The Virtual Musical Instrument

Real-Time Simulation of Musical Instruments using Finite-Difference Time-Domain Methods

Ph.D. Dissertation

Silvin Willemsen

Dissertation submitted April 29, 2021

Thesis submitted: April 29, 2021

PhD Supervisor: Prof. Stefania Serafin
Aalborg University

PhD Committee: Prof. X, Y University
Prof. X, Y University
Prof. X, Y University

PhD Series: Technical Faculty of IT and Design, Aalborg University

Department: Department of Architecture, Design and Media Technology

ISSN: xxxx-xxxx
ISBN: xxx-xx-xxxx-xxx-x

Published by:
Aalborg University Press
Skjernvej 4A, 2nd floor
DK – 9220 Aalborg Ø
Phone: +45 99407140
aauf@forlag.aau.dk
forlag.aau.dk

Printed in Denmark by Rosendahls, 2021

Curriculum Vitae

Silvin Willemsen

Here is the CV text.

Acknowledgements

I would like to thank my mom..

List of Publications

Listed below are the publications that I (co)authored during the Ph.D. project grouped by: the main papers, which are also included in Part IX, papers where I had a supervisory role, and finally, miscellaneous publications.

Main Publications

[A] S. Willemsen, N. Andersson, S. Serafin, and S. Bilbao, "Real-time control of large-scale modular physical models using the sensel morph," in *Proceedings of the 16th Sound and Music Computing (SMC) Conference*, 2019, pp. 275–280.

[B] S. Willemsen, S. Bilbao, N. Andersson, and S. Serafin, "Physical models and real-time control with the sensel morph," in *Proceedings of the 16th Sound and Music Computing (SMC) Conference*, 2019, pp. 95–96.

[C] S. Willemsen, S. Bilbao, and S. Serafin, "Real-time implementation of an elasto-plastic friction model applied to stiff strings using finite difference schemes," in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx-19)*, 2019, pp. 40–46.

[D] S. Willemsen, S. Serafin, S. Bilbao, and M. Ducceschi, "Real-time implementation of a physical model of the tromba marina," in *Proceedings of the 17th Sound and Music Computing (SMC) Conference*, 2020, pp. 161–168.

[E] S. Willemsen, R. Paisa, and S. Serafin, "Resurrecting the tromba marina: A bowed virtual reality instrument using haptic feedback and accurate physical modelling," in *Proceedings of the 17th Sound and Music Computing (SMC) Conference*, 2020, pp. 300–307.

[F] S. Willemsen, A.-S. Horvath, and M. Nascimben, "Digidrum: A haptic-based virtual reality musical instrument and a case study," in *Proceedings of the 17th Sound and Music Computing (SMC) Conference*, 2020, pp. 292–299.

[G] S. Willemsen, S. Bilbao, M. Ducceschi, and S. Serafin, "Dynamic grids for finite-difference schemes in musical instrument simulations," in

Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020in21), 2021.

[H] ——, "A physical model of the trombone using dynamic grids for finite-difference schemes," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020in21)*, 2021.

Publications with a Supervisory Role

[S1] R. S. Alecu, S. Serafin, S. Willemsen, E. Parravicini, and S. Lucato, "Embouchure interaction model for brass instruments," in *Proceedings of the 17th Sound and Music Computing (SMC) Conference*, 2020, pp. 153–160.

[S2] T. Lasickas, S. Willemsen, and S. Serafin, "Real-time implementation of the shamisen using finite difference schemes," in *Proceedings of the 18th Sound and Music Computing (SMC) Conference*, 2021.

[S3] C. Kaloumenou, S. Lambda, P. Pouliou, M. G. Onofrei, S. Willemsen, C. Jaller, and S. Serafin, "Recreating the amoeba violin using physical modelling and augmented reality," in *Proceedings of the 18th Sound and Music Computing (SMC) Conference*, 2021.

[S4] M. G. Onofrei, S. Willemsen, and S. Serafin, "Implementing complex physical models in real-time using partitioned convolution: An adjustable spring reverb," in *Proceedings of the 18th Sound and Music Computing (SMC) Conference*, 2021.

[S5] ——, "Real-time implementation of an elasto-plastic friction drum using finite difference schemes," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020in21)*, 2021.

Miscellaneous Publications

[M1] J. M. Hjerrild, S. Willemsen, and M. G. Christensen, "Physical models for fast estimation of guitar string, fret and plucking position," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 155–159, 2019.

[M2] K. Prawda, S. Willemsen, S. Serafin, and V. Välimäki, "Flexible real-time reverberation synthesis with accurate parameter control," in *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx-20)*, 2020.

[M3] M. Ducceschi, S. Bilbao, S. Willemsen, and S. Serafin, "Linearly-implicit schemes for collisions in musical acoustics based on energy quadratisation," *Journal of the Acoustical Society of America (JASA)*, 2021.

Abstract

English abstract

Abstract

Resumé

Danish Abstract

Contents

exactly as in [3]

Todo list

Preface

Starting this Ph.D. project, I did not have a background in mathematics, physics or computer science, which were three equally crucial components in creating the result of this project. After the initial steep learning curve of notation and terminology, I was surprised to find that the methods used for physical modelling are actually quite straightforward!

Of course it should take a bit of time to learn these things, but

Many concepts that seemed impossible at the beginning

I feel that the literature lacks a lot of the intuition needed for readers without a background in any of these topics. Rather, much of the literature I came across assumes that the reader has a degree in at least one of the aforementioned topics. This is why I decided to write this work a bit more pedagogical than what could be expected.

I believe that anyone with some basic skills in mathematics and programming is able to create a simulation based on physics within a short amount of time, given the right tools, which I hope that this dissertation could be.

The knowledge dissemination of this dissertation is thus not only limited to the research done and publications made over the course of the project, but also its pedagogical nature hopefully allowing future (or past) students to benefit from.

As with a musical instrument itself, a low entry level, a gentle learning curve along with a high virtuosity level is desired. Take a piano, for instance. Most will be able to learn a simple melody — such as "Frère Jacques" — in minutes, but to become virtuous requires years of practice.

This is the way I wanted to write this dissertation: easy to understand the basic concepts, but many different aspects touched upon to allow for virtuosity. Hopefully by the end, the reader will at least grasp some highly complex concepts in the fields of mathematics, physics and computer science (which will hopefully take less time than it takes to become virtuous at playing the piano).

Some basic calculus knowledge is assumed.

I wanted to show my learning process and (hopefully) explain topics such as *Energy Analysis*, *Stability Analysis*, etc. in a way that others lacking the same

knowledge will be able to understand.

Make physical modelling more accessible to the non-physicist.

Interested in physically impossible manipulations of now-virtual instruments.

Silvin Willemsen
Aalborg University, April 29, 2021

Part I

Introduction

Chapter 1

Physical Modelling of Musical Instruments

The history of physical modelling of musical instruments

Exciter-resonator approach.

The time-evolution of dynamic systems can be conveniently described by differential equations. Examples of a dynamic systems are a guitar string, a drum-membrane, or a concert hall; three very different concepts, but all based on the same types of equations of motion.

Though these equations are very powerful, only few have a closed-form solution. What this means is that in order for them to be implemented, they need to be approximated. There exist different approximation techniques to do this

1.1 Physical Modelling Techniques

- Modal Synthesis
- Finite-difference Time-domain methods
- Finite Element Methods
- Digital waveguides
- Mass-spring systems
- Functional transformation method
- State-space
- Wave-domain

- Energy-based

Advantages of finite-difference methods

Using FDTD methods can be quite computationally heavy. Moore's law [11]

1.2 Real-Time Implementation

Although many techniques to digitally simulate musical instruments exist proving that we have only recently reached the computing power in personal computers to make real-time playability of these models an option. The biggest challenge in real-time audio applications as opposed to those only involving graphics, is that the sample rate is extremely high. As Nyquist's sampling theory tells us, a sampling rate of at least 40 kHz is necessary to produce frequencies up to the human hearing limit of 20 kHz **[Nyquist]**. Visuals

Real-time: no noticable latency

exactly as in [3]

1.3 Why?

1.3.1 Audio plugins

Samples, or recordings, of real instruments are static and unable to adapt to changes in perfomance. Moreover, capturing the the entire interaction space of an instrument is nearly impossible. Imagine recording a violin with every single combination of bowing force, velocity, position, duration and other aspects such as vibrato, pizzicato. Even if a complete sample library could be created, this would contain an immense amount of data.

Samples vs. Physical Modelling:

Trade off between storage and speed

Using musical instrument simulations, on the other hand, allows the sound to be generated on the spot based on physical parameters that the user can interact with.

1.3.2 Resurrect old or rare instruments

Even popular instruments require maintenance and might need to be replaced after years of usage.

1.3.3 Go beyond what is physically possible

1.4 Thesis Objectives and Main Contributions

Over the past few decades, much work has been done on the accurate modelling of physical phenomena. In the field of sound and musical instruments..

From [6] to [5]

The main objective of this thesis is to implement existing physical models in real time using FDTD methods. Many of the physical models and methods presented in this thesis are taken from the literature and are thus not novel.

Secondly, to combine the existing physical models to get complete instruments and be able to control them in real time.

As FDTD methods are quite rigid, changing parameters on the fly, i.e., while the instrument simulation is running, is a challenge. Other techniques, such as modal synthesis, are much more suitable for this, but come with the drawbacks mentioned in Section 1.1. Therefore, a novel method was devised to smoothly change parameters over time, introducing this to FDTD methods.

1.5 Thesis Outline

- Physical models
  - Resonators
  - Exciters
  - Interactions
- Dynamic Grids
- Real-Time Implementation and Control
- Complete instruments
  - Large-scale physical models
  - Tromba Marina
  - Trombone

Notes

- Think about how to define real-time.
- Create an intuition for different parts of the equation
- Talk about input and output locations and how that affects frequency content (modes).

One over number $\rightarrow$ reciprocal of number

Example: When the waveform consists entirely of harmonically related frequencies, it will be periodic, with a period equal to the reciprocal of the fundamental frequency (from An Introduction to the Mathematics of Digital Signal Processing Pt 2 by F. R. Moore)

Chapter 2

FDTD Methods and Analysis Techniques

"Since Newton, mankind has come to realize that the laws of physics are always expressed in the language of differential equations."
- Steven Strogatz
(https://youtu.be/O85OWBJ2ayo?t=44)

This chapter introduces some important concepts needed to understand the physical models presented later on in this document. By means of a simple mass-spring system and the 1D wave equation, the notation (and terminology) used throughout this document will be explained, together with some important analysis techniques. Before we dive into the mathematics, let us go over some useful terminology.

Differential equations

As mentioned in Chapter 1 differential equations are used to describe the motion of dynamic systems. These systems can describe anything from the transverse displacement of a string to the air pressure in an acoustic tube.

A characteristic feature of these equations is that, rather than an absolute value or *state* of a system, the time derivative of its state – its velocity – or the second-order time derivative – its acceleration – is described. From this, the absolute state of the system can be computed.

This state is usually described by the variable u which is (nearly) always a function of time, i.e., $u = u(t)$. If the system is distributed in space, u also becomes a function of space, i.e., $u = u(x, t)$, or with two spatial dimensions, $u = u(x, y, t)$, etc. Though this work only describes systems of up to two spatial dimensions, one could potentially extend to systems of infinite spatial dimensions evolving over time!

If u is only a function of time, the differential equation that describes the motion of this system is called an *ordinary differential equation* (ODE). If u is also a function of at least one spatial dimension, the equation of motion is a called a *partial differential equation* (PDE).

