

Physical Modelling for Real-time Sound Synthesis

Bridging the gap between accuracy and real-time applications

Section 1. Project summary / abstract

In the past few decades, much research has been done on digital sound synthesis. Using a physical model (PM) of a musical instrument, rather than for example using sampling synthesis for creating sound, makes the playability of the digital instrument very flexible. In other words, the now-virtual instrument can easily adapt to changes in performance and playing style. Many PMs already exist and can output high quality sound which is sometimes indistinguishable from the real instrument.

“Why not just use the real instrument?” you might ask. Physical modelling makes it possible to extend traditional instruments in ways that would be impossible in the real world. Changing the shape, size or material of the instrument over time potentially results in interesting sounds and could even extend the possibilities for expression for the musician. Furthermore, old or rare instruments that can not be played anymore – as they are, for example, damaged or valuable museum pieces – can be resurrected virtually.

Going a step further, different components of different instruments can be linked together to create hybrid instruments. Virtually connecting, for example, the mouthpiece of a trumpet to a guitar string could potentially lead to interesting sounds.

There are various techniques to implement physical models. There exists a tradeoff between accuracy and speed, making high-quality physical modelling very computationally expensive. Only recently have we been able to run existing highly accurate PMs in real-time, making them playable for performers and musicians. The PhD project will focus on implementing existing PMs using techniques focusing on accuracy in real-time. As the implemented models will be playable in real-time, the use of different controllers, such as the Sensel Morph [1]. and the Haply [2] will be investigated.

To summarise, the goal of this project will be to push the state of the art in real-time accurate PMs and explore the possibilities that come with this, such as changing physically fixed parameters in real-time and combining components of different instruments to allow for novel interactions and create sounds never heard before. To showcase the project, an application will be made where the user can build their own real-time virtual instrument using the physical models created during the course of the PhD.

Section 2. The scientific content of the PhD project

a. Background

Physical modelling, along with sampling synthesis (also known as wavetable synthesis [3]) and spectral modelling, is a technique to synthesise sound. According to many, physical modelling is the best technique to realistically and naturally simulate real-world musical instruments [4–6]. As this technique simulates the instrument based on its physics rather than using pre-recorded samples, it is more flexible to player-interaction and thus more realistic when

synthesising sound in performance. Although physical models (PMs) could potentially sound indistinguishable from the instrument that they are simulating, it has been impossible, until recently, to make high-quality – defined here as non-linear and based on finite-difference schemes (FDSs) – PMs ‘playable’ in real-time [3]. With the computational power we currently possess, we can run the PMs in real-time and make them available for musicians in latency-less applications. These applications include digital instrument plugins that can be used by music producers, but also resurrect old or rare instruments that can not be played anymore due to damage, or because they are too valuable.

Even though there are already existing real-time PMs (see Section b), to my opinion, they do not make use of the full potential of using physical modelling for sound synthesis. PMs make it possible for parameters like shape, size, material properties, etc. to be changed, which is physically impossible or very hard to do. Furthermore, different instrument components can be combined to create hybrid instruments which can result in unique sounds that can only be created using PMs.

b. State-of-the-art

This section will give a brief introduction on the state of the art on physical modelling techniques, real-time implementation of these, modular environments for sound synthesis and instrument extension.

Physical Modelling

The physics of musical instruments can be well described by partial differential equations (PDEs) (see [6] among many others). Many of these equations and other knowledge currently available on the physics of musical instruments have been collected by N. H. Fletcher and T. D. Rossing in [7]. In the past decades, much research has been done on implementing these PDEs to model and simulate different musical instruments. Great overviews of implementation techniques are given by, for example, Vesa Välimäki et al. in [4] and Julius O. Smith in [3, 5]. Some techniques that are described in this literature can be found below:

Modal Synthesis decomposes a system into a series of uncoupled ‘modes of vibration’. First used in a musical context by Morrison and Adrien in [8], it is a technique that is still used today due to its computational efficiency, especially when simulating higher dimensional systems such as plates (2D) or rooms (3D) **correct right?**. It is especially effective when used to describe a linear system [9] with a small number of long-resonating modes [3]. When used to describe non-linear systems, however, the modes become ‘coupled’ and the system will quickly become more computationally expensive.

