R-FF: A Single Reed Haptic Library for the TorqueTuner

Maxwell Gentili-Morin
IDMIL/CIRMMT/McGill University
Montreal, Quebec, Canada
maxwell.gentili-morin@mail.mcgill.ca

Marcelo M. Wanderley
IDMIL/CIRMMT/McGill University
Montreal, Quebec, Canada
marcelo.wanderley@mcgill.com

ABSTRACT

R-FF (Reed Force Feedback) is a library of clarinet reed equations adapted for use on the TorqueTuner (TT), a one degree of freedom, rotary force-feedback (FF), haptic device. We created the R-FF library by analyzing the clarinet's physical model and adapting the reed component that defines the forces between the user and the mouthpiece into force-feedback models. The models were designed for implementation on small and portable micro-controllers, like the ESP-32 module, found in the TorqueTuner(TT). Accompanied by a clarinet sound model, results show that the R-FF library could grow into a platform that allows non-wind players to explore the performance space of a single reed instrument, feeling the dynamics of the reed-embouchure system with their fingers instead of with their mouth.

Author Keywords

Haptics, force-feedback,Physical modeling, reed, educational tools

CCS Concepts

•Applied computing → Sound and music computing; •Human-centered computing → Haptic devices; •Computing methodologies → Physical simulation;

1. INTRODUCTION

Physical simulations of instruments and sounds like a string or clarinet as the term suggests, implements the equations that determine how the system functions into a digital environment for a variate of applications, like the acoustic modeling of a saxophone mouthpiece for better designs[16]. One in specific is in the creation of haptically enable digital musical instruments (DMI) that let the user control the physical model itself. It would allow the user to not only hear what they play but also receive feedback directly from the system, restoring lost feedback channel in DMIs[10]. Once we separated the interface from the sound by no longer using acoustic bodies, so did we cut out the force and vibrotactile feedback that came from interacting with the instrument. By combining physical simulations with haptic interfaces, we are starting as a community to bring back that channel of information. The question is how we achieve this for physical models of woodwind instrument like the clarinet as the waveguide models for string have already been done as seen in Onofrei et al.[9].

We explored this avenue by creating the R-FF library. A set of equations that originate from the clarinet physical model. To test them, we implemented them onto a one degree of freedom (DoF), rotary haptic device called the TorqueTuner. The library is accompanied by a set of sound models, designed in Max MSP, and a Python based user interface. The equations calculate the pressure and flow values produced by a wind instrument's single reed. The goal was to explore and experiment with these force models over multiple iterations and produce a new set of effects for the TorqueTuner. All code and hardware are open-source and meant to be reproducible.

1.1 Mapping the Gestures of the clarinet

The choice to go with clarinets also lies in its use in gesture analysis. For example, the analysis of the airflow across the reed as a gesture by Rovan et al.[11]. They implemented the airflow model derived from a data set of clarinet recordings as a timbral gesture on a midi wind controller.

1.2 Woodwind Single Reed Model

Thanks to the vast amount of research done on the clarinet and on physical modeling of woodwind instruments[13], we were capable of adapting the human-reed interactions inherent to woodwinds into haptic models. Generally speaking, the woodwind model is composed of the reed, bore, and bell components[12]. The reed can be considered a pressure-controlled excitor that inputs an amount of pressure into the instrument. That pressure then gets converted into acoustic energy within the bore and outputs a sound whose note and timbre is dependent on the dimensions of the pipe. In our case, we are only focusing on the reed model and do not need to implement the bore and bell portions. This means that the equations to be focused on are the Reed Flow equation and the solution to determining the pressure difference at the opening of the mouthpiece[6]. The Reed Flow equation determines the quantity of air flow (u) through the reed and proportionately determines how loud the resulting sound is. It utilizes the Bernoulli equation to represent the difference in pressure between the mouth and the mouthpiece cavities and Hooke's law to calculate the reed tip opening.

$$u = wx\left(\frac{2|p\Delta|}{\rho}\right)^{\frac{1}{2}} sgn(p\Delta) \tag{1}$$

NIME'24, 4–6 September, Utrecht, The Netherlands.

