USING ACOUSTIC MODELLING TO DESIGN AND PRINT A MICROTONAL CLARINET

Nicholas J Bailey[1], Théo Cremel[2], Alex South[3]

[1] *Science and Music Research Group, The University of Glasgow*

[2] *Grenoble INP*

[3] *Scottish Clarinet Quartet*

Correspondence should be addressed to: nick@n-ism.org

Abstract: **Several projects have successfully used 3D-printing technology to produce viable musical instruments. Such instruments are normally replicas of existing ones with normal intonation. The authors have previously constructed microtonal rehearsal aids and musical instruments for the performance of 19-EDO music, and have compared the use of conventional clarinets using alternative fingerings and modification of pitch through embouchure with performances based on augmented wind controllers and synthesizers. Both approaches compromise the performance potential of the instrument in different ways. We now seek to create a new microtonal clarinet design by 3D-printing an instrument designed to render 19-EDO scales specifically. An object-oriented acoustic model of the instrument has been constructed in C++. A programmatic description of an arbitrary clarinet was written in OpenSCAD, a 3D modeling language, to permit physical realisation of instruments that were acoustically modeled. The software was initially used to predict the outcome of printing a replica Denner clarinet, and the results demonstrate a degree of agreement which was sufficient to suggest its use in designing a 19-EDO instrument. Strategies for the design of keywork on such an instrument are discussed and a candidate design printed and evaluated by a professional player.**

1. 19-EDO PLAYING WITH CONVENTIONAL CLARINETS

Intially, performance of 19-EDO pieces was undertaken using conventional clarinets with altered fingerings. Ingrid Pearson (Royal College of Music, London) is an authority on performance with early clarinets which considerably under-perform modern instruments in accuracy of intonation and demand greater effort from the player in considerably correcting the pitch produced. Players of such early instruments may therefore be expected to have a more developed facility in altering the natural intonation of the instrument. Dr Pearson developed an alternative fingering with the aid of the pitch-tracker facility of Rosegarden[1] and performed the second of three songs on Turkish texts[2] by the composer Graham Hair, “Wine”. The original fingerings were developed further by Alex South, who also recorded all three songs[3] using an acoustic instrument.

The use of a conventional instrument in the hands of an expert brings with it the possibilities of extremely fine musical expressivity arising from thousands of hours of practice. Requiring such performers to adopt a fingering scheme departing radically from the one conned so thoroughly that it is virtually a reflex is a significant cognitive demand when performing these works on conventional instruments. In the event, this proved to be a greater demand for the woodwind player than for the singer. We conjecture that string players may fall between the extremes, as the pitches “exist on the string” as they do within the compass of the voice, albeit at places which are not as regularly as explored as in 12-EDO music.

A clarinettist wishing to play 19-EDO music on the standard acoustic instrument (whether Böhm or Öhler system) must be prepared to learn a number of new fingerings. Because nineteen is a prime number, there is only one pitch per octave that has the same intonation as in 12-EDO (stipulated here to be a concert ‘A’.) For the other eighteen notes per octave, the desired pitch may only be achieved by using new fingerings combined with embouchure corrections (across the range of the clarinet, this amounts to a set of over sixty new fingerings). The absence of common pitches means that there is also an extra layer of complexity introduced in the reading of the score, above and beyond the difficulties in producing the notes with an acceptable sound and in moving from one pitch to another rapidly and smoothly: with the exception of the concert ‘A’s, every written note must be produced with one of the new fingerings. This requires considerable inhibition of existing scorereading skills, in addition to the acquisition of new fingering patterns.

Comparing 19-EDO with 24-EDO (a quartertone scale), it might be imagined that finding fingerings for the former would be the more straightforward problem. However, 24-EDO shares twelve pitches per octave with 12-EDO, and consequently ‘only’ twelve new fingerings per octave need to be found. In 24-EDO, furthermore, pitches notated in the ordinary way (as naturals, flats and sharps), have their standard fingerings and may be read without difficulty. It is helpful to conceptualize the seeking of new fingerings by starting from a ‘base-fingering’ for the nearest 12-EDO pitch. In some cases the solution was straightforward, involving only the addition of one or two extra keys to the base-fingering, (e.g. C') or an embouchure correction of a small fraction of a tone (e.g. A), although some of the resulting notes (e.g. $D\flat'$) are cross-fingerings (where a tone hole is closed ‘downstream’ of the first open tone hole), and tend to have a different timbre to ordinary fingerings (usually duller). In other cases (e.g. $D\flat''$) significant difficulties were experienced in finding an acceptable fingering, for example in regions of the clarinet where the notes are obtained via keywork which operates pads over holes a long way from the finger operating the key in question, rendering cross-fingerings impossible and resulting in almost impossible demands on embouchure flexibility. Pearson proposes the fingering shown in Figure 1a which naturally sounds a quartertone sharp to the desired pitch unless one of the little-finger keys is half-closed. Alternatively, those keys may be left open and the note fingered a 12-EDO semitone higher, the pitch being corrected through the use of embouchure, as in Figure 1b.