The literature uses different types of notation for taking (continuous-time) partial derivatives. Applied to a state variable u these can look like

$$\frac{\partial^2 u}{\partial t^2} \quad \text{(classical notation)}$$
$$u_{tt} \quad \text{(subscript notation)}$$
$$\partial_t^2 u \quad \text{(operator notation)}$$

all of which mean a second-order derivative with respect to time t, i.e., u's acceleration. In this remainder of this document, the operator notation will be used. Often-used partial derivatives and their interpretation are shown below

maybe not yet as this is super general still

$$\partial_t^2 u \quad \text{(acceleration)} \qquad \partial_x^2 u \quad \text{(curvature)}$$
$$\partial_t u \quad \text{(velocity)} \qquad \partial_x u \quad \text{(slope)}$$

Note: difference between 1D, 2D spatial, and 1D, 2D displacement (polarisation). Transverse displacement of 1D wave here

Now that an equation has been established, how

Discretisation using FDTD methods

Finite-difference time-domain (FDTD) methods essentially subdivide a continuous equation in discrete points in time and space, a process called *discretisation*. Once an ODE or PDE is discretised using these methods it is now called a *Finite-Difference (FD) Scheme*.

grid figure

We start by defining a discrete *grid* over time and space which we will use to approximate our continuous equations. A system $u = u(x, t)$ defined over time t and one spatial dimension x, can be approximated using a *grid function* u_l^n. Here, subscript l superscript n describe the spatial and temporal indices respectively and arise from the discretisation of the continuous variables x and t according to $x = lh$ and $t = nk$. The spatial step h, also called the *grid spacing* describes the distance (in m) between two neighbouring grid points and the temporal step k, or *time step* is the time (in s) between two consecutive temporal indices. The latter can be calculated $k = 1/f_\mathrm{s}$ for a sample rate f_s (in Hz). In many audio applications $f_\mathrm{s} = 44100$ Hz which will be used in this work (unless denoted otherwise).

To summarise

$$u(x, t) \approx u_l^n \quad \text{with} \quad x = lh \quad \text{and} \quad t = nk \tag{2.1}$$

Now that the state variable has a discrete counterpart, this leaves the derivatives to be discretised. We start by introducing shift operators that can be applied to a grid function and changes its indexing (either spatial or temporal). Forward and backward shifts in time, together with the identity operation (not altering u_l^n) are

$$e_{t+}u_l^n = u_l^{n+1}, \quad e_{t-}u_l^n = u_l^{n-1}, \quad \text{and} \quad 1u_l^n = u_l^n. \tag{2.2}$$

Similarly, forward and backward shifts in space are

$$e_{x+}u_l^n = u_{l+1}^n, \quad \text{and} \quad e_{x-}u_l^n = u_{l-1}^n. \tag{2.3}$$

These shift operators are rarely used in isolation, though they do appear in energy analysis techniques detailed in Section 2.3. The operators do, however, form the basis of commonly used FD operators. The first-order derivative in time can be discretised three different ways. The forward, backward and centred difference are

check centred instead of centered, full document sweep

$$\partial_t \approx \begin{cases} \delta_{t+} \triangleq \frac{1}{k}\left(e_{t+} - 1\right), & \text{(2.4a)} \\ \delta_{t-} \triangleq \frac{1}{k}\left(1 - e_{t-}\right), & \text{(2.4b)} \\ \delta_{t\cdot} \triangleq \frac{1}{2k}\left(e_{t+} - e_{t-}\right), & \text{(2.4c)} \end{cases}$$

where "$\triangleq$" means "equal to by definition". These operators can then be applied to grid function u_l^n to get

$$\partial_t u \approx \begin{cases} \delta_{t+}u_l^n = \frac{1}{k}\left(u_l^{n+1} - u_l^n\right), & \text{(2.5a)} \\ \delta_{t-}u_l^n = \frac{1}{k}\left(u_l^n - u_l^{n-1}\right), & \text{(2.5b)} \\ \delta_{t\cdot}u_l^n = \frac{1}{2k}\left(u_l^{n+1} - u_l^{n-1}\right). & \text{(2.5c)} \end{cases}$$

Note that the centred difference has a division by $2k$ as the time difference between $n+1$ and $n-1$ is twice the time step.

The same can be done in for a first-order derivative in space, where the forward, backward and centred difference are

figure here for interpretation of accuracy of operators

$$\partial_x \approx \begin{cases} \delta_{x+} \triangleq \frac{1}{h}\left(e_{x+} - 1\right), & \text{(2.6a)} \\ \delta_{x-} \triangleq \frac{1}{h}\left(1 - e_{x-}\right), & \text{(2.6b)} \\ \delta_{x\cdot} \triangleq \frac{1}{2h}\left(e_{x+} - e_{x-}\right), & \text{(2.6c)} \end{cases}$$

and when applied to u_l^n are

$$\partial_x u \approx \begin{cases} \delta_{x+}u_l^n = \frac{1}{h}\left(u_{l+1}^n - u_l^n\right), & \text{(2.7a)} \\ \delta_{x-}u_l^n = \frac{1}{h}\left(u_l^n - u_{l-1}^n\right), & \text{(2.7b)} \\ \delta_{x\cdot}u_l^n = \frac{1}{2h}\left(u_{l+1}^n - u_{l-1}^n\right). & \text{(2.7c)} \end{cases}$$

Higher order differences can be obtained from first-order difference operators. The second-order difference in time may be approximated using

$$\partial_t^2 \approx \delta_{t+}\delta_{t-} = \delta_{tt} = \frac{1}{k^2}\left(e_{t+} - 2 + e_{t-}\right), \tag{2.8}$$

where "2" is the identity operator applied twice. and similarly for the second-order difference in space

$$\partial_x^2 \approx \delta_{x+}\delta_{x-} = \delta_{xx} = \frac{1}{h^2}\left(e_{x+} - 2 + e_{x-}\right). \tag{2.9}$$

More explanation about combining operators can be found in Section 3.2.1.

Averaging operators

Operators and derivatives in 2D will be discussed in Chapter 4.

Accuracy

Taylor series expansion...

2.0.1 Identities

Extremely useful are

An identity is something that holds true

$$\delta_{tt} u_l^n = \frac{2}{k}\left(\delta_{t\cdot} u_l^n - \delta_{t-} u_l^n\right) \tag{2.10a}$$

$$\delta_{tt} \tag{2.10b}$$

That these identities hold can easily be proven by expanding the operators

Implementation

In the following

- Continuous-time
- Discrete-time
- Implementation (update equation)

Something something newton's second law

initial conditions

It is important to note that a discrete FD scheme is an *approximation* to a continuous PDE, not a sampled version of it. This means that the resulting schemes are rarely an exact solution to the original continuous equation.

2.1 The Mass-Spring System

Though a complete physical modelling field on its own (see Chapter 1), mass-spring systems are also sound-generating systems themselves.

Continuous-time

The ODE of a simple mass-spring system is defined as

$$\frac{d^2u}{dt^2} = -\omega_0^2 u \tag{2.11}$$

Discrete-time

The , u is approximated using

$$u(t) \approx u^n \tag{2.12}$$

where $t = nk$

Implementation

2.2 The 1D Wave Equation

Arguably the most important PDE in the field of physical modelling for sound synthesis is the 1D wave equation.

2.2.1 Continuous-time

The state of the system $u = u(x, t)$ meaning that is on top of being defined in time t it is distributed over space x. The 1D wave equation is defined as follows

$$\partial_t^2 u = c^2 \partial_x^2 u, \tag{2.13}$$

where c is the wave speed (in m/s). The state variable u can be used to describe transverse vibration in an ideal string, longitudinal vibration in an ideal bar and pressure in an acoustic tube. Although the behaviour of this equation alone does not appear in the real world, as no physical system is ideal, it is extremely useful as a test case and a basis for more complicated models. Here, it will be used to introduce various concepts and analysis techniques in the field of FDTD methods.

Intuition

Although the 1D wave equation often appears in the literature, an intuition or interpretation of why it works the way it does is hard to find. In the following, u describes the transverse displacement of an ideal string.

OR

I would like to use this opportunity to provide some extra explanation as to how and why the 1D wave equation in (2.13) works the way it does. This will hopefully provide some basic intuition into the workings of PDEs that will make it easier to work with later on. somethingsomething

Going back to Newton's second law

The acceleration of $u = u(x,t)$ at location x is determined by the second-order spatial derivative of u at that same location (scaled by a constant c^2). In the case that u describes the transverse displacement of an ideal string, this second-order derivative denotes the *curvature* of this string. As c^2 is always positive, the sign (or direction) of the acceleration is fully determined by the sign of the curvature. In other words, a 'positive' curvature at location x along the ideal string yields a 'positive' or upwards acceleration at that same location.

What a 'positive' or 'negative' curvature implies is more easily seen when we take a simple function describing a parabola, $y(x) = x^2$, and take its second derivative to get $y''(x) = 2$. The answer is a positive number which means that y has a positive curvature.

So, what does this mean for the 1D wave equation? As a positive curvature implies a positive or upwards acceleration as per Eq. (2.13), u with a positive curvature at a location x will start to move upwards and vice versa. Of course, the state of a physical system such as u will rarely have a perfect parabolic shape, but the argument still applies.

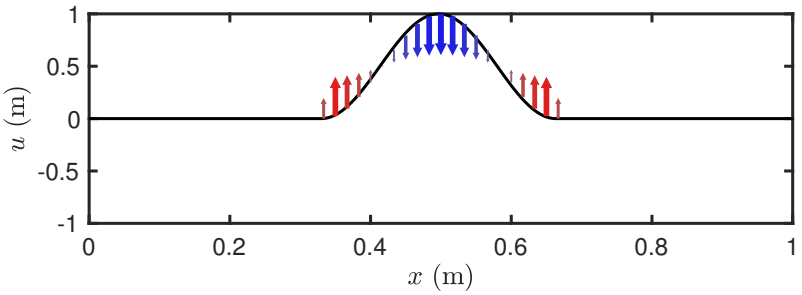

Fig. 2.1: The forces acting on the 1D wave equation due to curvature. The arrows indicate the direction and magnitude of the force, and simultaneously the acceleration as these are connected through Eq. (2.13).

different wording in caption

Boundary Conditions

When a system is distributed in space,

For analysis purposes, an infinite domain may be adopted, but for implementation, a finite domain needs to be established.

Consider a system of length L (in m) $x \in \mathcal{D}$ (x is an element in $\mathcal{D}$) where domain $\mathcal{D} = [0, L]$ Two alternatives are

$$u(0,t) = u(L,t) = 0 \quad \text{(Dirichlet, fixed)}, \tag{2.14a}$$
$$\partial_x u(0,t) = \partial_x u(L,t) = 0 \quad \text{(Neumann, free)}, \tag{2.14b}$$

2.2.2 Discrete-time

The most straightforward discretisation of Eq. (2.13) is the following FD scheme

$$\delta_{tt} u_l^n = c^2 \delta_{xx} u_l^n. \tag{2.15}$$

Other schemes exist (such as implicit)

Stability

The parameters that define (2.19) are linked through a stability condition. The famous Courant-Friedrichs-Lewy or *CFL condition* is defined as [**?**]

$$\lambda \leq 1 \quad \text{where} \quad \lambda = \frac{ck}{h} \tag{2.16}$$

where

$$\lambda = \frac{ck}{h} \tag{2.17}$$

is referred to as the *Courant number*.