Finite-Difference Time Domain methods (FDTDs) (or Finite-Difference Schemes (FDSs)) aim to solve PDEs by approximating them with difference equations, discretising a continuous system into grid-points in space and time. In a musical context, this technique was first used for the case in string vibration in [10–12] and later in [13, 14]. Stefan Bilbao extensively describes this method in [6, 9]. Although very computationally expensive, this technique can most accurately resemble any system, whether it is linear or non-linear, time-invariant or time-variant.

Digital Waveguide Modelling (or Digital Waveguides (DWG)) is a modelling technique that discretises wave propagation and scattering. The technique was first presented in [15], and is

mostly used for one-dimensional systems, such as strings and acoustic tubes and decomposes their system into travelling wave components. This technique has also been used in higher-dimensional systems, but is superior in efficiency when used in the one-dimensional case [4]. Some authors have combined DWGs with FDSs (such as in [16, 17]) to accurately model non-linear behaviour while maintaining high-speed implementation.

Mass-spring networks can be similar in nature to FDTD methods, but treat each grid point as an individual mass connected to other masses through springs in a network. Pioneered in a musical context by Cadoz in [18–20] it is currently being further developed by mi-creative in a real-time, interactive environment [21, 22].

Real-time implementation

Even though physical modelling has been a popular research field in the past few decades, relatively little research has been done on making the models work in real-time, i.e., ‘playable’ [23]. Several virtual string instruments and different electric pianos have been made real-time by Pfeifle and Bader in [24–26]. They used field programmable gate arrays (FPGAs) for implementing models based on FDSs. Furthermore, Roland’s V-series use COSM (Composite Object Sound Modelling) technology [27] that implement real-time physical models in hardware instruments. In the NESS project [28], Stefan Bilbao and his team focused on implementing systems using FDTD methods in real-time, which is more relevant to this PhD project. More on this can be found below.

Modular Environments for Sound Synthesis

As will be described in the next section, the goal of the project is to create a modular environment consisting of different virtual instrument components. An existing modular environment for sound synthesis is the Synth Kit by littleBits [29] where the different modules are oscillators, filters, sequencers, etc., essentially making it a modular synthesiser where the components can physically be moved around and connected. Another example is MuX by Decochon [30], a modular synthesiser that the user can build and control in virtual reality. Both of these examples present an alternative way to control a modular synthesiser and are not based on physical models.

As part of the NESS project mentioned above, Webb and Bilbao presented a real-time physically modelled implementation of strings/bars and plates in a modular environment in [31]. Using C combined with AVX and multithreading they are able to make a system that can run a model of a plate with approximately 7000 grid points virtually connected to 10 strings run on a 4-core processor with a base clock rate of 3.1GHz. Their paper will be a good starting point for creating real-time models for musical instruments rather than ‘simple’ 1D and 2D models.

Extending Physical Models: from physically static to virtually dynamic

There is a lot of potential when using physical modelling to make physically fixed parameters virtually dynamic, i.e., variable in real-time. The main reasons to do this are 1) to create sonically interesting results and 2) to extend the possibilities and (potentially) enhancing expressivity for the musician. An example of existing dynamic physical models is Michon’s BladeAxe [32], which allows the user to alter the string length in real-time with the only control being the speed of an LFO. In [33], the authors present a flute of which the length can be

changed in real-time using a foot-pedal. In [34], the authors present a string-bridge-plate model based on modal synthesis where many parameters can be changed in real-time. In [35], I presented extensions to a plate reverb where the length and width of the plate and the microphone positions could be altered in real-time. In the PhD project, extensions like these will be further explored.

c. Project objectives

This section will state multiple research questions and give some background on each:

How can computationally expensive physical models be made playable in real-time?

A drawback of using physical modelling for sound synthesis is that the algorithms used are generally very computationally expensive causing most existing physical models to not run in real-time. In other words, these virtual instruments are not ‘playable’. One of the focuses of this PhD project will be to implement existing physical models and optimise them so that they work in real-time.

How can (the sound of) traditional instruments be extended upon?

As mentioned before, using PMs to simulate real-life instruments relieves the physical limitations that the real world imposes on them. These extensions can be divided into two categories:

1. making physically fixed parameters of the now-virtual instrument component dynamic, and
2. combining components of different instruments.

