Wind Instrument Optimization Made Practical

Burton Patkau,[1†] *Antoine Lefebvre,*[2] *Edward Kort*[3]

[1]Vericom Systems Ltd.
[2]Développement Informatique Antoine Lefebvre
[3]Wood Wind Flutes
[†]bpatkau@vericomsystems.com

ABSTRACT

Mathematical modelling of wind instruments has progressed to the point where it is now feasible to model and optimize instrument designs on a personal computer. WIDesigner, a computer application for wind instrument design optimization, exploits several existing techniques to demonstrate this feasibility. Its current objective is laying out the bore and toneholes of a wind instrument to optimize intonation across a specified scale of notes. To make rapid impedance calculations, WIDesigner uses the Transfer Matrix Method on wind instrument bores, coupled with empirical models of instrument mouthpiece behaviour. The principal optimization algorithms used are BOBYQA for local optimization, and a custom variant of the DIRECT algorithm for global optimization.

1. INTRODUCTION

Historically, the design of wind instruments has depended on iterative improvement through trial-and-error changes to real instruments. Meanwhile, much academic work has gone into developing numerical predictive models of instrument components. This work has progressed to the point that researchers are assembling models to predict the performance of existing designs and create new designs optimized for performance.

This paper discusses key techniques used in one such effort, WIDesigner [1]. Although none of these techniques is novel, taken together they make it feasible to optimize the intonation of wind instruments on a personal computer:

- Transfer Matrix models in the frequency domain to keep computational requirements low.
- Empirical mouthpiece models to encapsulate the influence of the drive mechanism.
- Two optimization algorithms for high-performance local and global optimization.

WIDesigner is a Java application to assist the design of woodwind instruments. In its current manifestation, it calculates instrument dimensions to provide optimum tuning accuracy over the full playing range of an instrument.

2. OPTIMIZATION APPROACH IN WIDESIGNER

WIDesigner works with internal models of instruments and tunings, which can be saved as XML files. The instrument model includes the position and diameter of toneholes, the position of transitions in the instrument bore, and the internal diameter at these bore points. Each class of instrument also records features, specific to the class, of the sound-generation mechanism at the mouthpiece. The tuning model includes, for each note, the pattern for fingering toneholes, and the target frequency for the note.

To evaluate an instrument's tuning, WIDesigner finds a playing frequency close to the target frequency of each note, and calculates the tuning error between the playing frequency and the target frequency. It does not attempt to calculate impedances over broad frequency ranges, predict *a priori* playing frequencies, or even keep track of what register a playing frequency might fall into.

To optimize an instrument, WIDesigner minimizes a real-valued objective function over a constrained parameter space.

$$\text{minimize } F(\vec{x}), \text{ subject to } \vec{x} \in \mathbb{R}^n,\ l_i \le x_i \le u_i \qquad (1)$$

The parameter vector $\vec{x}$ maps to the instrument geometry. WIDesigner supports over 30 different mappings, from simple mappings such as the vector of tonehole diameters, to compound mappings that cover almost all dimensions of the instrument. Some mappings provide indirect views of the geometry; for example, tonehole spacing rather than absolute tonehole position, or bore diameter ratios rather than absolute bore diameters. These indirect mappings allow the constraints l_i and u_i to represent physical limitations more directly, such as the maximum space between a player's fingers, or the direction of a bore taper. The constraints allow the user to ensure that solutions are physically feasible, within the validity limits of the prediction model, physically usable, and perhaps even aesthetically satisfactory.

The function to be minimized, $F(\vec{x})$, could be based on reactance or reflectance, but we have found the most effective measure to be the sum of squares of tuning errors in cents over all notes in the tuning.

$$F(\vec{x}) = \sum_{i=1}^{m} (g(f_{p,i}(\vec{x}), f_{t,i}))^2 \qquad (2)$$

where $f_{p,i}(\vec{x})$ is the predicted playing frequency for note *i* with geometry $\vec{x}$, $f_{t,i}$ is the target playing frequency for note *i*, and $g(f_1, f_2)$ is the frequency difference in cents:

$$g(f_1, f_2) = 1200 \log_2(f_1/f_2) \qquad (3)$$

For any optimization, the user has full control over the constraint vectors, $[l_1 \quad l_2 \quad ...]$ and $[u_1 \quad u_2 \quad ...]$. WIDesigner can save constraint vectors as XML files for later re-use.