Regarding implementation, as the time step k is based on the sample rate and thus usually fixed and c is a user-defined wavespeed, it is useful to rewrite Eq. (2.16) to be in terms of the grid spacing h:

$$h \geq ck. \tag{2.18}$$

Dispersion

Implementation

$$u_l^{n+1} = 2u_l^n - u_l^{n-1} + \lambda^2 \left(u_{l+1}^n - 2u_l^n + u_{l-1}^n \right) \tag{2.19}$$

If $\lambda = 1$, (2.19) is an exact solution to Eq. (2.13). This is no

2.2.3 Output

Matrix form

2.2.4 Operators in Matrix Form

Finite-difference operators, such as δ_{x+}, δ_{x-} and $\delta_{x\cdot}$ can be written in matrix form:

$$\mathbf{D}_{x+} = \frac{1}{h}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & -1 & 1 & \\ & & & & -1 & \ddots \\ & \mathbf{0} & & & & \ddots \end{bmatrix} \quad \mathbf{D}_{x-} = \frac{1}{h}\begin{bmatrix} \ddots & & & & \mathbf{0} & \\ \ddots & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & -1 & 1 & \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix}$$

$$\mathbf{D}_{x\cdot} = \frac{1}{2h}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ \ddots & 0 & 1 & & & \\ & -1 & 0 & 1 & & \\ & & -1 & 0 & 1 & \\ & & & -1 & 0 & \ddots \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix}$$

The matrices $\mathbf{D}_{x+}$ and $\mathbf{D}_{x-}$ can be multiplied to get $\mathbf{D}_{xx}$:

$$\mathbf{D}_{xx} = \mathbf{D}_{x+}\mathbf{D}_{x-} = \frac{1}{h^2}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ \ddots & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & \ddots \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix} \tag{2.20}$$

and two $\mathbf{D}_{xx}$'s to get

$$\mathbf{D}_{xx}\mathbf{D}_{xx} = \mathbf{D}_{xxxx} = \frac{1}{h^4}\begin{bmatrix} 5 & -4 & 1 & & & \mathbf{0} & \\ -4 & 6 & \ddots & \ddots & & & \\ 1 & \ddots & \ddots & -4 & 1 & & \\ & \ddots & -4 & 6 & -4 & \ddots & \\ & & 1 & -4 & \ddots & \ddots & 1 \\ & & & \ddots & \ddots & 6 & -4 \\ & \mathbf{0} & & & 1 & -4 & 5 \end{bmatrix} \tag{2.21}$$

which is used for a stiff string with a simply supported boundary condition.

Averaging operators μ_{x+}, μ_{x-} and $\mu x\cdot$ are defined in a similar way:

$$\mathbf{M}_{x+} = \frac{1}{2}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & 1 & 1 & \\ & & & & 1 & \ddots \\ & \mathbf{0} & & & & \ddots \end{bmatrix} \qquad \mathbf{M}_{x-} = \frac{1}{2}\begin{bmatrix} \ddots & & & & \mathbf{0} & \\ \ddots & 1 & & & & \\ & 1 & 1 & & & \\ & & 1 & 1 & & \\ & & & 1 & 1 & \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix}$$

$$\mathbf{M}_{x\cdot} = \frac{1}{2}\begin{bmatrix} \ddots & \ddots & & & \mathbf{0} & \\ \ddots & 0 & 1 & & & \\ & 1 & 0 & 1 & & \\ & & 1 & 0 & 1 & \\ & & & 1 & 0 & \ddots \\ & \mathbf{0} & & & \ddots & \ddots \end{bmatrix}$$

Note the multiplication by $1/2$ rather than $1/h$ (or $1/2h$) for all operators.

Only spatial operators are written in this matrix form and then applied to state vectors at different time steps ($n+1$, n and $n-1$), examples of which can be found below.

Output sound

After the system is excited (see III), one can listen to the output

FFT

Waveform

The location, or frequency of the harmonic partials can also be analytically / numerically derived using modal analysis as will be explained in 2.5.

2.3 Energy Analysis

Debugging physical models.

2.3.1 Mathematical tools

Continuous-time

For two functions $f(x)$ and $g(x)$ and $x \in \mathcal{D}$ their l_2 inner product along with the l_2 norm is defined as

$$\langle f, g \rangle_{\mathcal{D}} \int_{\mathcal{D}} fg dx \quad \text{and} \quad \|f\|_{\mathcal{D}} = \sqrt{\langle f, f \rangle_{\mathcal{D}}} \tag{2.22}$$

Discrete-time

Inner product of any time series f^n and g^n and the discrete counterpart to (2.22) is

$$\langle f^n, g^n \rangle_{\mathcal{D}} = \sum_{l \in \mathcal{D}} h f_l^n g_l^n \tag{2.23}$$

where the multiplication by h is the discrete counterpart of dx the continuous definition.

2.4 von Neumann Analysis

Much literature gives the stability condition using an "it can be shown that" argument (fx. [2]). Here, I would like to take the opportunity to

Finding stability conditions for models

Sin identity:

$$\sin(x) = \frac{e^{jx} - e^{-jx}}{2j} \quad \Longrightarrow \quad \sin^2(x) = \frac{e^{j2x} - 2e^{jx-jx} + e^{-j2x}}{-4} = \frac{e^{j2x} + e^{-j2x}}{-4} + \frac{1}{2}. \tag{2.24}$$

Cos identity:

$$\cos(x) = \frac{e^{jx} + e^{-jx}}{2} \quad \Longrightarrow \quad \cos^2(x) = \frac{e^{j2x} + 2e^{jx-jx} + e^{-j2x}}{4} = \frac{e^{j2x} + e^{-j2x}}{4} + \frac{1}{2}. \tag{2.25}$$

2.5 Modal Analysis

Modes are the resonant frequencies of a system. The amout of modes that a discrete system contains dependends on the amount of moving points. A mass-spring system thus has one resonating mode, but a FD scheme of the 1D wave equation with N = 30 (including the boundaries) and Dirichlet boundary conditions will have 29 modes.

definition of modes

This section will show how to obtain the modes for an FD scheme implementing the 1D wave equation as done in Section . We start with Eq. (6.34) (using c instead of γ)

$$\delta_{tt} u = c^2 \delta_{xx} u, \tag{2.26}$$

which can be written in matrix form as

$$\frac{1}{k^2}\left(\mathbf{u}^{n+1} - 2\mathbf{u}^n + \mathbf{u}^{n-1}\right) = c^2 \mathbf{D}_{xx}\mathbf{u} \tag{2.27}$$

Following [3] we assume a solution of the form $\mathbf{u} = \boldsymbol{\phi} z^n$. Substituting this into Eq. (2.27) yields the characteristic equation

$$(z - 2 + z^{-1})\boldsymbol{\phi} = c^2 k^2 \mathbf{D}_{xx}\boldsymbol{\phi}. \tag{2.28}$$

This is an eigenvalue problem where the p'th solution is defined as

$$\begin{aligned} z_p - 2 + z_p^{-1} &= c^2 k^2 \text{eig}_p(\mathbf{D}_{xx}) \\ z_p + (-2 - c^2 k^2 \text{eig}_p(\mathbf{D}_{xx})) + z_p^{-1} &= 0 \end{aligned} \tag{2.29}$$

where $\text{eig}_p(\cdot)$ denoting the pth eigenvalue of '$\cdot$'. If the CFL condition for the scheme is satisfied, the roots will lie on the unit circle. Furthermore we can substitute a test solution $z_p = e^{j\omega_p k}$ solve for the eigenfrequencies:

$$\begin{aligned} e^{j\omega_p k} + e^{-j\omega_p k} - 2 - c^2 k^2 \text{eig}_p(\mathbf{D}_{xx}) &= 0 \\ \frac{e^{j\omega_p k} + e^{-j\omega_p k}}{-4} + \frac{1}{2} + \frac{c^2 k^2}{4}\text{eig}_p(\mathbf{D}_{xx}) &= 0 \end{aligned}$$

Then using Eq. (2.24) we get

$$\begin{aligned} \sin^2(\omega_p k/2) + c^2 k^2 \text{eig}_p(\mathbf{D}_{xx}) &= 0 \\ \sin(\omega_p k/2) &= ck\sqrt{-\text{eig}_p(\mathbf{D}_{xx})} \\ \omega_p &= \frac{2}{k}\sin^{-1}\left(ck\sqrt{-\text{eig}_p(\mathbf{D}_{xx})}\right) \end{aligned} \tag{2.30}$$

which is Eq. (6.53) in [3].

2.6 Dispersion analysis

Part II

Resonators

Resonators

Though the physical models described in the previous part are also considered resonators, they are *ideal* cases. In other words, you would not be able to find these “in the wild" as they do not includes effects such as losses or dispersion in the case of the 1D wave equation.

Scaling

Scaling, or non-dimensionalisation can be useful to reduce the parameter space

Domain $x \in [0, L]$ is scaled to $x' = x/L$ such that $x' \in [0, 1]$.

The 1D wave equation in (2.13) can be rewritten to

$$\partial_t^2 u = \gamma^2 \partial_{x'x'} u \tag{2.31}$$

where scaled wavespeed $\gamma = c/L$ has units of frequency.

Although this parameter reduction might be useful for resonators in isolation, when they are connected

Moreover, for the parameters to make

As, later on, different resonators in isolation will be connected, the overview is better kept when all parameters are written out

- Bars and Stiff Strings
- 2D Models
- Brass

Chapter 3

Stiff string

Consider the transverse displacement of a string of length L described by $u = u(x, t)$ defined for $x \in \mathcal{D}$ with domain $\mathcal{D} = [0, L]$ and time $t \geq 0$.

Stiff string and stuff Used in many of the publications in Part IX

3.1 Continuous time

Circular cross-sectional area

$$\rho A \partial_t^2 u = T \partial_x^2 u - EI \partial_x^4 u \tag{3.1}$$

parameterised by material density ρ (in kg/m^3), cross-sectional area $A = \pi r^2$ (in m^2), radius r (in m) tension T (in N), Young's modulus E (in Pa) and area moment of inertia $I = \pi r^4/4$ (in m^4). If either E or I is 0, Eq (3.1) reduces to the 1D wave equation in (2.13) where $c = \sqrt{T/\rho A}$. If instead $T = 0$, Eq. (3.1) reduces to the ideal bar equation.

The 4th-order spatial derivative models *frequency dispersion* a phenomenon that causes different frequencies to travel at different speeds.

3.1.1 Dispersion Analysis

Adding Losses

Before moving on to the discretisation of Eq. (3.1), losses are added

First appeared in [1]

$$\rho A \partial_t^2 u = T \partial_x^2 u - EI \partial_x^4 u - 2\sigma_0 \rho A \partial_t u + 2\sigma_1 \rho A \partial_t \partial_x^2 u \tag{3.2}$$

where the loss coefficients σ_0 and σ_1 describe the frequency dependent and frequency independent losses respectively.

A more compact way to write Eq. (3.4), and as is also found often in the literature [3] is to divide both sides by ρA to get

etc.

$$\partial_t^2 u = c^2 \partial_x^2 u - \kappa^2 \partial_x^4 u - 2\sigma_0 \partial_t u + 2\sigma_1 \partial_t \partial_x^2 u \tag{3.3}$$

check wavespeed or wave speed (entire document)

where $c = \sqrt{T/\rho A}$ is the wave speed (in m/s) as in the 1D wave equation in (2.13) and $\kappa = \sqrt{EI/\rho A}$ is a *stiffness coefficient* (in m^2/s).

Intuition

Although Eq. (3.4) might look daunting at first, the principle of Newton's second law remains the same.

Something about the 4th spatial derivative and the loss terms here...