These “physical impossibilities” could potentially create sounds never heard before and will therefore be interesting to explore during the course of the PhD project. Both categories are individually addressed below:

PMs can go beyond what traditional musical instruments can do. ‘Virtualising’ traditional musical instruments comes with an infinite amount of (physically impossible) possibilities – solid structures can be made flexible, material properties can be changed, etc. It would thus be interesting to investigate what parameters of the model could be made dynamic to create sonically interesting outcomes and/or allow musicians for a higher level of expressiveness.

Instruments can generally be divided in an exciter and a resonator component [36]. Examples of exciters are the bow of a violin or the player’s lips for a trumpet. Resonators would in this case be the string and body of the violin, or the trumpet itself (the hollow tube). All of this can be modelled and combined to create hybrid instruments. Using the previous examples, we can now make a bowed trumpet, or a lip-excited string.

How can the now-virtual instruments be controlled in a natural way?

A challenge in playing virtual instruments is their control. As is the case with many physical instruments, one interacts immediately with the sound-creating object, such as a string on a guitar or a membrane on a drum. This allows the musician to be a lot more expressive than if he/she only used the keyboard and mouse. During the course of this PhD project, experiments

have already been done with the Sensel Morph, which is a controller containing ca. 20,000 pressure sensors that allow for highly expressive control of the instruments [1].

Expressivity, however, is not the only thing that makes an instrument interesting and enjoyable to play. The interaction with a virtual instrument could feel very ‘dry’ or unnatural as there is no haptic feedback; something which is the case in (nearly) all physical instruments. In virtual instruments, this ‘feel’ could be simulated by a controller allowing for haptic feedback such as the Haply [2] or attaching ‘haptuators’ [37] to other controllers.

d. Key methods

This section describes the methods I will use to answer the above research questions.

How can computationally expensive physical models be made playable in real-time?

The implementation process will be the same for all PMs. The steps for making the real-time PMs are listed below:

MATLAB

First the PM will be implemented as a prototype in the MATLAB environment. FDTD methods will be used to implement the instrument(component) with the highest accuracy possible. An energy analysis (as found in [6]) will be carried out to verify energy conservation and confirm correctness of the model.

C++ / JUCE

After correctness of the implementation has been determined, it can now be implemented in real-time. For this, C++ together with the JUCE framework will be used as this is a very suitable language for real-time implementation [38, 39].

Optimisation and simplification

The possibilities of using GPU, CPU Parallelisation (Multi-threading) and Advanced Vector Intrinsics (AVX) will be explored to speed up existing algorithms to make them perform in real-time. If after optimisation, the algorithm is still not fast enough, the model will be simplified. A main focus while doing this is to preserve (perceptual) sound quality as much as possible. Parts of the model could also be replaced by computationally more efficient implementation techniques such as using DWGs. The way the PM will be optimised and simplified depends on the model at hand.

How can (the sound of) traditional instruments be extended upon?

When simulating a traditional musical instrument using PMs, it relieves all physical restrictions on the instrument. In terms of a PM, we can say that (almost) all parameters are free or can be made dynamic.

The collection of parameters the PMs rely on will be investigated by listening to the sound output after a change in their values. I hypothesise that the most interesting parameters to change are the ones that directly influence musical qualities such as pitch or timbre. For example, the length of a string is directly correlated to pitch. The radius of a string is too, but a change in this will change the inharmonicity of the string and thus also the timbre.

A problem when using FDSs, is that the number of grid points depends on the parameters mentioned above. This will thus also dynamically change, possibly causing stability issues. Using energy and stability analysis techniques, I will investigate whether it will be possible to change these parameters without causing the system to become unstable.

e. Significance and outcome

The main outcome of the PhD project will be an application that combines all of the aforementioned research questions: a real-time modular instrument that the user can build themselves, where the physical parameters of the components can be dynamically changed. As said before, the physics of musical instruments can be well described by PDEs. The application will contain some sort of solver where the user can insert the PDE (or FDS) and have the application analyse and solve it. The analysis stage is important to make sure that the custom PDE will be stable.

Furthermore, the application will be playable by means of the different controllers mentioned above. The mapping of these will also be made available to the user so that also the interaction is part of the modularity of the application.