$$x = H - \frac{A_r p\Delta}{k} \quad (2)$$

Where w is the width of the reed channel, $p\Delta$ is the pressure difference between the downstream and upstream pressures, ρ is the fluid density constant, and A_r being the Surface area of the exposed reed and k being the reed stiffness constant. Most variables here are only symbolically present as most units are not taken considered in the final equation due to it being normalized. Now combining the two equations, we get:

$$u = wH\left(1 - \frac{p\Delta}{p_c}\right)\left(\frac{2p\Delta}{\rho}\right)^{\frac{1}{2}} sgn(p\Delta) \quad (3)$$

Where p_c is the required pressure to close the reed completely, H is the reed tip equilibrium, and w is the width of the reed channel.

$$p_c = \frac{kH}{A_r}$$

Next, the pressure difference is a set of solutions calculated by taking the intersection of the pressure difference $(p_\Delta^- - p_\Delta)$ and the reed flow (u) for different values of outgoing downstream pressure (p_Δ^-).

$$Z_c u = p_\Delta^- - p_\Delta \quad (4)$$

$$Z_c\left(wH\left(1 - \frac{p\Delta}{p_c}\right)\left(\frac{2p\Delta}{\rho}\right)^{\frac{1}{2}}\right) sgn(p\Delta) = p_\Delta^- - p_\Delta. \quad (5)$$

1.3 TorqueTuner

The TorqueTuner (TT) is a 1-DoF haptic knob with a commercially available Moteus controller and brushless motor being controlled by an ESP32 microcontroller. More specifically, the TinyPico ESP32 microcontroller. ESP32 microcontrollers feature a 32-bit, dual-core microprocessor that is capable of Wifi and Bluetooth, perfect for IoT devices and in this case, of untethering a DMI from a computer. from The TT was originally designed to map sensor data from associated digital musical instruments to haptic effect parameters found on the TorqueTuner [5]. At that stage, the hardware was meant to operate in a more portable solution and thus does not exert as much torque as the current version. Even before the TT, there were tests done using haptic 1-DoF motors such as by Snibbe et al.[14] where they studied haptic gestures for controlling device. Haptic effect authoring tools also exist that are usable with the TT, for example, Feelix[15] and ForceHost [2]. There is also a wealth of related work discussing the numerous, current challenges facing force-feedback, such as affordability and replicability[3]; two issues that the TT and this author attempt to address.

2. IMPLEMENTATION

To develop the R-FF library, we employed pre-existing platforms in order to minimize prototyping and focus on designing the haptic models. The platforms used were Python's matplotlib and numpy, for calculating and displaying the equations; MAX/MSP, for the sound generation; and the TorqueTuner's C++ haptic development environment. We designed the haptic library within Python and, in iterations, ported the finished equations over to the TorqueTuner's ESP-32 coding environment.

2.1 Generating the Haptic Library

Taking equations 3 and 4, we identified the essential parameters (H and k) and normalized the inputs and outputs of the equations so that it is suitable for use as a haptic effect.

$$u = \left(H\sqrt{|p\Delta|} - \frac{|p\Delta|^{\frac{3}{2}}}{k}\right) sgn(p\Delta) \quad (6)$$

When implementing the pressure difference solution, we chose to pre-calculate it in Python and output a table of values. This table was then uploaded with the TorqueTuner firmware to be used as an effect.

$$Z_c u = p_\Delta^- - p_\Delta$$

$$Z_c\left(H\sqrt{|p\Delta|} - \frac{|p\Delta|^{\frac{3}{2}}}{k}\right) sgn(p\Delta) = p_\Delta^- - p_\Delta.$$

2.2 The Three Iterations

As seen in Figure.1, the initial haptic effect we added to the R-FF library was the flow equation of the clarinet reed. With this effect, a midi keyboard was used in tandem for controlling the pitch, and the sound model was a clarinet patch in the open-source audio plugin, called Surge XT. The TorqueTuner with the new air-flow haptic effect acted basically like a force-feedback capable modulation wheel. Using OSC, the TorqueTuner controlled the amplitude value of the Surge XT patch with the same value it was outputting for the haptic effect.

The reed-flow haptic effect was then improved upon by creating three zones, as shown in Figure.2, which interact with the sound model in different manners. Users would move between the three zones when interacting with the flow model yielding different qualities based on equivalent effects on the clarinet. The first zone reflects the regular sound result coming from the clarinet whilst zone two and three represented tongue slapping and overblowing respectively. These three zones were made possible with a set of mappings (Table.1) that linked key values generated in the reed effect to parameters in the sound model. The OSC messages coming from the TorqueTuner were fed into a MAX/MSP patch where they were conditioned and then sent to a EuroRack system via control voltage (CV) as seen in Figure.5.