Boundary Conditions

The boundary conditions found in (2.14) can be extended

3.2 Discrete Time

Equation (3.4) can be discretised as

$$\rho A \delta_{tt} u_l^n = T \delta_{xx} u_l^n - EI \delta_{xxxx} u_l^n - 2\sigma_0 \rho A \delta_{t\cdot} u_l^n + 2\sigma_1 \rho A \delta_{t-} \delta_{xx} u_l^n \tag{3.4}$$

The δ_{xxxx} operator is the the second-order spatial difference in Eq. (2.9) applied to itself.

$$\delta_{xxxx} = \delta_{xx}\delta_{xx} = \frac{1}{h^4}\left(e_{x+}^2 - 4e_{x+} + 6 - 4e_{x-} + e_{x-}^2\right) \tag{3.5}$$

3.2.1 Combining operators

$$\delta_{t-}\delta_{xx} = \frac{1}{k}\left(1 - e_{t-}\right)\frac{1}{h^2}\left(e_{x+} - 2 + e_{x-}\right) = \frac{1}{kh^2}\left(e_{x+} - 2 + e_{x-} - e_{t-}(e_{x+} - 2 + e_{x-})\right)$$

Taking the mixed derivative for the

$$\delta_{t-}\delta_{xx} u_l^n = \begin{cases} \frac{1}{k}\left(\delta_{xx} u_l^n - \delta_{xx} u_l^{n-1}\right) & \text{expanding } \delta_{t-} \\ \frac{1}{h^2}\left(\delta_{t-} u_{l+1}^n - 2\delta_{t-} u_l^n + \delta_{t-} u_{l-1}^n\right) & \text{expanding } \delta_{xx} \end{cases}$$

and after expansion of the second operator both result in

$$\delta_{t-}\delta_{xx} u_l^n = \frac{1}{hk^2}\left(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} - u_{l-1}^{n-1}\right) \tag{3.6}$$

Chapter 4

2D Systems

The 2D wave equation be used to model an ideal membrane such as done in Paper

In this work, it is mainly used to model a simplified body in papers...

State variable $u = u(x, y, t)$ where $t \geq 0$ and $(x, y) \in \mathcal{D}$ where $\mathcal{D}$ is 2-dimensional. The state variable can then be discretised according to $u(x, y, t) \approx u_{l,m}^n$ with space $x = lh$ and $y = mh$ and time $t = nk$ and $k = 1/f_\mathrm{s}$. For simplicity the grid spacing in both the x and y directions are set to be the same but could be different.

The same shift operators as defined in Chapter 2 can be applied to grid function $u_{l,m}^n$

$$e_{y+}u_{l,m}^n = u_{l,m+1}^n, \quad \text{and} \quad e_{y-}u_{l,m}^n = u_{l,m-1}^n. \tag{4.1}$$

Additional operators:

$$\Delta = \partial_x^2 + \partial_y^2 \tag{4.2}$$

4.1 2D Wave Equation

4.2 Thin plate

4.3 Stiff membrane

Chapter 5

Brass

This will be the first appearance of a first-order system.

Part III

Exciters

Exciters

Now that a plethora of resonators have been introduced in part II, different mechanisms to excite them will be introduced here. First, different examples of

- Simple pluck ((half) raised-cos)
- Hammer
- Bow Models
- Lip reed

Chapter 6

Unmodelled Excitations

Different title here?

6.1 Initial conditions

Hammer

Full raised cosine

Pluck

- Cut-off raised cosine
- Triangle (for string)

6.2 Signals

6.2.1 Pulse train

For brass

6.2.2 Noise

Noise input

Chapter 7

Modelled Excitations

7.1 Hammer

Hammer modelling

7.2 The Bow

The bow...

Helmholtz motion..

See fx. https://www.youtube.com/watch?v=6JeyiM0YNo4

Characteristic triangular motion (wave shapes?)

7.2.1 Static Friction Models

In static bow-string-interaction models, the friction force is defined as a function of the relative velocity between the bow and the string only. The first mathematical description of friction was proposed by Coulomb in 1773 [**?**] to which static friction, or *stiction*, was added by Morin in 1833 [**?**] and viscous friction, or velocity-dependent friction, by Reynolds in 1886 [**?**]. In 1902, Stribeck found a smooth transition between the static and the coulomb part of the friction curve now referred to as the Stribeck effect [**?**]. The latter is still the standard for static friction models today.

7.2.2 Dynamic Friction Models

As opposed to less complex bow models, such as the hyperbolic [source] and exponential [source] models, the elasto-plastic bow model assumes that the friction between the bow and the string is caused by a large quantity of bristles, each of which contributes to the total amount of friction.

7.3 Lip-reed

Lip-reed model

7.3.1 Coupling to Tube

Part IV

Interactions

The models described in part II already sound quite convincing on their own. However, these are just individual components that can be combined to approximate a fully functional (virtual) instrument. The following chapters will describe different ways of interaction between individual systems. Chapter 8 describes ways to connect different systems and Chapter 9 describes collision interactions between models.

ε

Newtons third law (action reaction)

Subscripts are needed

Somewhere in this Chapter have a section about fretting and how to generate different pitches using only one string

Interpolation and spreading operators

Using $l_\mathrm{c} = \lfloor x_\mathrm{c}/h \rfloor$ and $\alpha_\mathrm{c} = x_\mathrm{c}/h - l_\mathrm{c}$ is the fractional part the location of interest.

$$I_0(x_\mathrm{c}) = \begin{cases} 1, & \text{if } l = l_\mathrm{c}, \\ 0, & \text{otherwise} \end{cases} \tag{7.1}$$

$$I_1(x_\mathrm{c}) = \begin{cases} (1-\alpha_\mathrm{c}), & \text{if } l = l_\mathrm{c} \\ \alpha_\mathrm{c}, & \text{if } l = l_\mathrm{c}+1 \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

$$I_3(x_\mathrm{c}) = \Big\{ \ldots \tag{7.3}$$

The following identity is very useful when solving interactions between components:

$$\langle f, J_p(x_\mathrm{c}) \rangle_{\mathcal{D}} = I_p(x_\mathrm{c}) f \tag{7.4}$$

Chapter 8

Connections

Something about connections

Pointlike $\delta(x - x_c)$ or distributed E_c

8.1 Rigid connection

The simplest connection is Forces should be equal and opposite.

If component a is located 'above' component b, and their relative displacement is defined as $\eta = a - b$, then a positive η is going to have a negative effect on a and a positive effect on b and vice-versa. This is important for the signs when adding the force terms to the schemes.

8.2 Spring-like connections

8.2.1 Connection with rigid barrier (scaled)

Consider the (scaled) 1D wave equation with an additional force term F^n

$$\delta_{tt} u_l^n = \gamma^2 \delta_{xx} u_l^n + J(x_c) F^n \tag{8.1}$$

where

$$F^n = -\omega_0^2 \mu_{t\cdot} \eta^n - \omega_1^4 (\eta^n)^2 \mu_{t\cdot} \eta^n - 2\sigma_\times \delta_{t\cdot} \eta^n \tag{8.2}$$

and

$$\eta^n = I(x_c) u_l^n. \tag{8.3}$$

To obtain F^n, an inner product of scheme (8.1) needs to be taken with $J(x_c)$ over domain $\mathcal{D}$ which, using identity (7.4) yields

$$\delta_{tt} I(x_c) u_l^n = \gamma^2 I(x_c) \delta_{xx} u_l^n + \underbrace{I(x_c) J(x_c)}_{\|J(x_c)\|_{\mathcal{D}}^2} F^n. \tag{8.4}$$

As u is connected to a rigid barrier according to (8.3), a shortcut can be taken and Eqs. (8.2) and (8.3) can be directly substituted into Eq. (8.4) to get

$$\delta_{tt}\eta^n = \gamma^2 I(x_\mathrm{c})\delta_{xx}u_l^n + \|J(x_\mathrm{c})\|_{\mathcal{D}}^2\left(-\omega_0^2\mu_{t\cdot}\eta^n - \omega_1^4(\eta^n)^2\mu_{t\cdot}\eta^n - 2\sigma_\times\delta_{t\cdot}\eta^n\right). \tag{8.5}$$

and solved for η^{n+1}:

$$\begin{aligned}\Big(1 + \|J(x_\mathrm{c})\|_{\mathcal{D}}^2 k^2[\omega_0^2/2 + \omega_1^4(\eta^n)^2/2 + \sigma_\times/k]\Big)\eta^{n+1} \\ = 2\eta^n - \Big(1 + \|J(x_\mathrm{c})\|_{\mathcal{D}}^2 k^2[\omega_0^2/2 + \omega_1^4(\eta^n)^2/2 - \sigma_\times/k]\Big)\eta^{n-1} \\ + \gamma^2 k^2 I(x_\mathrm{c})\delta_{xx}u_l^n\end{aligned} \tag{8.6}$$

This can then be used to calculate F^n in (8.2) and can in turn be used to calculate u_l^{n+1} in (8.1).

8.2.2 String-plate connection

In this example, let's consider a string connected to a plate using a nonlinear damped spring. This could be interpreted as a simplified form of how guitar string would be connected to the body.

Continuous

The systems in isolation are as in (3.4) and (**??**), but with an added force term:

$$\partial_t^2 u = c^2\partial_x^2 u - \kappa_\mathrm{s}^2\partial_x^4 u - 2\sigma_{0,\mathrm{s}}\partial_t u + 2\sigma_{1,\mathrm{s}}\partial_t\partial_x^2 u - \delta(x - x_\mathrm{c})\frac{f}{\rho_\mathrm{s}A} \tag{8.7a}$$

$$\partial_t^2 w = -\kappa_\mathrm{p}^2\Delta\Delta w - 2\sigma_{0,\mathrm{p}}\partial_t w + 2\sigma_{1,\mathrm{p}}\partial_t\partial_x^2 w + \delta(x - x_\mathrm{c}, y - y_\mathrm{c})\frac{f}{\rho_\mathrm{p}H} \tag{8.7b}$$

where

$$f = f(t) = K_1\eta + K_3\eta^3 + R\dot{\eta} \tag{8.8}$$

and

$$\eta = \eta(t) = u(x_\mathrm{c}, t) - w(x_\mathrm{c}, y_\mathrm{c}, t) \tag{8.9}$$

Discrete

System (8.7) can then be discretised as

$$\delta_{tt}u_l^n = c^2\delta_{xx}u_l^n - \kappa_\mathrm{s}^2\delta_{xxxx}u_l^n - 2\sigma_{0,\mathrm{s}}\delta_{t\cdot}u_l^n + 2\sigma_{1,\mathrm{s}}\delta_{t-}\delta_{xx}u_l^n - J_\mathrm{s}(x_\mathrm{c})\frac{f^n}{\rho_\mathrm{s}A}\,, \tag{8.10}$$

$$\delta_{tt}w_l^n = -\kappa_\mathrm{p}^2\delta_\Delta\delta_\Delta w_l^n - 2\sigma_{0,\mathrm{p}}\delta_{t\cdot}w_l^n + 2\sigma_{1,\mathrm{p}}\delta_{t-}\delta_{xx}w_l^n + J_\mathrm{p}(x_\mathrm{c}, y_\mathrm{c})\frac{f^n}{\rho_\mathrm{p}H}\,, \tag{8.11}$$

where

$$f^n = K_1 \mu_{tt} \eta^n + K_3 (\eta^n)^2 \mu_{t\cdot} \eta^n + R \delta_{t\cdot} \eta^n, \tag{8.12}$$

and

$$\eta^n = I(x_\mathrm{c}) u_l^n - I(x_\mathrm{c}, y_\mathrm{c}) w_l^n. \tag{8.13}$$

Expansion

System (8.10) can be expanded at the connection location x_c by taking an inner product of the schemes with their respective spreading operators.