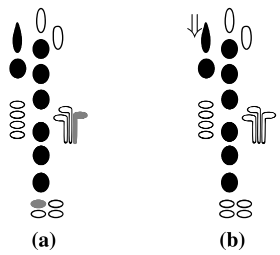


Figure 1: Alternative methods for producing a written 19-EDO $D\flat''$ on a Böhm system B♭ clarinet. Grey indicates a half-closed key.

Although it is true that

1. intonation-shifting for performances within the 12-EDO system is something which players do all the time for a number of reasons (e.g. temperature of the hall, position of note in a chord, having to play in unison with a piccolo),
2. new fingerings are required for players moving from modern

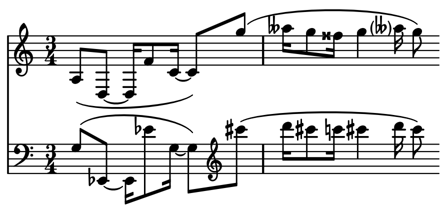


Figure 2: The openning two phrases from “Wine”, the first of Graham Hair’s “Three Songs from the Turkish”. The top stave shows the sounding notes; the bottom stave shows the notes required to obtain this result from a re-tuned MIDI synthesizer.

to ’period’ instruments, and

3. quartertones (and sometimes smaller divisions of the tone) are commonly called for in contemporary classical music,
4. this performer has found that the difficulties described above mean there is something peculiarly challenging about the use of new fingerings in 19-EDO, a challenge which would be augmented still further, of course, in higher prime-number temperaments.

2. Augmenting MIDI Controllers for 19-EDO

As well as the use of non-standard fingerings, it is also possible to perform microtonally using a re-tuned MIDI synthesizer. This is the method adopted for more recent performances of Graham Hair’s Turkish Songs and other 19-EDO works. In this case, the clarinetist plays Yamaha WX7 wind controller which provides a standard fingerings defined by the manufacturer. The possibility of using a MIDI synthesizer means that retuning becomes feasible, and increases the range of control possibilities by which to access the microtonal scale. Performance aspects of this approach are described in the documentary film “Making Music with Nineteen Tones”[4] and technical aspects in the can be found in the shorter ”Putting the Wind Up Pure Data”[5]. Using this technology it is possible to articulate the notes of the pieces more readily and with less cognitive load on the clarinetist. However, the MIDI wind controller, with a mere decade-or-two of development behind it, can not approach the degree of subtlety available from the acoustic instrument’s century-or-two’s development. There is also no established conservertoire training or performance tradition to compete with the acoustic clarinet’s.

The continuo player’s lot is not a much happier one. We know of no commercially available MIDI controller with keyboards like the 19-key-per-octave claviers which were more-or-less widely available prior to the establishment of Whol(erpobt)temperament. We have therefore developed a method of using a synthesizer tuned at hyperchromatic intervals controlled by a standard keyboard. The player is forced to read a kind of keyboard *scordatura* notation, where only A440 sounds as written. scordify19, a short program written in lex, transforms a string of lilypond score tokens so that playing the notes appearing on the score produces the desired performance. An example of how the scordatura version of the clarinet melody from “Wine” is shown in Figure 2. In 19-EDO, the A𝄫 at the beginning of the second bar is not an enharmonic equivalent of the preceding G♮, and is accordingly rendered as a different note in scordatura. Performing from such notation at the keyboard places demand on the player who, as well as now being required to play “wrong” notes, needs to span very wide intervals on the keyboard to produce a relatively narrower-sounding intervals.