After receiving feedback on this improved iteration, a visual portion was created within Python that leveraged the work we did in that environment. We added the pressure value equation to the R-FF library, replaced the Eurorack system with the compiled MAX/MSP clarinet Physical Model version found in FAUST (Functional AUdio Stream)[1], and added a force sensitive resistor (FSR). On wind instruments, the pressure response is felt when blowing into the instrument. As such, a more accurate representation of the person-mouthpiece interaction is achieved in the haptic model with the inclusion of the pressure effect. The pressure mode on the TorqueTuner was created to house the respective equation. When in pressure mode, the user can feel and interact with the pressure equation. The pressure and the flow equation are active in said mode and their output are sent into a Max patch to be mapped to the pressure input and the amplitude input of the Clarinet Physical Model. To add an extra dimension of interaction with the model, an FSR was added. In its current state, the FSR is used to represent wind instrument articulations. By applying pressure on the FSR, like placing the tongue on the

[1] https://faust.grame.fr/

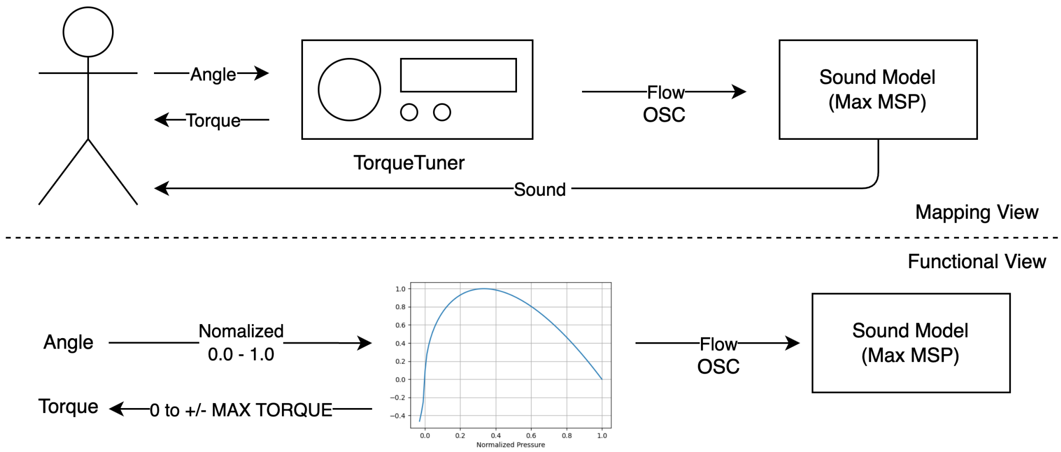

Figure 1: A graphical representation of the first iteration of the haptic reed model on the TorqueTuner. It is subdivided into its mapping representation, where there is a focus on the values being transmitted and the functional representation[8].

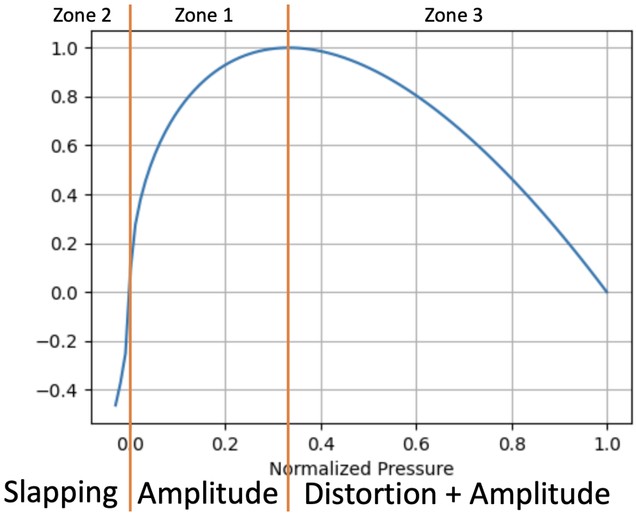

Figure 2: The multi-modal version of the flow equation used for the second iteration of the reed model. Each zone corresponds to a different effect on the sound model.

reed to cut the pressure and airflow in, it proportionally dampens the pressure and airflow value being sent to the clarinet model but sends to the motor an undamped torque value. In other words, a linear value of torque based on how much the motor turns, representing the closing of the reeds opening. This dichotomy is done in firmware and in the pressure mode.