8.2.3 Solving for f

8.2.4 Non-dimensional

The scaled system can be written as:

$$\partial_t^2 u = \gamma^2 \partial_x^2 u - \kappa_\mathrm{s}^2 \partial_x^4 u - 2\sigma_{0,\mathrm{s}} \partial_t u + 2\sigma_{1,\mathrm{s}} \partial_t \partial_x^2 u - \delta(x - x_\mathrm{c}) F \tag{8.14}$$

$$\partial_t w = -\kappa_\mathrm{p}^2 \Delta\Delta w - 2\sigma_{0,\mathrm{p}} \partial_t w + 2\sigma_{1,\mathrm{p}} \partial_t \partial_x^2 w + \delta(x - x_\mathrm{c}, y - y_\mathrm{c}) F \tag{8.15}$$

where

$$F = F(t) = \omega_1^2 \eta + \omega_3^4 \eta^3 + \sigma_\mathrm{c} \dot{\eta} \tag{8.16}$$

and

$$\eta = \eta(t) = u(x_\mathrm{c}, t) - w(x_\mathrm{c}, y_\mathrm{c}, t) \tag{8.17}$$

$$I(x_\mathrm{c}) \delta_{tt} u_l^n = c^2 \left(I(x_\mathrm{c}) \delta_{xx} u_l^n \right) + I(x_\mathrm{c}) J(x_\mathrm{c}) F \tag{8.18}$$

Chapter 9

Collisions

Something about collisions

9.1 Classic models

Note that when using Eq. (7.37) in [3]

9.2 Michele's tricks

Part V

Dynamic Grids

Chapter 10

Dynamic Grids

Often in math, you should view the definition not as a starting point, but as a target. Contrary to the structure of textbooks, mathematicians do not start by making definitions and then listing a lot of theorems, and proving them, and showing some examples. The process of discovering math typically goes the other way around. They start by chewing on specific problems, and then generalising those problems, then coming up with constructs that might be helpful in those general cases,and only then you write down a new definition (or extend an old one). - Grant Sanderson (AKA 3Blue1Brown) https://youtu.be/O85OWBJ2ayo?t=359

10.1 Background and Motivation

Simulating musical instruments using physical modelling – as mentioned in Part I– allows for manipulations of the instrument that are impossible in the physical world. Examples of this are changes in material density or stiffness, cross-sectional area (1D), thickness (2D) and size. Apart from being potentially sonically interesting, there are examples in the physical world where certain aspects of the instrument are manipulated in real-time.

check if is still true

Tension in a string is changed when tuning it

Some artists even use this in their performances [8, 9]

The hammered dulcimer is another example where the strings are tensioned over a bridge where one can play the string at one side of the bridge, while pushing down on the same string on the other side [7].

1D:

- Trombone
- Slide whistle
- Guitar strings

  - Fretting finger pitch bend
  - Above the nut [9]
  - Tuning pegs directly [8]
- Hammered dulcimer [7]
- Erhu?

2D:

- Timpani
- Bodhrán: https://youtu.be/b9HyB5yNS1A?t=146
- Talking drum (hourglass drum): https://youtu.be/B4oQJZ2TEVI?t=9
- Flex-a-tone (could also be 1D tbh..): https://www.youtube.com/watch?v=HEW1aG8XJQ

A more relevant example is that of the trombone, where the size of the instrument is changed in order to play different pitches. Modelling this using FDTD methods would require

In his thesis, Harrison points out that in order to model the trombone, grid points need to be introduced

Something about time-dependent variable coefficient Stokes flow: https://arxiv.org/abs/10

Time-varying propagation speed in waveguides: https://quod.lib.umich.edu/cgi/p/pod/idx/fractional-delay-application-time-varying-propagation-speed.pdf?c=icmc;idno=bbp2372.

Special boundary conditions (look at!): Modeling of Complex Geometries and Boundary Conditions in Finite Difference/Finite Volume Time Domain Room Acoustics Simulation (`https://www.researchgate.net/publication/260701231_Modeling_of_Complex_Geometries_and_Boundary_Conditions_in_Finite_DifferenceFinite_Volume_Time_Domain_Room_Acoustics_Simulation`)

10.2 Method

Iterations have been:

- Interpolated boundary conditions
- Linear interpolation

These sections are taken from the JASA appendix

In this appendix, some iterations done over the course of this project will be shown in more detail. In the following, the 1D wave equation with a wave speed of $c = 1470$ m/s, a length of $L = 1$ m, Dirichlet boundary conditions and a sample rate of $f_s = 44100$ Hz is considered, and – through Eq. (**??**) – satisfies the CFL condition with equality. These values result in $N = 30$, or a

grid of 31 points including the boundaries. Then, the wave speed is decreased to $c \approx 1422.6$ m/s, i.e., the wave speed that results in $N = 31$ and satisfies the stability condition with equality again.

10.2.1 Full-Grid Interpolation

One way to go from one grid to another is performing a full-grid interpolation [3, Chap. 5]. If the number of points changes according to Eq. (**??**), i.e., if $N^n \neq N^{n-1}$ the full state of the system ($u_l^n, u_l^{n-1}\ \forall l$) can be interpolated to the new state. See Figure 10.1.

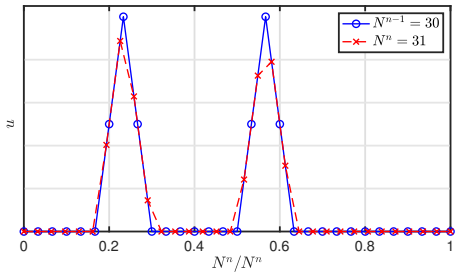

Fig. 10.1: Upsampling u (with an arbitrary state) using (linear) full-grid interpolation with $N^{n-1} = 30$ and $N^n = 31$. The horizontal axis is normalised with respect to N^n.

An issue that arises using this method is that the Courant number λ will slightly deviate from the CFL condition as c changes. Using Eq. (**??**) with L/ck approaching 31 (from below), the minimum value of $\lambda \approx 30/31 \approx 0.9677$. This, employing Eq. (**??**), has a maximum frequency output of $f_{\text{max}} \approx 18,475$ Hz. The Courant number will deviate more for higher values of c and thus lower values for N – for instance, if N approaches 11 (from below), $\lambda \approx 10/11 \approx 0.9091$ and $f_{\text{max}} \approx 16,018$ Hz.

Another problem with full-grid interpolation, is that it has a low-passing effect on the system state, and thus on the output sound. Furthermore, this state-interpolation causes artefacts or 'clicks' in the output sound as the method causes sudden variations in the states.

All the aforementioned issues could be solved by using a (much) higher sample rate and thus more grid points, but this would render this method impossible to work in real time.

10.2.2 Adding and removing Points at the Boundary

To solve the issues exhibited by a full-grid interpolation, points can be added and removed at a single location and leave most points unaffected by the parameter changes. A good candidate for a location to do this is at a fixed

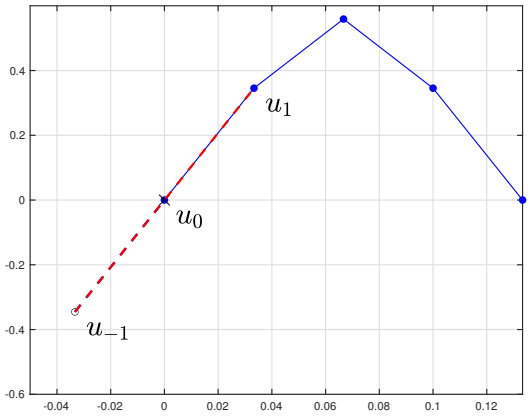

Fig. 10.2: The simply supported boundary condition: both the state and the curvature at the boundary – at $l = 0$ – should be 0.

(Dirichlet) boundary. The state u at this location is always 0 so points can be added smoothly.

As c decreases, h can be calculated according to Eq. (**??**) and decreases as well.

This has a physical analogy with tuning a guitar string. Material enters and exits the neck (playable part of the string) at the nut, which in discrete time means grid points appearing and disappearing at one boundary.

To yield smooth changes between grid configurations, an interpolated boundary has been developed, the possibility of which has been briefly mentioned in [3, p. 145]. The Dirichlet condition in Eq. (2.14a) can be extended to be the simply supported boundary condition:

$$u(x,t) = \frac{\partial^2}{\partial x^2} u(x,t) = 0 \quad \text{where} \quad x = 0, L, \tag{10.1}$$

or, when discretised,

$$u_l^n = \delta_{xx} u_l^n = 0, \quad \text{where} \quad l = 0, N. \tag{10.2}$$

This means that on top of that the state of the boundary should be 0, the curvature around it should also be 0. One can again solve for the virtual grid points at the boundary locations, yielding

$$u_{-1}^n = -u_1^n \quad \text{and} \quad u_{N+1}^n = -u_{N-1}^n. \tag{10.3}$$

This is visualised in Figure 10.2.

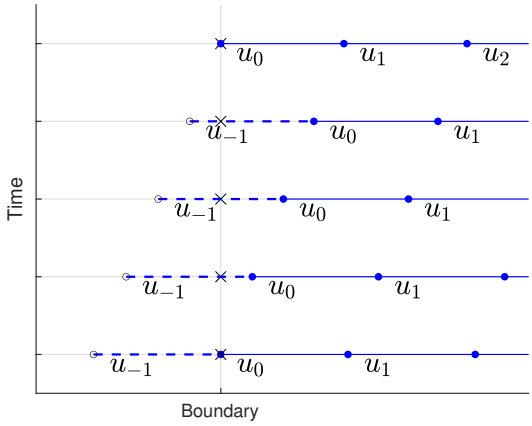

Fig. 10.3: The grid changing over time

If the flooring operation in Eq. (**??**) is removed this introduces a fractional number of grid points.

The by-product of using a fractional N this is that the CFL condition in (2.16) can now always be satisfied with equality no matter what the wave speed is.

An issue with this method is that removing points is much harder than adding.

their interactions change through a change in the grid spacing and wave speed. This interaction, though, is defined by λ which

10.2.3 Cubic interpolation

10.2.4 Sinc interpolation

10.2.5 Displacement correction

The displacement correction can be interpreted as a spring force pulling u_M^n and w_0^n to the average displacement.

$$
\begin{aligned}
u_M^{n+1} &= 2u_M^n + \lambda^2(u_{M-1}^n - 2u_M^n + u_{M+1}^n) - K\left(u_M^n - \frac{u_M^n + w_0^n}{2}\right) \\
w_0^{n+1} &= 2u_M^n + \lambda^2(w_{-1}^n - 2w_0^n + w_1^n) - K\left(w_0^n - \frac{u_M^n + w_0^n}{2}\right)
\end{aligned} \tag{10.4}
$$

$$
\begin{aligned}
u_M^{n+1} &= 2u_M^n + \lambda^2(u_{M-1}^n - 2u_M^n + u_{M+1}^n) + \frac{K}{2}(w_0^n - u_M^n) \\
w_0^{n+1} &= 2u_M^n + \lambda^2(w_{-1}^n - 2w_0^n + w_1^n) - \frac{K}{2}(w_0^n - u_M^n)
\end{aligned} \tag{10.5}
$$

with $K = K(\alpha)$

$$K = (1 - \alpha)^{\epsilon}. \tag{10.6}$$

10.3 Analysis and Experiments

10.3.1 Interpolation technique

10.3.2 Interpolation range

10.3.3 Location

... where to add and remove points
Using the whole range, we can still add/remove points at the sides.

10.4 Discussion and Conclusion

Part VI

Real-Time Implementation and Control

Real-Time Implementation and Control

It's all fun and dandy with all these physical models, but what use are they if you can't control them in real time?!

Chapter 11

Real-Time Implementation

JUCE Give overall structure of code

Implementation of the physical models using FDTD methods

As mentioned in Chapter 1, FDTD methods are used for high-quality and accurate simulations, rather than for real-time applications. This is due to their lack of simplifications.

Usually, `MATLAB` is used for simulating

Here, an interactive application is considered real-time when

Control of the application generates or manipulates audio with no noticeable latency.

Also the application needs to be controlled continuously

Helps to (informally) evaluate the models by interacting with it in a natural way (rather than static parameters)

11.1 MATLAB vs. C++

It is usually a good idea to prototype a physical modelling application in MATLAB for several reasons:

- Easier to debug
  - Plotting functionality
  - No need for memory handling
- Instability due to programming errors

11.1.1 Speed

Here is where the power of C++

11.1.2 Syntax

Indexing in Matlab is 1-based, meaning that the index of a vector starts at 1. If `u` is a vector with 10 elements, the first element is retrieved as `u(1)` and the last as `u(10)`. C++, on the other hand, is 0-based and retrieving the first and last element of a size-10 vector happens through `u[0]` and `u[9]` respectively.