Section 3. Work and publication plans

a. Work and Time Plans

| Year | 2018 | | 2019 | | | | 2020 | | | | 2021 | |
|---------------------------------------|------|-----|------|----|-----|----|------|-----|---|---|------|---|
| Quarter | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 |
| Literature study | | | | | | | | | | | | |
| Implementing existing physical models | | | | | | | | | | | | |
| Real-time implementation | | | | | | | | | | | | |
| Build modular PM framework | | | | | | | | | | | | |
| Writing thesis | | | | | | | | | | | | |
| PhD Courses | | | | | | | | | | | | |
| Writing conference / journal paper | | | | | | | | | | | | |
| Publishing papers | | | C1 | C2 | | J1 | C4 | | | | J2 | |
| Milestones | | MS1 | MS2 | | MS3 | | | MS4 | | | | |

| | | | |
|---------------------|----------------------------|--------------------|-------------|
| Activities finished | Activities being performed | Planned activities | Buffer time |
|---------------------|----------------------------|--------------------|-------------|

Milestones:

MS1 (Nov): Implement FDS Non-linear Bowed-String model (proving understanding of FDSs).

MS2 (Jan): Make bow-model work in real-time using C++/JUICE.

MS3 (Sep): Make first full virtual instrument in real-time.

MS4 (Apr): Started modular application: different instrument components (modules) can interactively be connected together.

MS5 (March): Finished modular application: many modules, FDS solver, energy / stability analysis included.

b. Outline of thesis

The thesis will be written as a collection of the papers published throughout the course of the project. This will be accompanied by an extended summary that will introduce the real-time modular application and each module individually. Lastly, several tutorials on, for example, energy and stability analysis will be given and included to the thesis as appendices.

c. Tentative publication list

C1 [40]: S. Willemsen, N. Andersson, S. Serafin and S. Bilbao, “Real-time control of large scale physical models using the Sensel Morph,” *Proceedings of the 16th Sound and Music Computing Conference (SMC)*, pp. 275-280, 2019.

C2 [41]: S. Willemsen, S. Bilbao and S. Serafin, “Real-time implementation of an elasto-plastic friction model applied to stiff strings using finite difference schemes,” *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx)*, 2019.

C3. S. Willemsen, S. Bilbao and S. Serafin, “Necessary and sufficient conditions for the elasto-plastic friction model,” *SOMEWHERE*, 2019.

C4. S. Willemsen, S. Bilbao and S. Serafin, “Introducing two hybrid virtual instruments: the bowed trumpet, and the lip-excited string,” *Proceedings of the 17th Sound and Music Computing Conference (SMC)*, 2020.

J1: S. Willemsen, M. Ducceschi, S. Bilbao and S. Serafin, “Modelling the Tromba Marina: non-linearities using non-iterative methods,” *Journal for New Music Research*, 2020.

J2: S. Willemsen, S. Bilbao and S. Serafin, “The modular physical modelling instrument using finite difference schemes in real-time,” *Computer Music Journal*, 2021.

Section 4. Supervisor/student co-operation agreements

Time management

Day-to-day management of the project is the student’s responsibility. The student is required to inform the supervisor about changes in milestones or the project plan in general.

Meetings and Communication

All communication between the student and the supervisor will happen through emails and face- to-face meetings which should happen least once every two weeks. If either the student or the supervisor is unavailable for more than 48 hours, the other party should be informed.

Collaboration

The supervisor is required to give feedback on content written by the student. This content includes but is not limited to, conference/journal papers and the PhD thesis.

Section 5. Plan for PhD Courses (both general and project related courses)

| Courses | Place/organised by | ECTS | General / Project Course | Status |
|---|------------------------------|------|--------------------------|----------|
| Introduction to the PhD Study | AAU / Lars Haastруп Pedersen | 0.5 | General | Finished |
| Writing and Reviewing Scientific Papers | AAU / Jakob Stoustrup | 3.75 | General | Finished |
| Design and Analysis of Experiments | AAU / Esben Høg | 4.5 | General | Finished |
| Virtual, Augmented, Mixed Realities | AAU / Stefania Serafin | 5 | Project | Enrolled |
| | | | | |
| | | | | |

Section 6. Plan for fulfilment of knowledge dissemination

The work done during the course of the project will be presented through conference and journal papers. The conferences and journals that I plan to publish at, include, but are not limited to:

- The Sound and Music Computing (SMC) Conference
- The International Conference on Digital Audio Effects (DAFx)
- The International Conference on New Interfaces for Musical Expression (NIME)
- International Symposium on Computer Music Multidisciplinary Research (CMMR)
- Audio Engineering Society (AES) Conferences / Journal
- Computer Music Journal
- Journal for New Music Research

Furthermore, I will teach (assist) the Physical Models for Sound Synthesis course (SMC8), the Sound Processing course (SMC7) and possibly the Real-time Interaction and Performance and Sound and Music Signal Analysis courses. Lastly, I will supervise group projects in the Sound and Music Computing studies.