2.3 Communicating between Components

The protocol used as mentioned above was the Open Sound Control (OSC) protocol. We used the integrated Puara framework[7] on the TorqueTuner to transmit all data from the TorqueTuner to both the Python visual component and the Max patch. The values sent are found in Table.1.

2.4 Visual Component

With the advent of the third iteration, we provide a python based graphical user interface to link the haptic effects felt on the TorqueTuner to a visual representation. The users can also modify the exposed variables H and k of the TorqueTuner's haptic effect right from the interface over OSC.

2.5 Generating Mappings from the TT to the Sound Models

Our goal with the mappings between the TorqueTuner and the Max patch was to closely mimic certain effects found on a clarinet/wind instrument. In the second implementation, we focused on 3 clarinet modalities as seen in Figure.2: Tongue slapping (zone 2), regular blowing (zone 1), and over-blowing (zone 3). Table two and Figure.5 describes the OSC parameters and how they are mapped from the TT to the Eurorack module. The `\flow` directly controls the amplitude of the sound, while the `\pluck` sends a quick attack envelope to the sound source, producing the equivalent of a tongue slap on the clarinet. The `\Angle Out` in turn is used to determine when to increase the distortion that one would normally hear in a clarinet when they start to over-blow into the instrument.

In the third implementation, we focused on tongue attacks and the resulting pressure inequality between the mouth and the mouthpiece when the tongue is pressed against the reed. The mappings created were done in firmware. The FSR reading proportionally increases the torque to its max on the TT but send an inversely proportioned pressure reading to the max patch, essentially acting as the inequality barrier.

3. RESULTS

During the three stages of development, several peer were given the opportunity to demo the new R-FF library on the TorqueTuner. Most found the effects and the interaction with the sound models to be well chosen and interconnected. The haptic models on the TorqueTuner and interfaces also proved to be stable with no issues present during the demo. During the demos, some were found to experiment with the bounds of the effects. By pushing the effects to their limits an harmonic oscillation due to aliasing in the TT's haptics update rate caused interesting sonic results. The users wanted to understand the correlation between what they were feeling on the TorqueTuner and what they were hearing. As such, it would suggest that the R-FF library, with further development, could become a tool for people to better understand how the reed-mouthpiece system may affect the sound.

Table 1: Set of mappings from OSC to Sound Characteristics

OSC Message	Type	Destination
\pluck	one to one	percussive envelope
\Minimum Torque	One to One	above envelopes amplitude
\Flow	One to one and Divergent	Amplitude/Sustain
\Angle Out	One to one	Harmonic Distortion
\Button	One to one	Note Advancement

