requires some experience and is definitely problem-dependent.

*Current research*: different notes produced by the model (tuned with the  $f_0$ -based optimization) may sound with timbre discrepancies. This is not very surprising since many parameters combination lead to the same sound frequency (but not necessarily with the same timbre). Therefore, current research is focussed on including timbre descriptors in the optimization. Flowchart presented in figure 2 remains unchanged, but instead of the only fundamental frequency, other timbre descriptors are considered ([17]). This is expected to provide a way of having the parameters tuned so that the model reproduces at best, not only a target  $f_0$ , but a target sound (possibly played by a real soloist).

## 4 Other improvements of the playability of the model

Physical models of wind instruments are preferably played in real time through breath controllers. The use of a keyboard as a playing interface generate additional limitations that may harm the playability. To cope with this issue, a layer of control is added between the player and the model inputs. Thus, direct controls of the player are used to parameterize higher-level precompiled gestures for several playing modes (listed below).

*Attack* : at noteon, the evolution in time of the mouth pressure  $p_m$  is imposed and parameterized by the midi velocity. A very simple model of variation (attack/decay/sustain) has been chosen. A similar evolution of parameter  $\epsilon$  may also be imposed during the attack.

*Vibrato* : the lips tension is imposed (through the tuning parameter  $\omega_l$ ) by the mapping. However a vibrato may be generated by a sinusodal modulation of  $\omega_l$ . The amplitude of the modulation is zero during the attack, and after a delay starts to increase with time. Note that an additional modulation of noise amount (parameter  $\epsilon$ ) has also been retained.



*Legato* : when a key is pressed while at least one other key has not been released, it is supposed that the player wants the model to play legato. The time of transition between the two notes is made dependent on the midi velocity of the last note played.

*Humanization* : in order to avoid long-lasting notes to sound unnatural (because of fixed parameters), “humanization” was introduced for some parameters (typically  $\omega_l$  and  $p_m$ ) as random fluctuations around values given by the mapping or the player. Exaggerated fluctuations can be used to evoke the playing technique of a beginner.

## 5 Conclusion

The work presented here is part of the BRASS project whose aim is to propose to keyboardists virtual trumpets, trombones and saxophones based on physical modelling. Four virtual instruments are expected to play at the same time on recent personal computers (Apple or PC). The resulting software will be available in June/July 2005 and will be demonstrated at the conference.

This paper is focussed on the playability aspects of the virtual instruments. Therefore many aspects of the project are not even mentioned in this paper (see <http://www.arturia.com/en/brass/brass.php> to have an overview of the project; sound examples should be available soon).

## Acknowledgements

The authors would like to thank Xavier Rodet and Axel Röbel for providing the code developed in [13], and Lester Ingber for distributing the VFR as a free software package called ASA (<http://www.ingber.com/#ASA-CODE>). The BRASS project was funded by the RIAM program (<http://www.riam.org/riam/>).

## References

1. M. E. McIntyre, R. T. Schumacher, and J. Woodhouse. On the oscillations of musical instruments. *J. Acoust. Soc. Amer.*, 74:1325–1345, 1983.
2. C. Vergez and X. Rodet. Comparison of real trumpet playing, latex model of lips and computer model. In *Proceedings ICMC'97*, pages 180–187, Thessalonike, September 1997.
3. C. Vergez and X. Rodet. Trumpet and trumpet player : physical modeling in a musical context. In *Proceedings of the International Congress of Acoustics (ICA)*, page CDROM nIV, Rome, 2001.
4. C. Vergez and X. Rodet. New algorithm for nonlinear propagation of a sound wave. application to a physical model of a trumpet. *Journal of Signal Processing*, 4(1):79–87, January 2000. Special issue on nonlinear signal processing.
5. C. Vergez and X. Rodet. Air flow related improvements for basic physical models of brass instruments. In *Proceedings of ICMC'2000*, Berlin, German, August 2000.

6. A. Almeida, C. Vergez, R. Caussé, and X. Rodet. Physical model of an oboe: comparison with experiments. In *International Symposium on Musical Acoustics*, pages 112–115, Nara, Japan, Avril 2004.
7. K. Ishizaka and J.L. Flanagan. Synthesis of voiced sounds from a two-mass model of the vocal cords. Technical report, Bell. Syst. Techn. J., 1972.
8. H. Bailliet. Modelling of the French Horn-Player system. Master's thesis, INSA Toulouse, 1994.
9. W. J. Strong. Computer Simulation of a Trumpet. *J. Acoust. Soc. Amer.*, Suppl. 1(87):S138, 1990.
10. X. Rodet, P. Depalle, G. Fleury, and F.Lazarus. Modèles de signaux et modèles physiques d'instruments: études et comparaisons. In *Actes du Colloque Modèles Physiques de Grenoble*, 1990.
11. S. Adachi and M. Sato. Trumpet sound simulation using a two-dimensional lip vibration model. *J. Acoust. Soc. Amer.*, 99(2):1200–1209, February 1996.
12. H. Bouasse. *Instruments à vent*. Delgrave, 1929. ré-édité par Blanchard (Paris, 1986).
13. C. Yeh, A Röbel, and X. Rodet. Multiple fundamental frequency estimation of polyphonic music signals. In *Proceedings of ICASSP*, pages 225–228 (Vol. III), 2005.
14. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem.Phys.*, 21(6):1087–1092, 1953.
15. William H. Press, Saul A. Teukolsky, Xilliam T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2 edition, 1992.
16. L. Ingber. Very fast simulated re-annealing. *J. Math. Comput. Modelling*, 12:967–973, 1989.
17. G. Peeters, S. Mac Adams, and P. Herrera. Instrument sound description in the context of mpeg-7. In *Proceeding of ICMC*, Berlin, Germany, August 27<sup>th</sup>, September 1<sup>st</sup> 2000.