Section 7. Agreements on immaterial rights to patents

Immaterial rights to patents will follow the standard university rules.

Section 8. External co-operation

From the start of the PhD project, I have collaborated to a great extent with Stefan Bilbao and his team at the University of Edinburgh. I travelled to Edinburgh for two weeks to learn different analysis techniques related to FDSs and plan to do this again during the course of the PhD. Moreover, both past and future papers have/will be in collaboration with him and his team.

Next to this, I will collaborate with Karolina Prawda, a PhD student part of Vesa V'alim'aki's team at Aalto University Finland, through the NordicSMC project.

Section 9. Financing budget

The PhD project is funded through a PhD Stipend at the Department of Architecture, Design Media Technology (no. 7-18017).

Section 10. References

- [1] Sensel Inc., “Sensel Morph,” Available at <https://sensel.com/>, Accessed April 01, 2019.
- [2] Haply Robotics, “Haply Robotics,” Available at <https://haply.co/robotics.html>, Accessed April 01, 2019.
- [3] J. O. Smith, “Virtual acoustic musical instruments: Review and update,” *Center for computer research in music and acoustics (CCRMA)*, 2010.
- [4] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, “Discrete-time modelling of musical instruments,” *Institute of Physics Publishing*, 2006.
- [5] J. O. Smith, *Physical audio signal processing: For virtual musical instruments and digital audio effects*, W3K Publishing, 2010.
- [6] S. Bilbao, *Numerical sound synthesis: Finite difference schemes and simulation in musical acoustics*, John Wiley & Sons, 2009.
- [7] N.H. Fletcher and T.D. Rossing, *The physics of musical instruments*, Springer, second edition, 1998.
- [8] J.D. Morrison and J.-M. Adrien, “Mosaic: A framework for modal synthesis,” *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, 1993.
- [9] S. Bilbao, B. Hamilton, R. Harrison, and A. Torin, “Chapter 19: Finite-difference schemes in musical acoustics: A tutorial,” in *Springer Handbook of Systematic Musicology*, pp. 349–384. Springer, 2018.
- [10] P. Ruiz, “A technique for simulating the vibrations of strings with a digital computer,” M.S. thesis, University of Illinois, 1969.
- [11] L. Hiller and P. Ruiz, “Synthesizing musical sounds by solving the wave equation for vibrating objects: Part 1,” *Journal of the Audio Engineering Society*, vol. 19, no. 6, pp. 462–470, 1971.
- [12] L. Hiller and P. Ruiz, “Synthesizing musical sounds by solving the wave equation for vibrating objects: Part II,” *Journal of the Audio Engineering Society*, vol. 19, no. 7, pp. 542–550, 1971.
- [13] A. Chaigne, “On the use of finite differences for musical synthesis. Application to plucked stringed instruments,” *Journal d’Acoustique*, vol. 5, no. 2, pp. 181–211, 1992.
- [14] A. Chaigne and A. Askenfelt, “Numerical simulations of struck strings. I. A physical model for a struck string using finite difference methods,” *Journal of Acoustical Society of America*, vol. 95, no. 2, pp. 1112–1118, 1994.
- [15] J.O. Smith, “Physical modeling using digital waveguides,” *Computer music journal*, vol. 16, no. 4, pp. 74–91, 1992.
- [16] C. Erkut and M. Karjalainen, “Finite difference method vs. digital waveguide method in string instrument modeling and synthesis,” *International Symposium on Musical Acoustics*, 2002.