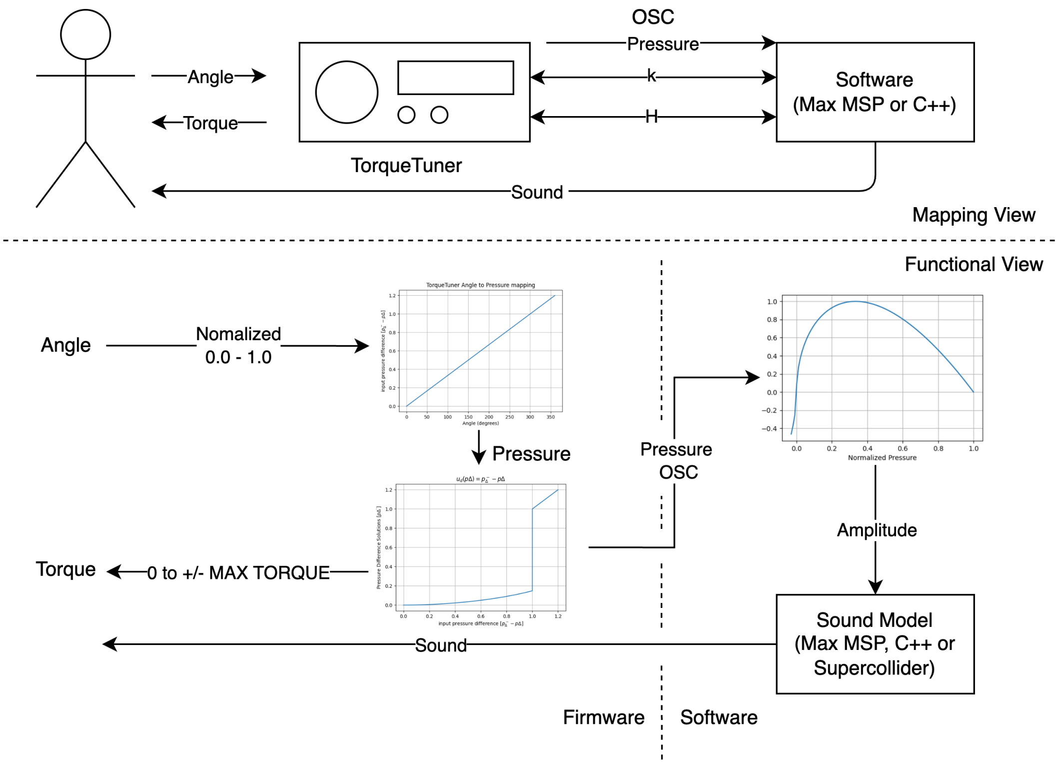

Figure 3: This is the general view of the reed models third implementation.

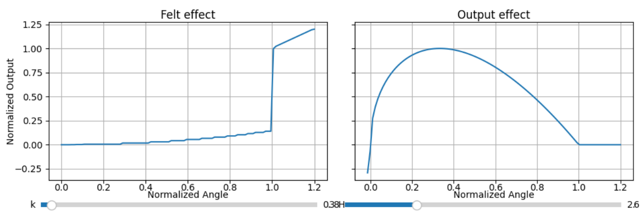

Figure 4: The 'felt effect' is what users received as haptic force feedback while the 'output effect' is what was sent to the sound module for processing, usually applied to the amplitude envelope. The H value is the Reed Tip opening variable, and the k value is the reed stiffness.

4. DISCUSSION

Most implementations of haptics in music controllers use a string synthesis methods, like the Karplus–Strong, or variations on the spring system, which the controller would directly interact with. This can be seen in Berdahl et al.[1], where the authors demonstrate the basic principle of attaching a haptic device to a simple spring system in Pure Data. By moving the device, the position, velocity, and acceleration can be used to generate the force values felt and heard. Normally, such interfaces would require a multi-DoF haptic device to replicate the gestures that are involved when striking or brushing a string. When it comes to studying and implementing a physical system like that of a woodwind, the gesture vocabulary of most commercially available haptic devices does not work. It is possible, however, to transpose the gestures between the mouth and mouthpiece like blowing, over-blowing, and tongue slapping onto a one degrees of freedom device, like the TorqueTuner, by focusing on the mappings as we did in the implementations. This is especially true as a one to two DoF interaction space is better suited for woodwind gestures versus with three DoF. Blowing is more of a direct motion and tonguing predominantly operates perpendicular to the reed, reducing the need for a third degree.

5. FUTURE WORK

The R-FF library has, over the course of the project, proven to be a potential tool for beginners and non-wind instrument players to help them understand the underlying concepts, like the importance of blowing, and how the reed's stiffness or biting down too much on the mouthpiece may affect the playability of the instrument. Having a tool that externalizes the forces felt when blowing into the instrument can benefit beginners who are initially overwhelmed with everything, they need to simultaneously pay attention to, like finger placement on the clarinet, making sure to blow enough into the instrument, not puffing out the cheeks, etc.