11.2 Do's and don'ts in Real-Time FD schemes

Some of the things I learned (the hard way)...

- Create a limiter
- Structure your application into classes
- Use pointer switches
- Comment your code (hehe)

Create a limiter

Programming errors happen. To save your speakers, headphones or – most importantly – your ears, create a limiter.

```
double limit (double val)
{
    if (val < -1)
    {
        val = -1;
        return val;
    }
    else if (val > 1)
    {
        val = 1;
        return val;
    }
    return val;
}
```

```
for  i = 0:lengthSound
    uNext = ...
end
```

Use pointer switches

One of the most important things in working with FD schemes for real-time audio applications is to

```
for n = 1:lengthSound
    ...
    uPrev = u;
    u = uNext;
end
```

In C++ this is done using

```
double updateStates()
{
    double* uTmp = u[2];
    u[2] = u[1];
    u[1] = u[0];
    u[0] = uTmp;
}
```

Chapter 12

Control

12.1 Sensel Morph

150 Hz

12.2 Phantom OMNI

Part VII

Complete Instruments

Complete Instruments

This part will give several examples of full instrument models that have been developed during the PhD. Chapter **??** shows a three instrument-inspired case-studies using a large-scale modular environment, Chapter 14 describes the implementation of the tromba marina and Chapter 15 that of the trombone.

Chapter 13

Large Scale Modular Physical models

In the paper "Real-Time Control of Large-Scale Modular Physical Models using the Sensel Morph" [**?**] we presented a modular physical modelling environment using three instruments as case studies.

13.1 Bowed Sitar

- Stiff String
- Thin Plate
- Pluck
- Bow
- Non-linear spring connections

13.2 Dulcimer

- Stiff String
- Thin Plate
- Hammer (simple)
- Non-linear spring connections

13.3 Hurdy Gurdy

- Stiff String
- Thin Plate
- Pluck
- Bow
- Non-linear spring connections

Chapter 14

Tromba Marina

14.1 Introduction

14.2 Physical Model

14.2.1 Continuous

14.2.2 Discrete

14.3 Real-Time Implementation

14.3.1 Control using Sensel Morph

14.3.2 VR Application

Chapter 15

Trombone

Published in [I]

15.1 Introduction

Interesting read: https://newt.phys.unsw.edu.au/jw/brassacoustics.html

15.2 Physical Model

Most has been described in Chapter 5

15.2.1 Continuous

Just to save the conversation with Stefan about Webster's equation:

Using operators ∂_t and ∂_x denoting partial derivatives with respect to time t and spatial coordinate x, respectively, a system of first-order PDEs describing the wave propagation in an acoustic tube can then be written as

$$\frac{S}{\rho_0 c^2}\partial_t p = -\partial_x(Sv) \tag{15.1a}$$

$$\rho_0 \partial_t v = -\partial_x p \tag{15.1b}$$

with acoustic pressure $p = p(x,t)$ (in N/m^2), particle velocity $v = v(x,t)$ (in m/s) and (circular) cross-sectional area $S(x)$ (in m^2). Furthermore, ρ_0 is the density of air (in kg/m^3) and c is the speed of sound in air (in m/s). System (15.1) can be condensed into a second-order equation in p alone, often referred to as Webster's equation [**?**]. Interesting! In NSS it is the acoustic potential right? Can you go from that to a second-order PDE in p? There is a time-derivative hidden there somewhere right? (Just wondering :))Yes, the form

in p alone is the one you usually see. You get it by differentiating the first equation, giving you a $\dot{v}$ on the RHS, and then you can substitute the second equation in...I used the velocity potential one because it has direct energy balance properties. Right. So Webster's eq. in p and Ψ are identical (will exhibit identical behaviour), except for the unit of the state variable..?yes that's right...using the velocity potential allows you to do all the energy analysis easily, in terms of physical impedances. But the scheme you get to in the end is the same, just one derivative down. Alright cool! Thanks for the explanation :) For simplicity, effects of viscothermal losses have been neglected in (15.1). For a full time domain model of such effects in an acoustic tube, see, e.g. [**?**].

15.2.2 Discrete

15.3 Real-Time Implementation

Unity??

15.4 Discussion

more for your info, don't think I want to include this: To combat the drift, experiments have been done involving different ways of connecting the left and right tube. One involved alternating between applying the connection to the pressures and the velocity. Here, rather than adding points to the left and right system in alternating fashion, points were added to pressures p and q and velocities v and w in an alternating fashion. Another experiment involved a "staggered" version of the connection where (fx.) for one system (either left or right), a virtual grid point of the velocity was created from known values according to (**??**), rather than both from pressures. This, however, showed unstable behaviour. No conclusory statements can be made about these experiments at this point. ← which is exactly why I don't want to include this section

As the geometry varies it matters a lot where points are added and removed as this might influence the way that the method is implemented. speculative section coming up The middle of the slide crook was chosen, both because it would be reasonable for the air on the tube to "go away from" or "go towards" that point as the slide is extended or contracted, and because the geometry does not vary there. Experiments with adding / removing grid points where the geometry varies have been left for future work. even more speculative.. → It could be argued that it makes more sense to add points at the ends of the inner slides as "tube material" is also added there. This would mean that the system should be split in three parts: "inner slide", "outer slide" and "rest", and would complicate things even more.

check whether all references are used

References

[1] J. Bensa, S. Bilbao, R. Kronland-Martinet, and J. O. Smith, "The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides," *Journal of the Acoustical Society of America (JASA)*, vol. 114, no. 2, pp. 1095—1107, 2003.

[2] S. Bilbao, "A modular percussion synthesis environment," in *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, 2009.

[3] ——, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. John Wiley & Sons, 2009.

[4] S. Bilbao, C. Desvages, M. Ducceschi, B. Hamilton, R. Harrison-Harsley, A. Torin, and C. Webb, "Physical modeling, algorithms, and sound synthesis: The ness project," *Computer Music Journal*, vol. 43, no. 2-3, pp. 15–30, 2019.

[5] S. Bilbao, J. Perry, P. Graham, A. Gray, K. Kavoussanakis, G. Delap, T. Mudd, G. Sassoon, T. Wishart, and S. Young, "Large-scale physical modeling synthesis, parallel computing, and musical experimentation: The ness project in practice," *Computer Music Journal*, vol. 43, no. 2-3, pp. 31–47, 2019.

[6] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer, 1998.

[7] T. L. Glenn, "Amazing hammered dulcimer musician - joshua messick," 2014. [Online]. Available: https://youtu.be/veuGTnzgNRU?t=215

[8] J. Gomm, "Passionflower," 2011. [Online]. Available: https://www.youtube.com/watch?v=nY7GnAq6Znw

[9] J. Mayer, "Gravity (live in l.a.)," 2008. [Online]. Available: https://youtu.be/dBFW8OvciIU?t=284

[10] F. Mittelbach, *The LATEX companion*, 2nd ed. Addison-Wesley, 2005.

[11] G. E. Moore, "Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114—117, 1965.

Part VIII

Appendix

Appendix A

List of Symbols

The list of symbols found below contains often-used symbols in the thesis in the context that they are normally used. Depending on the context they might carry a different meaning (y being displacement of the lip-reed in Chapter 15 but the vertical spatial coordinate for 2D systems in fx. Chapter 4). Some might also be accompanied by a subscript in the main document

Symbol	Description	Unit
α	Fractional part of	
A	Cross-sectional area of string	m^2
c	Wave speed	$\mathrm{m/s}$
$\frac{d^n}{dt^n}$	n^{th} order derivative with respect to t	-
∂_t^n	n^{th} order partial derivative with respect to t	-
δ_{t+}, δ_{t-}, $\delta_{t\cdot}$	Forward, backward and centred difference in time	-
δ_{x+}, δ_{x-}, $\delta_{x\cdot}$	Forward, backward and centred difference in space	-
μ_{t+}, μ_{t-}, $\mu_{t\cdot}$	Forward, backward and centred average in time	-
μ_{x+}, μ_{x-}, $\mu_{x\cdot}$	Forward, backward and centred average in space	-
E	Young's Modulus	Pa ($\mathrm{kg{\cdot}m^{-1}{\cdot}s^{-2}}$)
f	Force or frequency	N or Hz
f_s	Sample rate	Hz
F	Scaled force	depends on system

Symbol	Description	Unit
h	Grid spacing	m
H	Membrane / Plate thickness	m
I	Area moment of inertia	m^4
l	Spatial index to grid function	-
L	Length	m
k	Time step ($= 1/f_\mathrm{s}$)	s
K	Spring coefficient	N/m
κ	Stiffness coefficient	m^2/s (1D) $\mathrm{m}^4{\cdot}\mathrm{s}^{-2}$ (2D)
n	Sample index to grid function	-
N	Number of points string	-
t	Time	s
T	Tension	N (N/m in 2D)
u	State variable	m
x	Spatial dimension (horizontal for 2D systems)	m
y	Vertical spatial dimension	m
γ	Scaled wave speed	s^{-1}
λ	Courant number for 1D wave eq. ($= ck/h$)	-
μ	Stiffness free parameter	-
ν	Poisson's ratio	-
η	Relative displacement spring	m
ρ	Material density	$\mathrm{kg}{\cdot}\mathrm{m}^{-3}$

Subscripts

c	Connection
p	Plate
s	String

Appendix B

List of Abbreviations

Abbreviation	Definition
FDS	Finite-difference scheme
FDTD	Finite-difference time-domain

Part IX

Papers

Paper Errata

Here, some errors in the published papers will be listed:
Real-Time Tromba [E]:

- The minus sign in Eq. (28) (and thus Eqs. (31) and (35)) should be a plus sign.
- $\sigma_{1,\mathrm{s}}$ in Eq. (21) should obviously be $\sigma_{1,\mathrm{p}}$
- the unit of the spatial Dirac delta function δ should be m^{-1}

DigiDrum [G]:

- σ_0 and σ_1 should be multiplied by ρH in order for the stability condition to hold.
- stability condition is wrong. Should be:

$$h \geq \sqrt{c^2k^2 + 4\sigma_1 k + \sqrt{(c^2k^2 + 4\sigma_1 k)^2 + 16\kappa^2 k^2}} \tag{1}$$

- Unit for membrane tension is $\mathrm{N/m}$.

Dynamic grids [H]

- Reference in intro for 'recently gained popularity' should go to [4]
  Note: not really an error, but should be changed before resubmission

Paper A

Paper A title

List of authors

The paper has been published in the
Journal or Proceedings Vol. XX(X), pp. XXX–XXX, 201X.

The layout has been revised.

Abstract

Here is an abstract.

A Introduction

Here is an introduction [10].

B Conclusion

Here is the conclusion.

Paper B

Real-Time Control of Large-Scale Modular Physical Models using the Sensel Morph

REAL-TIME CONTROL OF LARGE-SCALE MODULAR PHYSICAL MODELS USING THE SENSEL MORPH

Silvin Willemsen, Nikolaj Andersson, Stefania Serafin
Multisensory Experience Lab, CREATE,
Aalborg University Copenhagen
Copenhagen, Denmark
{sil, nsa, sts}@create.aau.dk

Stefan Bilbao
Acoustics and Audio Group
University of Edinburgh
Edinburgh, UK
s.bilbao@ed.ac.uk

ABSTRACT

In this paper, implementation, instrument design and control issues surrounding a modular physical modelling synthesis environment are described. The environment is constructed as a network of stiff strings and a resonant plate, accompanied by user-defined connections and excitation models. The bow, in particular, is a novel feature in this setting. The system as a whole is simulated using finite difference (FD) methods. The mathematical formulation of these models is presented, alongside several new instrument designs, together with a real-time implementation in JUCE using FD methods. Control is through the Sensel Morph.