3. MODELLING THE BORE

WIDesigner uses the Transfer Matrix Method (TMM) to model the instrument's frequency response. A number of published articles discuss the use of transfer matrices [2-5]. In the context of optimization software, which computes the system's response thousands of times for each instrument fingering while the geometry is evolving, it is of primary importance that each evaluation of the system be as efficient as possible. The TMM meets this requirement.

WIDesigner starts with a $[\ P\ U\ ]^T$ state vector for the open termination of the instrument, and multiplies the state vector by the transfer matrix of each bore section and tonehole from the bottom of the instrument to the mouthpiece.

Given the $[\ P\ U\ ]^T$ state vector calculated for the bore, as seen by the mouthpiece, the mouthpiece calculator returns a $[\ P\ U\ ]^T$ state vector representing the state of the entire instrument as seen by the driving source. From this final state vector, WIDesigner calculates a final instrument impedance. Figure 1 illustrates the overall process.

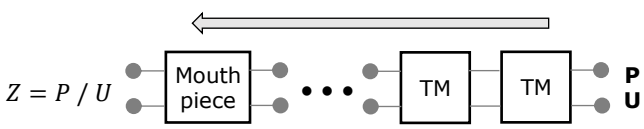

Figure 1. Instrument modelled as a cascade of 2-port networks.

WIDesigner uses the instrument impedance function to predict playing frequencies. Typically, the predicted frequency is the frequency closest to the target frequency at which the imaginary portion of $P\ /\ U$ is zero. Note that the impedance function at this point is not the raw bore impedance; it includes the effect of the drive source and instrument performance characteristics.

The transfer matrix of cylindrical and conical bore segments with boundary layer losses are formulated following Eq. (2) and (5) respectively from [6]. Toneholes use the results given in [7], principally those for unflanged toneholes in a thick pipe. The radiation impedance of the open-end, for the initial $[\ P\ U\ ]^T$ state vector, is calculated following [8] for unflanged tubes and tubes with a finite flange.

4. MODELLING THE MOUTHPIECE

4.1. General Approach

WIDesigner implements a different mouthpiece calculator for each class of instrument. At present, WIDesigner uses empirical mouthpiece models; each model has one or two calibration parameters. A model is calibrated by adjusting its parameters to give the best fit between WIDesigner's tuning predictions and measured tuning from a real instrument. WIDesigner supports calibration functions to assist the process; these are simply optimization functions for which the parameter vector $\vec{x}$ is the mouthpiece model's calibration parameters.

The expectation is that the calibration parameters depend only on the mouthpiece, so for example, calibration done on a whistle before drilling the toneholes would still apply as the toneholes were laid out.

4.2. Native American Flute (NAF) Mouthpiece Model

The NAF mouthpiece model is derived from the FluteCalc program that Dan Gordon developed for transverse flutes [9]. We have adjusted the FluteCalc formulas for use in NAF design, and added two empirical parameters – one static value and one parameter used for calibration.

4.3. Whistle and Flute Mouthpiece Model

The mouthpiece model for whistles and flutes acknowledges that an instrument can produce, for a given note, a range of pitches within one register depending on the blowing pressure. For each note, the model predicts both a maximum frequency, at which the instrument jumps to a higher register, and a minimum frequency, at which the instrument drops to a lower register or fails to sustain the note.

The mouthpiece model calculates an adjusted impedance for the overall instrument. The maximum frequency is that at which the imaginary portion of the impedance is zero. To predict the minimum frequency the mouthpiece model uses a model for loop gain [10] to calculate the frequency at which the gain drops below 1.

In performance, the actual playing frequency will be between the minimum and maximum. While a skilled performer can adjust the blowing pressure from note to note to maintain correct intonation, playing is more comfortable when there is a regular increase in blowing pressure going up the scale. The mouthpiece model predicts actual playing frequencies based on such a regular increase in blowing pressure.

For a transverse flute, the model treats the flute headspace as a closed tube in parallel with main bore, as seen from the embouchure hole.