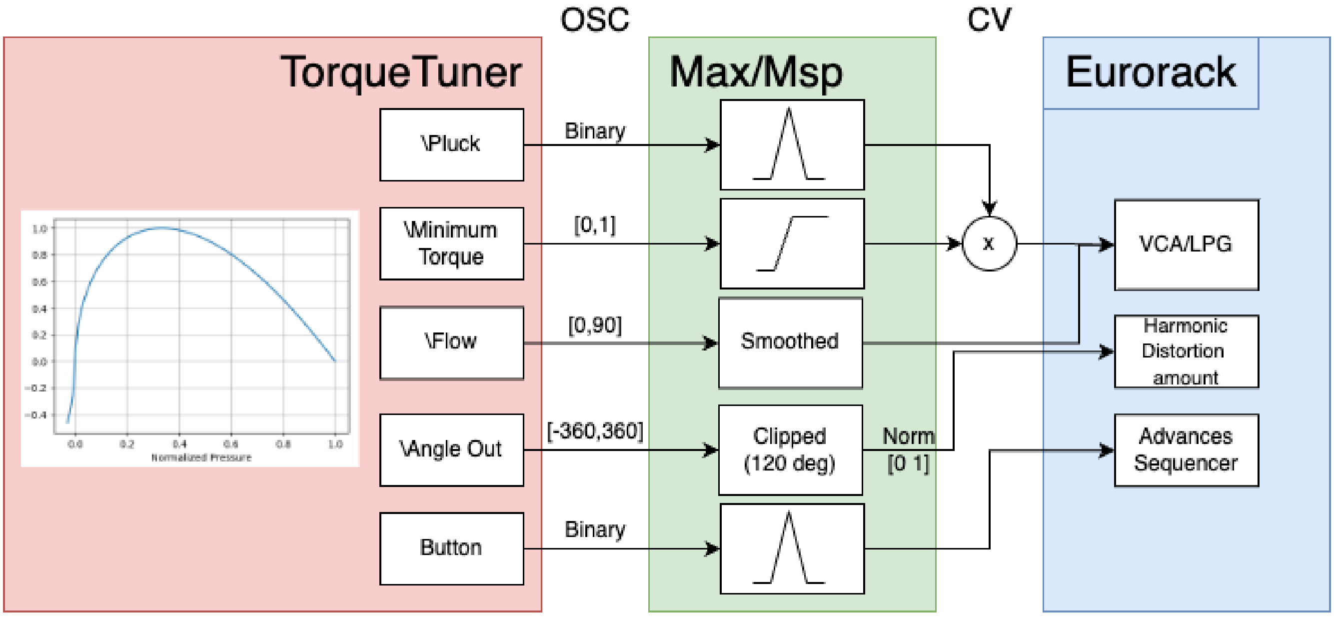

Figure 5: This is the mappings created between the TorqueTuner, the Max MSP patch and the Eurorack Module.

A beginner clarinetist must acquire a good and consistent embouchure and intonation when practicing, striking a balance between sound quality and techniques. From a study by King[4], it was found that clarinet students from 42 Ohio bands were most challenged by register changes, tongue placement, fingering, and pitch/intonation in that order. King also lists what the band instructors focus the most on when in rehearsal, that being the embouchure. What if the R-FF library can be used to address the listed core challenges that student clarinetists face like the register change, tongue placement and intonation. It can also be beneficial to reduce the required time for working on the embouchure during rehearsal. Students can spend time practicing certain parts of their pieces with the R-FF instead of a clarinet. This would allow students to segment their practice from having to focus on their fingering technique and their embouchure to just the embouchure. If not, it can be used as a tool to teach students the core concepts of the embouchure so that they know what to expect when interacting with the clarinet; on average, mistakes made early on when learning an instrument tends to carry on if not quickly addressed.

Moving forward, we shall continue to develop the R-FF library by adding new haptic effects that explores the gesture space of the core musical techniques beginner clarinetists find most difficult. To do so, data collection on the musician's embouchure would beneficial to better understand the interaction systems that have up till now only been modeled. The equations parameters have not been explored as a result. Also, we shall address the aliasing observed when interacting with the TorqueTuner and increase the sampling rate of the haptic effects.

6. CONCLUSION

Over the course of the project, we conducted research into the application of acoustic physical models in haptics creating the Reed Force Feedback (R-FF) library and implemented them into the open-source haptic dev-tool, called the TorqueTuner. The work underwent three distinct iteration and received feedback during its demos that aided in its development. Closing on the end of development we found that the R-FF library can with more work become a tool to help with learning the more subtle aspects in playing a wind-instrument. We also will continue to improve the latency on the effects in order to reduce any fluctuations.

7. ACKNOWLEDGEMENTS

Thanks to Gary Scavone and Champ Darabundit for their help with the reed and flow equations, Bavo Van Kerrebroeck and Colin Raab for their feedback, Ziyue Piao for her guidance in the development of the TorqueTuner firmware, Emil and Travis West, Kasey Pocius and João Tragtenberg for their help and support while writing this paper. Part of this research was funded by an NSERC (Natural Sciences and Engineering Research Council of Canada) Discovery Grant from the second author.