1. INTRODUCTION

Physical models for sound synthesis have been researched for several decades to mathematically simulate the sonic behaviour of musical instruments and everyday sounds. Various techniques and methodologies have developed, ranging from mass-spring models [1–3] to modal synthesis [4] and waveguide based models [5]. The latter two techniques may be viewed as numerical simulation techniques applied to the systems of partial differential equations (PDEs). These equations define the dynamics of a musical instrument, either real or imagined.

Mainstream time-domain simulation techniques, such as finite difference (FD) methods, were first applied to the case of string vibration by Ruiz [6] and Hiller and Ruiz [7, 8], and then later by other authors [9] including, most notably Chaigne [10] and Chaigne and Askenfelt [11]. The general use of finite-difference schemes (FDSs) in sound synthesis is described in [12]. Modularized physical modelling sound synthesis, whereby the user may construct a virtual instrument using basic canonical components dates back to the work of Cadoz and collaborators [1–3]. It has been also used as a design principle in the context of FD methods [13–15], where the canonical elements are strings and plates, with a non-linear connection mechanism. Though computational cost of such methods is high, standard computing power is now approaching a level suitable for real-time performance for simpler systems.

We are interested in bridging the gap between large-scale modular physical modelling synthesis and sonic interaction design [16], to be able to play with such simulations in real-time. Specifically, we are interested in using the expressivity of the Sensel Morph [17] to control our simulations, using both percussive and bowing excitations. Our ultimate goal is to create models that are both mathematically accurate and efficient. This goal is nowadays possible thanks to improvements in hardware and software technologies for sound synthesis, yet it has rarely been achieved. The ultimate goal is to provide a modular efficient synthesizer based on accurate simulations, where real-time expressivity can also be achieved. This synthesizer has already been informally evaluated by composers and sound designers, who appreciated the current sonic palette.

This paper is structured as follows: Section 2 describes the physical models used in the implementation and Section 3 shows a general description of the FD methods used to digitally implement these models. Furthermore, Section 4 elaborates on the real-time implementation, Section 5 shows several different configurations of the physical models inspired by real musical instruments, Section 6 will present the results on CPU usage and evaluation and discuss this and finally, in Section 7, some concluding remarks appear.

2. MODELS

In this section, the PDEs for the damped stiff string and plate will be presented. The notation used will be the one found in [12] where the subscript for state variable u denotes a single derivative with respect to time t or space x respectively. Furthermore, to simplify the presented physical models, non-dimensionalization (or scaling) will be used [12].

2.1 Stiff string

A basic model of the linear transverse motion of a string of circular cross section may be described in terms of several parameters: the total length L (in m), the material density ρ (in kg·m^{-3}), string radius r (in m), Young's modulus E (in Pa), tension T (in N), and two loss parameters σ_0 and σ_1.

The PDE for a damped stiff string may be written as [12]

$$u_{tt} = \gamma^2 u_{xx} - \kappa^2 u_{xxxx} - 2\sigma_0 u_t + 2\sigma_1 u_{txx}. \quad (1)$$

In this representation, spatial scaling has been employed using a length L, so the solution $u = u(x,t)$ is defined for $t \geq 0$ and for dimensionless coordinate $x \in [0,1]$. Furthermore, parameters $\gamma = \sqrt{T/\rho\pi r^2 L^2}$ and $\kappa = \sqrt{Er^2/4\rho L^4}$ and have units s^{-1}.

In this work, the string is assumed clamped at both ends, so that

$$u = u_x = 0 \quad \text{where} \quad x = \{0,1\}. \quad (2)$$

A model of a bowed string [12] may be incorporated into (1) as

$$u_{tt} = \ldots - \delta(x - x_\mathrm{B}) F_\mathrm{B} \phi(v_\mathrm{rel}), \quad \text{with} \quad (3a)$$
$$v_\mathrm{rel} = u_t|_{(x=x_\mathrm{B})} - v_\mathrm{B}, \quad (3b)$$

where $F_\mathrm{B} = f_\mathrm{B}/M_\mathrm{s}$ is the excitation function (in $\mathrm{m/s}^2$) with externally-supplied bowing force $f_\mathrm{B} = f_\mathrm{B}(t)$ (in N) and total string mass $M_\mathrm{s} = \rho\pi r^2 L$ (in kg). The relative velocity v_rel is defined as the difference between the velocity of the string at bowing point x_B and the externally-supplied bowing velocity $v_\mathrm{B} = v_\mathrm{B}(t)$ (in m/s) and ϕ is a dimensionless friction characteristic, chosen here as [12]

$$\phi(v_\mathrm{rel}) = \sqrt{2a} v_\mathrm{rel} e^{-a v_\mathrm{rel}^2 + 1/2}. \quad (4)$$

Furthermore, $\delta(x - x_\mathrm{B})$ is a spatial Dirac delta function selecting the bowing location $x = x_\mathrm{B}$. The single bowing point can be extended to a bowing area [12]. More detailed models of string dynamics, again in a bowed string context, have been proposed by Desvages [18].

Another, and more simple way to excite the string is by extending Equation (1) to

$$u_{tt} = \ldots + E_\mathrm{e} F_\mathrm{e} \quad (5)$$

using an externally-supplied distribution function $E_\mathrm{e} = E_\mathrm{e}(x)$ and excitation function $F_\mathrm{e} = F_\mathrm{e}(t)$. In this case, the excitation region is allowed to be of finite width.

2.2 Plate

Under linear conditions, a rectangular plate of dimensions L_x and L_y may be parameterized in terms of density ρ (in $\mathrm{kg \cdot m}^{-3}$), thickness H (in m), Young's modulus E (in Pa) and a dimensionless Poisson's ratio ν, as well as two loss parameters σ_0 and σ_1.

In terms of dimensionless spatial coordinates x and y scaled by $\sqrt{L_x L_y}$, the equation of motion of a damped plate is a variant of the Kirchhoff model [19]

$$u_{tt} = -\kappa^2 \Delta\Delta u - 2\sigma_0 u_t + 2\sigma_1 \Delta u_t. \quad (6)$$

Here, $u(x,y,t)$ is the transverse displacement of the plate as a function of dimensionless coordinates $x \in [0, \sqrt{a}]$, $y \in [0, 1/\sqrt{a}]$, where $a = L_x/L_y$ is the plate aspect ratio, as well as time t. Furthermore, Δ represents the 2D Laplacian [12]:

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \quad (7)$$

The stiffness parameter κ, with dimensions of s^{-1}, is defined by $\kappa = \sqrt{D/\rho H L_x^2 L_y^2}$ where $D = EH^3/12\left(1-\nu^2\right)$. As in the case of the stiff string, we chose to use clamped boundary conditions:

$$u = \mathbf{n} \cdot \nabla u = 0 \quad (8)$$

over any plate edge with outward normal direction $\mathbf{n}$ and where ∇u is the gradient of u.

2.3 Connections

Adding connections between different physical models, further referred to as elements, adds another term to Equation (3a), (5) or (6). Assuming that element α is a stiff string and β is a plate, the following terms are added to the aforementioned equations:

$$u_{tt} = ... + E_{\mathrm{c},\alpha} F_\alpha, \quad (9a)$$
$$u_{tt} = ... + E_{\mathrm{c},\beta} F_\beta, \quad (9b)$$

with force-functions $F_\alpha = F_\alpha(t)$ and $F_\beta = F_\beta(t)$ (in $\mathrm{m/s}^2$) and distribution functions $E_{\mathrm{c},\alpha}$ and $E_{\mathrm{c},\beta}$ which have chosen to be highly localised in our application and reduce to $\delta(x - x_{\mathrm{c},\alpha})$ and $\delta(x - x_{\mathrm{c},\beta}, y - y_{\mathrm{c},\beta})$ respectively, but can be extended to be connection areas [13]. We use the implementation as presented in [13] where the connection between two elements is a non-linear spring. The forces it imposes on the elements it connects are defined as

$$F_\alpha = -\omega_0^2 \eta - \omega_1^4 \eta^3 - 2\sigma_\times \dot{\eta}, \quad (10a)$$
$$F_\beta = -\mathcal{M} F_\alpha, \quad (10b)$$

where ω_0 and ω_1 are the linear (in s^{-1}) and non-linear (in $(\mathrm{m \cdot s})^{-1/2}$) frequencies of oscillation respectively, $\sigma_\times$ is a damping factor (in s^{-1}), $\mathcal{M}$ is the mass ratio between the two elements and η is the relative displacement between the connected elements at the point of connection (in m). Lastly, the dot above η denotes a derivative with respect to time.

3. FINITE-DIFFERENCE SCHEMES

To be able to digitally implement the continuous equations shown in the previous section, they need to be approximated. In this section, a high-level review of a finite difference approximation to a connected system of strings and plates is presented. For more technical details, see [13].

In the case of the stiff string, state variable $u(x,t)$ can be discretised at times $t = nk$, where $n \in \mathbb{N}$ and $k = 1/f_\mathrm{s}$ is the time step (at sample-rate f_s) and locations $x = lh$, with $l \in [0, \ldots, N]$ where the total number of points is $N+1$ and grid spacing h. We can now write the discretised state variable as u_l^n, representing an approximation to $u(x,t)$.

In the case of the plate, $u(x,y,t)$ is discretised to $u_{(l,m)}^n$ using $x = lh$ where $l \in [0, \ldots, N_x]$ with $N_x + 1$ being the total horizontal number of points and $y = mh$ where $m \in [0, \ldots, N_y]$ with $N_y + 1$ being the total vertical number of points.

In a general sense, when discretising PDEs as presented in Equations (1) and (6), we will need to solve for $\mathbf{u}^{n+1}$,

i.e., $\mathbf{u}$ at the next time step, where $\mathbf{u}$ is a vector of size $N-1$ containing values of $u_l\ \forall l$ for a string and $(N_x-1)(N_y-1)$ containing values of $u_{(l,m)}\ \forall (l,m)$ for a plate. Note that the vector sizes are smaller than the total number of grid points as we do not include the values at the boundaries (which are always 0). For a PDE expressed as a function of u_{tt}, its FDS will be of the form

$$\mathbf{u}^{n+1} = 2\mathbf{u}^n - \mathbf{u}^{n-1} + K\boldsymbol{\mathcal{F}}^n, \tag{11}$$

where

$$K = \frac{k^2}{1+\sigma_0 k}, \tag{12}$$

and $\boldsymbol{\mathcal{F}}^n$ is a combination of the discretised PDE (excluding terms containing u^{n+1}) together with connection and excitation terms.

3.1 Stiff String

In the case of the stiff string, $\boldsymbol{\mathcal{F}}^n$ in Equation (11) is a combination of the discretised PDE (1) $\mathbf{f}_\alpha^n$, connection term (9a) and bowing (3a)

$$\boldsymbol{\mathcal{F}}^n = \mathbf{f}_\alpha^n + \mathbf{E}_{\mathrm{c},\alpha}F_\alpha^n - \mathbf{J}(x_\mathrm{B}^n)F_\mathrm{B}^n\phi(v_\mathrm{rel}), \tag{13a}$$

or excitation (5) term

$$\boldsymbol{\mathcal{F}}^n = \mathbf{f}_\alpha^n + \mathbf{E}_{\mathrm{c},\alpha}F_\alpha^n + \mathbf{E}_\mathrm{e}F_\mathrm{e}^n, \tag{13b}$$

where $\mathbf{E}_{\mathrm{c},\alpha}$ contains the discretised distribution function for the connection ($1/h$ at connection index $l_{\mathrm{c},\alpha}$, rest 0's [12]), $\mathbf{E}_\mathrm{e}$ contains the discretised distribution function for the excitation (which will be presented in Equation (25) in the next section) and $\mathbf{J}(x_\mathrm{B}^n)$ is a spreading operator containing the discretised bowing distribution ($1/h$ at time-varying bowing position x_B). If x_B is between grid points, cubic interpolation is used to spread the bow-force over neighbouring grid points [12]. All vectors are columns of size $N-1$.