- [17] E. Maestre, C. Spa, and J.O. Smith, “A bowed string physical model including finite-width thermal friction and hair dynamics,” *Proceedings ICMC|SMC|2014*, pp. 1305–1311, 2014.
- [18] C. Cadoz, “Synthèse sonore par simulation de mécanismes vibratoires,” 1979, Thèse de Docteur Ingénieur, I.N.P.G. Grenoble, France.
- [19] Cadoz C., Luciani A., and J.-L. Florens, “Responsive input devices and sound synthesis by simulation of instrumental mechanisms,” *Computer Music Journal*, vol. 8, no. 3, pp. 60–73, 1983.
- [20] C. Cadoz, A. Luciani, and J.L. Florens, “CORDIS-ANIMA: a modeling and simulation system for sound and image synthesis: the general formalism,” *Computer music journal*, vol. 17, no. 1, pp. 19–29, 1993.
- [21] J. Villeneuve and J. Leonard, “Mass-interaction physical models for sound and multi-sensory creation: starting anew,” *Proceedings of the 16th Sound and Music Computing Conference*, 2019.
- [22] J. Leonard and J. Villeneuve, “MI-GEN: An efficient and accessible mass interaction sound synthesis toolbox,” *Proceedings of the 16th Sound and Music Computing Conference*, 2019.
- [23] S. Mehes, M. van Walstijn, and P. Stapleton, “Towards a virtual-acoustic string instrument,” *Proceedings of the 13th Sound and Music Computing Conference (SMC)*, 2016.
- [24] F. Pfeifle and R. Bader, “Real-time finite difference physical models of musical instruments on a field programmable gate array (FPGA),” *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx)*, 2012.
- [25] F. Pfeifle and R. Bader, “Real-time finite-difference method physical modeling of musical instruments using field-programmable gate array hardware,” *Journal of the Audio Engineering Society*, vol. 63, no. 12, pp. 1001–1016, 2015.
- [26] F. Pfeifle, “Real-time physical model of a wurlitzer and a rhodes electric piano,” *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx)*, 2017.
- [27] J. Bybee, “COSM REVISITED,” Available at https://www.boss.info/us/community/boss_users_group/1319/, Accessed July 01, 2019.
- [28] The University of Edinburgh, “NESS – Next Generation Sound Synthesis,” Available at <http://www.ness.music.ed.ac.uk/>, Accessed July 01, 2019.
- [29] littleBits Electronics Inc., “SynthKit – littleBits,” Available at <https://shop.littlebits.com/products/synth-kit>, Accessed July 01, 2019.
- [30] Decochon, “MuX–Build Sound,” Available at <https://www.playmux.com/>, Accessed July 01, 2019.
- [31] C.J. Webb and S. Bilbao, “On the limits of real-time physical modelling synthesis with a modular environment,” *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx)*, 2015.
- [32] R. Michon, J.O. Smith, M. Wright, and C. Chafe, “Augmenting the iPad: the BladeAxe,” *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2016.
- [33] S. Gelineck, N. Böttcher, L. Martinussen, and S. Serafin, “Virtual reality instruments capable of changing dimensions in real-time,” *2nd International Conference on Enactive Interfaces (ENACTIVE05)*, 2005.
- [34] M. van Walstijn and S. Mehes, “An explorative string-bridge-plate model with tunable parameters,” *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx)*, 2017.
- [35] S. Willemsen, S. Serafin, and J.R. Jensen, “Virtual analog simulation and extensions of plate reverberation,” *Proceedings of the 14th Sound and Music Computing Conference (SMC)*, pp. 314–319, 2017.
- [36] G. Borin, G. De Poli, and A. Sarti, “A modular approach to excitator-resonator interaction in physical models syntheses,” *Proceedings of the International Computer Music Conference*, 1989.
- [37] Tactile Labs, “Haptuator Original,” Available at <http://tactilelabs.com/products/haptics/haptuator/>, Accessed July 08, 2019.
- [38] S. Meyers, *Effective C++*, Addison-Wesley, third edition, 2005.
- [39] JUCE ROLI, “JUCE,” Available at <https://juce.com/>, accessed April 04, 2019.
- [40] S. Willemsen, N. Andersson, S. Serafin, and S. Bilbao, “Real-time control of large-scale modular physical models using the sensel morph,” *Proceedings of the 16th Sound and Music Computing Conference (SMC)*, pp. 151–158, 2019.

- [41] S. Willemsen, S. Bilbao, and S. Serafin, “Real-time implementation of an elasto-plastic friction model applied to stiff strings using finite difference schemes,” *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx)*, 2019.