8. ETHICAL STATEMENT

The authors do not recognize any potential conflicts of interest in this research project. All users consented to their feedback appearing in the paper and were known peers who were curious to try the haptic effects out, and helped with the project's quality assurance. Having been enormously inspired by various projects made in the NIME community, we think it's important to share this ongoing work with an open-source license. The use of accessible hardware, and 3D-printed parts should make this project more replicable. All source code and documentation to reproduce the instrument are made available via this public GitHub repository. We encourage people to use, share and modify this work to fit into their own creative practices.

9. ENVIRONMENTAL STATEMENT

We are conscientious of reducing our electrical waste by making use of the hardware available to us. The TorqueTuner is a project that utilizes recyclable material when possible like the 3D printed enclosure. We do acknowledge that the production of the material used to create the electronics have a permanent impact on the environment and the communities involved.

10. REFERENCES

[1] E. Berdahl, G. Niemeyer, and J. O. Smith. Using Haptic Devices to Interface Directly with Digital Waveguide-Based Musical Instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 183–186, Pittsburgh, PA, 2009.

[2] C. Frisson, M. Kirkegaard, T. Pietrzak, and M. M. Wanderley. ForceHost: an Open-Source Toolchain for Generating Firmware Embedding the Authoring and Rendering of Audio and Force-Feedback Haptics. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2022. https://nime.pubpub.org/pub/jtdpakvp.

[3] C. Frisson and M. M. Wanderley. Challenges and Opportunities of Force Feedback in Music. *Arts*, 12(4):147, Aug. 2023.

[4] R. King. Clarinet Pedagogy: Common Challenges and Solutions. page 771, 2018.

[5] M. S. Kirkegaard, M. Bredholt, C. Frisson, and M. Wanderley. TorqueTuner: A Self Contained Module for Designing Rotary Haptic Force Feedback for Digital Musical Instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 273–278, Birmingham, UK, 2020.

[6] M. McIntyre, R. Schumacher, and J. Woodhouse. On the Oscillations of Musical Instruments. *Journal of The Acoustical Society of America*, 74:1325–1345, 11 1983.

[7] E. A. L. Meneses, T. Piquet, J. Noble, and M. Wanderley. The Puara Framework: Hiding Complexity and Modularity for Reproducibility and Usability in NIMEs. pages 86–93, May 2023.

[8] D. V. Nort, M. M. Wanderley, and P. Depalle. Mapping Control Structures for Sound Synthesis: Functional and Topological Perspectives. *Computer Music Journal*, 38(3):6–22, 2014.

[9] M. G. Onofrei, F. Fontana, and S. Serafin. Perceptual Relevance of Haptic Feedback during Virtual Plucking, Bowing and Rubbing of Physically-Based Musical Resonators. *Arts*, 12(4), 2023.

[10] J. Rovan and V. Hayward. Typology of Tactile Sounds and their Synthesis in Gesture-Driven Computer Music Performance. *Editions IRCAM*, 01 2000.

[11] J. B. Rovan, M. M. Wanderley, S. Dubnov, and P. Depalle. Instrumental Gestural Mapping Strategies as Expressivity Determinants in Computer Music Performance. In *Kansei, The Technology of Emotion. Proceedings of the AIMI International Workshop*, pages 68–73, 1997.

[12] J. O. Smith. Efficient Simulation of the Reed-Bore and Bow-String Mechanisms. In *Proceedings of the 1986 International Computer Music Conference*, pages 275–280. Computer Music Association, 1986.

[13] J. O. Smith. *Physical Audio Signal Processing.* `http:`http://ccrma.stanford.edu/ jos/pasp/`//-ccrma.stanford.edu/~jos/pasp/`, accessed 2023-12-19. online book, 2010 edition.

[14] S. Snibbe, K. Maclean, R. Shaw, J. Roderick, W. Verplank, and M. Scheeff. Haptic Techniques for Media Control. *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, 08 2001.

[15] A. van Oosterhout, M. Bruns, and E. Hoggan. Facilitating Flexible Force Feedback Design with Feelix. In *Proceedings of the 2020 International Conference on Multimodal Interaction*, ICMI '20, page 184–193, New York, NY, USA, 2020. Association for Computing Machinery.

[16] S. Wang, E. Maestre, and G. Scavone. Acoustical Modeling of the Saxophone Mouthpiece as a Transfer Matrix. *The Journal of the Acoustical Society of America*, 149(3):1901–1912, Mar. 2021.