The songs are presented on the n-ISM “Clarinet-ism” project page[6] where performances of the acoustic and MIDI wind instruments may be heard with the composer at the keyboard.

3. Simulation of Candidate Instruments using Acoustic Models

At the end of the last century, Perry Cook and Gary Scavone released The Synthesis Tooklkit (STK)[7] incorporating simple models of the reed developed by Julius O Smith as early as 1986,

```
WindModel::Wind *whistle;
constexpr double instrumentBore {0.02};

try {
    if (argc != 20) {
        cerr << "hotair: There must be ten pipe lengths "
            << "and nine hole radii\n";
        exit(1);
    }
    WindModel::Element **elements {new WindModel::Element*[22]};
    elements[0] = new WindModel::Reed;
    for (int a {1}; a < 18; a += 2) {
        elements[a] = new WindModel::Pipe(
            atof(argv[a]),
            instrumentBore
        );
        elements[a+1] = new WindModel::ToneHole(
            atof(argv[a+1]),
            instrumentBore
        );
    }
    elements[19] = new WindModel::Pipe(atof(argv[19]), instrumentBore);
    elements[20] = new WindModel::OpenEnd(instrumentBore);
    elements[21] = nullptr;
    whistle = new WindModel::Wind(elements);
} catch (WindModel::BadElementListException bee) {
  cout << "hotair: failed to construct wind model.\n"
        << bee.what() << endl;
  exit(2);
}
```

Figure 3: Code fragment from “hotair” which constructs a wind instrument model from command line arguments. The bore diameter is fixed, and the nineteen arguments specify (<pipe_length><tone_hole_diameter>){9}<pipe_length>

with a review of a more contemporary state-of-the-art appearing in [8].

The model presented here is inspired by the “blowhole” class of STK, which models a pipe with a two-port register hole and a three-port tonehole, and is terminated in an open end, although because other features (file-based and real-time control and audio output) of STK will not be required, STK is not a dependency. The program also draws inspiration from the work of Hanna Robertson who, while still a postgraduate student, modified the STK blowhole implementation to produce a feadóg (Irish whistle) model[9].

The code was largely refactored to make it more applicable to instrument optimization using evolutionary programming, producing the program “hotair” which implements a selection of Elements in C++, and it is now possible to construct an arbitrary instrument with tone holes of given diameters at given intervals. In the single-thread version of the simulator, this is achieved as shown in Figure 3.

Audio is generated by coupling a simple reed model to a sequence of components derived from a base class Element. Fundamental frequency of the instrument for a given fingering is the only result under consideration here. The elements used consist of toneholes which scatter the incident wave using a single pole and zero to model each one. The tone holes are interconnected with Pipe Elements which consist of a pure, non-integer delay constructed from a delay line and a first-order Thiran all-pass filter[10] which introduces a time delay of up to $\pm\frac{1}{2}$ sample. Finally, an Excitation (a Reed) is connected to one end of the instrument and a Termination (an OpenEnd) to the other end via Pipes.

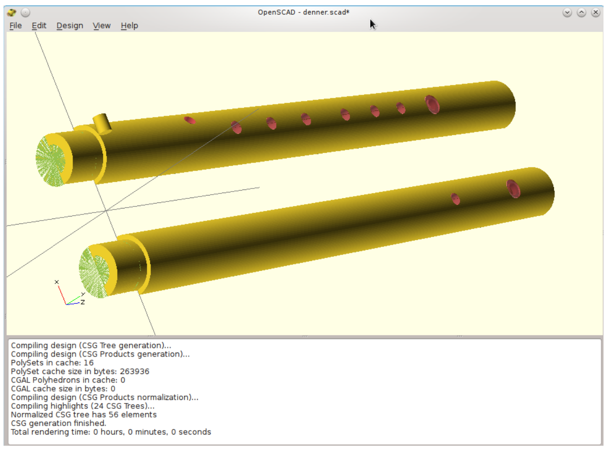
4. Construction of a Physical Instrument

The accuracy of the audio simulations thus obtained are satisfactory for the purpose of establishing instrument intonation as reported and demonstrated at a lecture-recital at the National Assiciation for Musical Instruments in Portugal conference, ANIMUSIC2014.