4.4. Reed Mouthpiece Model

The NAF, whistle, and flute, mouthpiece models return a $[\ P\ U\ ]^T$ state vector, from which WIDesigner calculates an impedance. In contrast, the reed mouthpiece model returns a "normalized admittance" state vector, $[\ ZoU\ P\ ]^T$; the division yields a normalized admittance, ZoY instead of an impedance. This allows the remaining WIDesigner components to treat flute and reed instruments identically. Equation (4) gives the normalized admittance from the reed mouthpiece model, where $[\ P_b\ U_b\ ]^T$ is the state vector of the bore as seen by the mouthpiece.

$$ZoY\ = ZoU_b/P_b\ + j\Delta Y \tag{4}$$

The mouthpiece model calculates the admittance adjustment ΔY as a linear function of frequency. The slope and intercept of this function are the calibration parameters of the model. In general, we expect the adjustment will always be positive for inward-striking reeds (single and double cane reeds), and positive or negative for outward-striking reeds (lip reeds) [11]. In our experience with real instruments, the slope has always been positive for both classes of reeds. At this time, we expect that the calibration values may be different for different players, even with the same reed or mouthpiece.

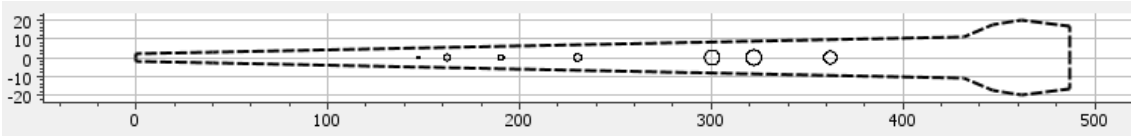

Figure 2. Hypothetical shawm design from WIDesigner

5. OPTIMIZERS

WIDesigner optimization requires optimization algorithms that are multi-variate, constrained, and derivative-free. Of the optimizers available to us, the two we have found most effective are BOBYQA [12] for local optimization, and a custom variant of DIRECT [13, 14] for global optimization.

5.1. BOBYQA

BOBYQA, Bounded Optimization by Quadratic Approximation, performs local optimization using quadratic approximation within a trust region. It serves as a fast algorithm for finding a local optimum, suitable for refining a design that is already reasonable. With BOBYQA, WIDesigner can optimize tonehole sizes and positions in a matter of seconds, testing on the order of 1000 instrument geometries.

5.2. DIRECT

DIRECT, DIviding RECTangles, is a more thorough algorithm for finding a global optimum in the whole search space. It starts with the whole search space between the constraint bounds, and divides it into successively smaller and smaller hyperrectangles. For each division, DIRECT divides a hyperrectangle in three along one of the dimensions, evaluating the function at the centre of each new hyperrectangle. At each iteration, DIRECT chooses hyperrectangles to divide that are large, or have low function values.

At present, WIDesigner uses a modified form of DIRECT:

- The optimizer will not divide hyperrectangles smaller than a specified minimum width. Once the best point found is in a hyperrectangle of the minimum size, the optimizer will return this point if there is no further improvement.
- Except for hypercubes, the optimizer divides a hyperrectangle on only one side in each iteration.
- Under some conditions, the optimizer uses an alternate strategy for selecting which hyperrectangles to divide in an iteration.

DIRECT is particularly useful when the optimization is not starting from a known good design, or when the search space has a number of local minima that can lead BOBYQA astray; this seems a particular problem when bore profile optimization is involved. For global optimization, WIDesigner runs DIRECT first, then uses BOBYQA to refine the solution found by DIRECT. This two-stage approach can optimize full instruments in a matter of minutes, testing tens of thousands of instrument geometries.

Figure 2 shows a sketch of the bore of a hypothetical two-octave shawm in D4, as optimized by WIDesigner. The shawm has 6 finger holes and a thumbhole, and plays 17 diatonic notes, from D4 through D6, plus cross-fingered flattened sevenths C5 and C6. WIDesigner optimized 19 dimensions of the instrument geometry, including the bore diameter at four points on the instrument bell, tuning all notes within 12 cents of Equal Temperament pitch. DIRECT tried about 17000 geometries, BOBYQA tried another 29000, in about 2 minutes on a 2.67GHz laptop CPU.