A parametric clarinet has been defined in OpenSCAD. The beginning of the main module, clarinet(), is shown in Figure 4. The remainder of the module produces the top and bottom halves of the instrument with the tone holes placed according the the list holeSpecs. Each entry in holeSpecs is at least the distance from the end of the instrument and diameter of the tone hole and the angle at which the hole should be made (0 being on the top of the instrument). An optional 4th parameter causes a circular rebate to be made surrounding the tone hole of the minumum depth required to ensure a flat surface over its entire area. This is used to make landing areas for the pads where it is necessary to close the hole with a key rather than the finger.

```
$fn=30;
module clarinet() {
// There is an extra 108mm length due to Alex's clarinet's bell
// Holes specification: height,diameter,angle[,pad-diameter]
holeSpecs = [
[ 68.6, 6.6, -30],
[ 98.6, 7.3,   0],
[127.0, 7.3,   0],
[156.2, 7.3,   0],
[190.5, 7.3,   0],
[217.5, 7.3,   0],
[242.0, 7.3,   0],
[256.0, 7.3, 180],
[272.9, 7.3,   0, 12],
[315.4, 7.3, 180, 12]
];
jointLength   = 432 - 75; // subtract Alex's mouthpiece
bore          = 15.3;
wallThickness = 8;
...
```

Figure 4: The beginning of the clarinet() module for the Denner-based clarinet used for initial intonation testing and model validation.



```
Compiling design (CSG Tree generation)...
Compiling design (CSG Products generation)...
PolySets in cache: 16
PolySet cache size in bytes: 263936
CGAL Polyhedrons in cache: 0
CGAL cache size in bytes: 0
Compiling design (CSG Products normalization)...
Compiling highlights (24 CSG Trees)...
Normalized CSG tree has 56 elements
CSG generation finished.
Total rendering time: 0 hours, 0 minutes, 0 seconds
```

Figure 5: The design of the clarinet shown in Figure 6 in the openSCAD preview window

Having rendered the openSCAD clarinet with the desired tone hole positions and inspected the model (Figure 5), it is printed and assembled (Figure 6).

5. Optimization using Evolutionary Programming

Optimizing a clarinet with a given number of tone holes with given sizes with regard a particular intonation has been attempted using Python and DEAP[11]. DEAP is an acronym for Distributed Evolutionary Algorithms in Python developed at the Laboratoire de Vision et Systèmes Numériques, Université Laval. Because the hotair program is written in C++, it is reasonably efficient in finding the best fingering set over a given instrument geometry. Although the fingerings produced may be counter-intuitive and not amenable to direct use in performance, one possibilty is in future to provide a mapping between a performance fingering and a production fingering mechanically, or indeed electromechanically. To this end, only fully-open and fully-closed toneholes are considered.

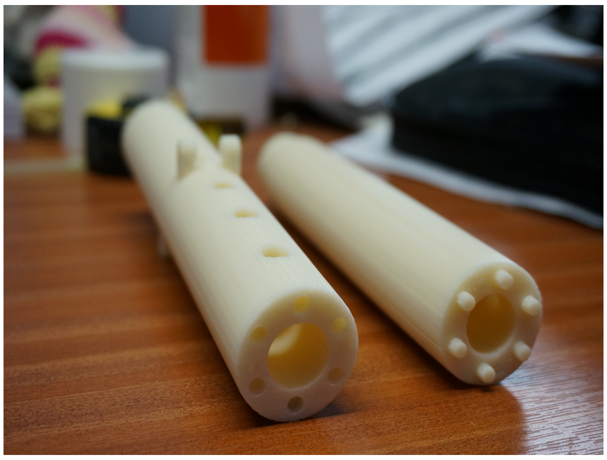


Figure 6: A 3D-printed Clarinet before having keys assembled. The instrument is constructed in 4 quarters split dorsally to facilitate the removal of filler material and transversely so that it can be produced by the small prototyping printer at The University of Glasgow's engineering workshop.

```
def evalClarinet(indv):
  command = ['./hotair']
  for i in range(19):
    scale = 0.02 if (i%2) else 0.5
    command += [ str(indv[i]*scale) ]
  try:
    simulation =
      subprocess.check_output(command).decode('utf8').split('\n')
    pitch_errors = [ float(l.split('\t')[2]) for l in simulation[:-1] ]
    lowest = float(simulation[0].split('\t')[0])
    abs_pitch_errors = numpy.abs(pitch_errors)
  except:
    lowest = 10000.0
    abs_pitch_errors = [10000.0]*19
  mean_error = numpy.sum(abs_pitch_errors)/19.0
  max_error = numpy.max(abs_pitch_errors)

  if isnan(mean_error): mean_error = 10000.0
  if isnan(max_error): max_error = 10000.0


  return mean_error, max_error, abs(lowest-110.0)
```

Figure 7: Code fragment from "optimise.py" evaluates the fitness of an instrument from the results of a hotair simulation

The target instrument for this study has a cylindrical bore and nine tone holes. Not all fingerings sound at all breath pressures, so hotair tests the 512 possible fingerings exhaustively by ramping the breath pressure between to specified values over a period of 10 seconds. For the majority of fingerings, sound is produced for most of that time; for some, the audio will essentially be zero-padded. The fundamental frequency is evaluated by a simple peak-location algorithm which consequently has a resolution of 0.1Hz.

Evaluation of each candidate instrument takes 206.9 real seconds on an Intel® Core^{TM} i3 CPU M350 at 2.27GHz using a single thread of execution. If a suitable cluster of computers are available, DEAP is designed to run these evaluations concurrently using SCOOP (Scalable COncurrent Operations in Python)[12].

optimize.py is the python script which makes use of the DEAP toolkit. The function which is used to evaluate the fitness of a candidate instrument is shown in Figure 7. An individual is represented as a list of 19 random floating point numbers initially chosen randomly in the range [0,1). This is conveniently generated natively by the DEAP toolkit, so rather than provide a customised creation function, they are instead scaled appropriately before being passed to the hotair program for simulation. Acoustic distance between elements (double the physical distance because the wave front traverses forward then in reflection) are scaled differently from the tone hole sizes.

The result of a single run of the hotair program for a randomly chosen geometry is shown in Figure 8. The first column represents the target frequency, calculated by raising the lowest (all tone holes covered) frequency by 2^{19} for each scale step. The second column is a binary representation of the fingering for which the model gives the nearest pitch to the target, and the third is the error from the target pitch in cents.

```
80.8        000000000    0
83.8021     000011111    -63.1579
...
150.229     101000000    -4.9517
155.811     111000000    -25.8684
```

Figure 8: Output from the hotair program. For each of the 19 divisions of the octave, a fingering pattern is produced which most nearly matches the desired frequency under simulation.

Python's subprocess module is used to invoke the corresponding simulation command, and the resulting output parsed. It could be that the simulation fails, or produces nan for any or all of its outputs. In this case, the fitness is recorded as a large number. The mean error and maximum error across all fingerings is calculated as well as as the deviation from 110Hz for the lowest note. These are returned as a 3-tuple and form the individual's 'fitness' (actually, unfitness, since DEAP has been required to minimize these quantities).

The evolutionary strategy used is $(\mu + \lambda)$ BLX (blend crossover)[13] with $\mu = 30$ and $\lambda = 70$. Mutations occur with probability 0.2 and have $\sigma = 1.0$.

DEAP optionally maintains a "Hall of Fame" in which are recorded the best-performing instruments whether or not they survive in the current population. Currently the best candidate has an average pitch error of 2.3 cents, and a worst-case pitch error of 7.2 cents, although the fingerings and tone hole disposition to achieve this are somewhat eccentric.

6. Conclusion and Future Work

Fingerings suggested as optimal by hotair take only intonational accuracy in to consideration, and ignore the practicality of performing the works on the physical manifestation of the instrument. This is a consideration because of counter-intuitive or even physically impractical fingering patterns, combined with the requirement always to be able to move smoothly between pitches. In reality, either a mechanical system of keys could be devised, or more likely, an electromechanical coupling between the player's fingers and the tone holes could be introduced.

In addition to the tightly constrained instrument geometries discussed in this paper, it may be feasible to consider alternative designs, or even evolve the designs themselves. For example, the modern clarinet has a polycylindrical bore rather than a simple cylinder, primarily to improve intonation when overblown. Without the modification to the cylindical profile, the perfect twelfth between the lower and upper registers varies according to the sounding length of the tube.