6. DESIGN AND CONSTRUCTION WITH WIDESIGNER

Design and construction of a new instrument still involves some iteration, even with the introduction of design optimization software such as WIDesigner. The optimization software helps the process converge more directly and reliably to the desired end: a well-tuned instrument. The steps below illustrate a typical process:

- Determine the target tuning, and select a draft design for the new instrument. Set the calibration parameters from an instrument with a similar mouthpiece.
- Produce optimized instrument designs with WIDesigner. Adjust the constraints to get a design that delivers the desired tuning accuracy, satisfies physical limitations of playability and aesthetics, and delivers the desired tone quality according to the designer's experience.
- Begin building the instrument. At intervals in construction, measure the current instrument geometry and tuning. Use these measurements to refine the WIDesigner calibration parameter and geometry values, and re-optimize the instrument design. For example, after crafting the mouthpiece and bore profile, measure the fundamental and one or more overtones, and use these to re-calibrate WIDesigner and re-optimize the tonehole size and location. After drilling undersized toneholes, measure the notes produced to check the calibration and re-optimize the tonehole size.
- Shape the mouthpiece and toneholes to produce the desired tone quality, and bring all the notes into tune. Final shaping proceeds based on measured frequencies rather than frequencies WIDesigner predicts. Tonehole adjustments may include aspects that WIDesigner's geometry model does not capture, such as elliptical toneholes, angled toneholes, or rounded edges.

7. NEXT STEPS

Clearly, the mouthpiece models would benefit from more data and research, to refine the models and ensure they accurately capture properties of real instruments. The current instrument model in WIDesigner is limited in what instruments it can describe: concert woodwinds require a description of keyed toneholes, and a Transfer Matrix model to go with it; brass instruments require description of valves, crooks and slides. Brass instruments would also benefit from enhancement of the

bore Transfer Matrix models to cope with bent crooks and widely flaring bells.

We also hope to investigate other aspects of instrument performance that we can model, such as timbre.

REFERENCES

[1] E. Kort, A. Lefebvre and B. Patkau, "Using WIDesigner," January 2016. [Online]. Available: https://github.com/edwardkort/WWIDesigner/wiki/Using-WIDesigner. [Accessed November 28, 2016].

[2] G. R. Plitnik and W. J. Strong, "Numerical method for calculating input impedances of the oboe.", *J. Acoust. Soc. Am.*, vol. 65, pp. 816-825, 1979.

[3] R. Caussé, J. Kergomard and X. Lurton, "Input impedance of brass musical instruments – Comparaison between experimental and numerical models," *J. Acoust. Soc. Am.*, vol. 75, pp. 241-254, 1984.

[4] D. Keefe, "Woodwind air column models," *J. Acoust. Soc. Am.*, vol. 88, pp. 35-51, 1990.

[5] D. Mapes-Riordan, "Horn modeling with conical and cylindrical transmission-line elements," *J. Audio Eng. Soc.*, vol. 41, pp. 471-482, 1993.

[6] A. Lefebvre, G. Scavone and J. Kergomard, "External tonehole interactions in woodwind instruments." *Acta Acustica united with Acustica*, vol. **99**, no. 6, pp. 975-985, 2013

[7] A. Lefebvre and G. P. Scavone, "Characterization of woodwind instrument toneholes with the finite element method," *J. Acoust. Soc. Am*., vol. 131, no. 4, April 2012.

[8] J.-P. Dalmont, C. J. Nederveen, and N. Joly, "Radiation impedance of tubes with different flanges: Numerical and experimental investigations", *J. Sound Vib*. vol. 244, no. 3, pp. 505–34, 2001.

[9] D. Gordon, "FluteCalc," personal communication, 2006.

[10] R. Auvray, B. Fabre and P.-Y. Lagree, "Regime change and oscillation thresholds in recorder-like instruments," *J. Acoust. Soc. Am.*, vol. 131, no. 2, pp. 1574-1585, February 2012.

[11] N. K. Fletcher, R. K. Silk, L. M. Douglas, "Acoustic admittance of air-driven reed generators," *Acoustica*, vol. 50, pp. 155-159, 1982.

[12] M. J. D. Powell, "The BOBYQA algorithm for bound constrained optimization without derivatives," Cambridge, England: Cambridge University, Department of Applied Mathematics and Theoretical Physics Technical Report NA2009/06, 2009.

[13] D. R. Jones, C. D. Perttunen and B. E. Stuckman, "Lipschitzian optimization without the Lipschitz constant," *J. Optimization Theory and Application*, vol. 79, no. 1, pp. 157-181, October 1993.

[14] S. G. Johnson, "The NLopt nonlinear-optimization package," [Online]. Available: http://ab-initio.mit.edu/wiki/index.php/NLopt_Algorithms#DIRECT_and_DIRECT-L. [Accessed 29 August 2016].