Since the commencement of this project, we note that a far more sophisticated modelling system, ART (Acoustic Research Tool), has been brought to an advanced stage of completion[14]. ART has the capacity to produce models from textual descriptions of the instrument including many and varied bore profiles, alternative tone hole models, and recognises visco-thermal losses. With the possibility of state-space modelling of diverse geometies, and the consequent direct measurement of the instruments' intonation, it is conceiveable that completely novel geometries may arise. Because direct solution in state-space yields a frequency response very easily, it is likely that a far more sophisticated fitness function could be deployed, incorporating fingering heuristics.

It is a happy coincidence that the perfect twelfth by which a tube with a single open end overblows, is divided into 19 equal semitones in 12-EDO. This suggest the possibility of creating instruments which overblow at the octave (like the saxophone which has a conical bore), but use a modified version of a standard clarinet keywork thus providing 19 equal divisions of the octave. Whether the retention of existing fingering patterns with differing pitch associations is a help or a hindrance is something which can only be determined with the help of practitioners.

Our final conclusion of our feasibility study into evolving the clarinet is that, while even quite simple time-domain models can produce pitch-accurate simulations of real instruments, the computational effort remains large. Furthermore, care must be exercised in framing the cost function. Because the optimization was required to minimize the pitch error without regard to any other parameter, many of the candidate instruments generated by these simulations were 'overoptimized' in the sense that very small holes were added in an attempt to reduce the intonation error resulting in arbitrary, counter-intuitive and essentially unperformable fingering combinations. In reality, an error of a few cents can be easily compensated by a professional player by use of embouchure and vocal tract adjustments. Modelling real instruments with a view to construction should combine fingering heuristics and non-cylindrical bores into their fitness functions.

More positively, it has proven relatively easy to produce a playable instrument parametrically using a 3D printer. With the addition of purely mechanical or even electomechanical keywork, and an improved model of the kind offered by ART inter alia, we expect to be able to produce a 19-EDO instrument suitable for performance which will be more satisfactory than the synthesizer/wind controller combination.

References

[1] G. Hair, I. Pearson, A. Morrison, N. Bailey, D. McGilvray, and R. Parncutt: *The Rosegarden Codicil: Rehearsing Music in Nineteen-Tone Equal Temperament*. In *Scottish Music Review*, volume 1(1):25, 2006.

[2] G. Hair: *Three Songs from the Turkish*. Musical Score (PDF file): http://www.n-ism.org/Scores/Graham/Micro.pdf.

[3] G. Hair: *Songs from the Turkish*. Ravello Records RR7877, 2014.

[4] G. Hair: *Making Music with Nineteen Tones (Tom Majerski)*. Video: https://vimeo.com/80450876.

[5] N. J. Bailey: *Putting the Wind up Pure Data*. Video: https://vimeo.com/77621987.

[6] G. Hair: *International Clarinet Mixed Consort Project*. Web page: http://www.n-ism.org/Projects/clarinetism.php.

[7] P. R. Cook and G. Scavone: *The synthesis toolkit (stk)*. In *Proceedings of the International Computer Music Conference*, pages 164–166. 1999.

[8] J. O. Smith: *Virtual acoustic musical instruments: Review and update*. In *Journal of New Music Research*, volume 33(3):283–304, 2004.

[9] H. Robertson: *Mimicking the fipple sound in STK*. Web page: http://www.music.mcgill.ca/ hannah/MUMT307/307project.html, 2012.

[10] T. I. Laakso, V. Valimaki, M. Karjalainen, and U. K. Laine: *Splitting the unit delay [FIR/all pass filters design]*. In *Signal Processing Magazine, IEEE*, volume 13(1):30–60, 1996.

[11] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné: *DEAP: Evolutionary Algorithms Made Easy*. In *Journal of Machine Learning Research*, volume 13:2171–2175, 2012.

[12] Y. Hold-Geoffroy, O. Gagnon, and M. Parizeau: *Once you SCOOP, no need to fork*. In *Proceedings of the 2014 Annual Conference on Extreme Science and Engineering Discovery Environment*, page 60. ACM, 2014.

[13] L. J. Eshelman and J. D. Schaffer: *Foundations of Genetic Algorithms*, volume 2, chapter Real-Coded Genetic Algorithms and Interval-Schemata, pages pp. 187–202. Morgan Kaufman Publishers, San Mateo, 1993.

[14] C. B. Geyer: *Time-domain Simulation of Brass and Woodwind Instruments*. Master's thesis, Universität für Musik und darstellende Kunst Wien, 2012.