It can be useful to talk about the *region of operation* of a FDS in terms of a 'stencil'. A stencil describes the number of grid points needed to calculate a single point at the next time step. The stiff string FDS has a stencil of 5 grid points. In other words, two grid points at either side of l – and l itself – are necessary to calculate u_l^{n+1}. See Figure 1 for a visualisation of this.

In order for the scheme to be stable, the grid spacing needs to abide the following condition [12]

$$h \geq \sqrt{\frac{\gamma^2k^2 + 4\sigma_1 k + \sqrt{(\gamma^2k^2+4\sigma_1 k)^2 + 16\kappa^2k^2}}{2}}. \tag{14}$$

The closer h is to this limit, the higher the quality of the implementation. The number of points N can then be calculated using

$$N = \mathrm{floor}\left(\frac{1}{h}\right). \tag{15}$$

3.2 Plate

In the case of the plate, $\mathbf{u}$ is a column vector of concatenated vertical 'strips' of the plate state as in [13] of size $(N_x-1)(N_y-1)$ and $\boldsymbol{\mathcal{F}}^n$ in Equation (11) is a combination of the discretised PDE (6) $\mathbf{f}_\beta^n$ and connection term (9b)

$$\boldsymbol{\mathcal{F}}^n = \mathbf{f}_\beta^n + \mathbf{E}_{\mathrm{c},\beta}F_\beta^n. \tag{16}$$

Here, $\mathbf{E}_{\mathrm{c},\beta}$ contains the discretised distribution function for the connection ($1/h^2$ at connection index $(l_{\mathrm{c},\beta}, m_{\mathrm{c},\beta})$, rest 0's [13]) and is a column vector of size $(N_x-1)(N_y-1)$. For the plate, the stencil will consist of 13 grid points as can be seen in Figure 2.

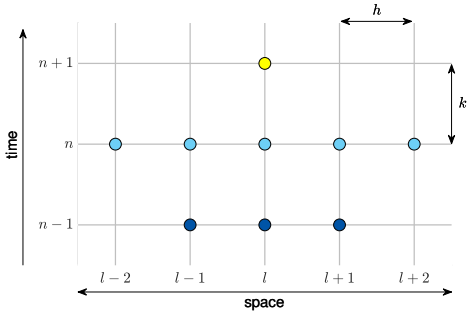

Figure 1. Stencil for a stiff string FDS with grid spacing h and time step k. The point l at the next time step (yellow) is calculated using 5 points at the current time step (blue) and 3 at the previous time step (dark blue).

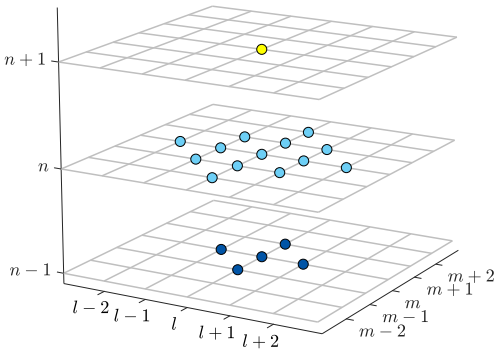

Figure 2. Stencil for a plate FDS. The point (l,m) at the next time step (yellow) is calculated using 13 points at the current time step (blue) and 5 at the previous time step (dark blue).

The grid spacing needs to abide the following condition [13]

$$h \geq 2\sqrt{k\left(\sigma_1^2 + \sqrt{\kappa^2+\sigma_1^2}\right)}, \tag{17}$$

(again, the closer h is to this limit the better) from which

N_x and N_y can be derived using

$$N_x = \text{floor}\left(\frac{\sqrt{a}}{h}\right) \quad \text{and} \quad N_y = \text{floor}\left(\frac{1}{h\sqrt{a}}\right). \quad (18)$$

3.3 Connections

In the following, we discretise the equations in (10) as shown in [13]. However, as these equations are not expressed as a function of u_{tt}, their FDS counterpart will be different. Moreover, instead of solving for $\mathbf{u}^{n+1}$, we need to solve for η^{n+1}, i.e., the relative displacement at the next time step, which will be in the form of

$$\eta^{n+1} = p^n F_\alpha^n + r^n \eta^{n-1}, \quad (19)$$

where $p^n = p(\eta^n)$ and $r^n = r(\eta^n)$ are functions of the relative displacement η if $\omega_1 \neq 0$ and constants if $\omega_1 = 0$. Again, assuming that element α is a stiff string and β is a plate, η can be calculated using

$$\eta^n = h_\alpha u^n_{\alpha,l_{\text{c},\alpha}} - h_\beta^2 u^n_{\beta,(l_{\text{c},\beta},m_{\text{c},\beta})}. \quad (20)$$

In other words, this is the difference between the state of element α at $l_{\text{c},\alpha}$ and the state of element β at $(l_{\text{c},\beta}, m_{\text{c},\beta})$ scaled by their respective (for plates, squared) grid spacings h_α and h_β. The next step is to obtain F_α^n, which can be used to easily calculate F_β^n. We first obtain values for $\mathbf{u}^{n+1}$ by solving (11) using (13a), (13b) or (16) (without the connection term!) for a string or plate respectively. As, at this point, no connection forces have been added yet, this state will be referred to as an intermediate state $\mathbf{u}^{\text{I}}$. This intermediate state can be used to obtain η^{n+1} using (20)

$$\eta^{n+1} = h_\alpha(u^{\text{I}}_{\alpha,l_{\text{c},\alpha}} + K_\alpha F_\alpha^n) - \left[h_\beta^2(u^{\text{I}}_{\beta,(l_{\text{c},\beta},m_{\text{c},\beta})} + K_\beta F_\beta^n)\right], \quad (21)$$

where K_α and K_β are as described in (12) using the damping coefficient σ_0 of their respective element. This can then be set equal to (19). Using Equation (10b), solving for F_α yields

$$F_\alpha^n = \frac{r^n \eta^{n-1} - (h_\alpha u^{\text{I}}_{\alpha,l_{\text{c},\alpha}} - h_\beta^2 u^{\text{I}}_{\beta,(l_{\text{c},\beta},m_{\text{c},\beta})})}{h_\alpha K_\alpha + \mathcal{M} h_\beta^2 K_\beta - p^n}. \quad (22)$$

4. IMPLEMENTATION

In this section, we elaborate more on the chosen values for the parameters described in the previous two sections and present the system architecture of the real-time application. The values for most parameters have been arbitrarily chosen and can – as long as they satisfy the conditions in Equations (14) and (17) – be changed. We used C++ along with the JUCE framework [20] for implementing the physical models and connections in real-time. The main hardware used was a MacBook Pro with a 2.2 GHz Intel Core i7 processor.

4.1 Stiff String

As many string properties stay constant, we chose to set the following parameters directly, rather than calculating them from their physical properties: $\kappa = 2$, $\sigma_0 = 1$, $\sigma_1 = 0.005$. An interesting parameter to make dynamic is the fundamental frequency f_0 (in s^{-1}) of the string. According to [12], the fundamental frequency can be approximately calculated using

$$f_0 \approx \frac{\gamma}{2}. \quad (23)$$

However, as the grid spacing h is dependent on the wave speed γ according to the condition found in (14), we must put a lower limit on the number of points N if we plan to dynamically increase γ.

Another way to change frequency is to add damping to the model at specific points acting as a (simplified) fretting finger. The advantage of this is that the condition (14) will never be violated. On top of this, a tapping sound will be introduced when fretting the string making it more realistic than changing the wave speed. If the string is fretted at single location $x_\text{f} \in [0, 1]$ and $l_\text{f} = \text{floor}(x_\text{f}/h)$ we use

$$u_l^n = \begin{cases} 0, & l = l_\text{f} - 1 \vee l = l_\text{f} \\ (1 - \alpha_\text{f}^\epsilon) u_l^n, & l = l_\text{f} + 1 \\ u_l^n, & \text{otherwise} \end{cases} \quad (24)$$

where $\alpha_\text{f} = x_\text{f}/h - l_\text{f}$ describes the fractional location of x_f between two grid points. Note that the grid point at the finger location and the grid point before are set to 0 to (recalling the stencil) prevent the states at either side of the finger to influence each other. The disadvantage of using this technique over regular linear interpolation, is that the effect of damping between grid points does not linearly scale to pitch. We thus added $\epsilon = 7$ as a heuristic value to more properly map finger position to pitch.

In some cases, N is fixed to a certain value (as opposed to calculating it from Equations (14) and (15)) for multiple strings of different pitches. Even though some bandwidth will be lost (in the higher frequency range), this will allow the strings to be perfectly tuned to each other.

4.1.1 Bowed String

Parameters for the bowed strings abide the following conditions: $|v_\text{B}| \leq 1$ m/s and $0 \leq F_\text{B} \leq 100$ N. It was chosen to discretise Equation (3b) implicitly making it necessary to use an iterative root-finding method such as Newton-Raphson [21].

4.1.2 Excited string

If simply excited, we set the distribution function to a raised cosine with width w_e (in grid points)

$$E_\text{e}(l) = \begin{cases} \frac{1 - \cos\left(\frac{2\pi(l - (l_\text{e} - w_\text{e}/2))}{w_\text{e}}\right)}{2}, & l_\text{e} - \frac{w_\text{e}}{2} < l < l_\text{e} + \frac{w_\text{e}}{2} \\ 0, & \text{otherwise} \end{cases} \quad (25)$$

scaled by the excitation function over time with excitation duration d_e (in samples)

$$F_\text{e}(n) = \begin{cases} \frac{1 - \cos\left(\frac{\pi(n - n_\text{e})}{d_\text{e}}\right)}{2}, & n_\text{e} \leq n < n_\text{e} + d_\text{e} \\ 0, & \text{otherwise} \end{cases} \quad (26)$$

A visualisation of this can be found in Figure 3.

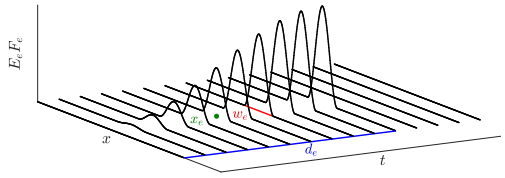

Figure 3. A visualisation of the excitation used in our implementation presented in Equation (5). The location of excitation x_e is shown in green, excitation width w_e in red and excitation duration d_e in blue (also see Equations (25) and (26)).

4.2 Plate

For the plate, the damping coefficients have been decided to be $\sigma_0 = 0.1$ and $\sigma_1 = 0.005$ and the aspect ratio is set to $a = 2$. The plate stiffness κ has been left as a user parameter to be changed dynamically and will be between the following bounds: $0.1 \leq \kappa \leq 50\ \mathrm{s}^{-1}$. In Equation (17), the grid spacing is calculated using the maximum value of κ to prevent stability issues. Using a sample rate of 44,100 Hz results in a plate with dimensions $N_x = 20$ and $N_y = 10$ (in grid points).

4.3 Connections

Increasing $\omega_1 \gtrsim 100,000\ (\mathrm{m{\cdot}s})^{-1/2}$ while keeping $0 < \omega_0 \lesssim 100\ \mathrm{s}^{-1}$ will cause audible non-linear behaviour, such as pitch-glides and rattling sounds. These effects will be more dominant when the plate stiffness is higher. In our implementation we set $\omega_0 = 100\ \mathrm{s}^{-1}$ and $\omega_1 = 100,000\ (\mathrm{m{\cdot}s})^{-1/2}$. The spring-damping $\sigma_\times = 1\ \mathrm{s}^{-1}$ is kept to a minimum ($0 \leq \sigma_\times \leq 10\ \mathrm{s}^{-1}$).

4.4 System Architecture

The system architecture can be seen in Figure 4. The top box denotes the Sensel Morph (described in more detail in the next section) controlling the application, and the white boxes show the different classes or components of the application. The black arrows indicate instructions that one class can give to another and the hollow arrows show data flows between classes. All arrows are accompanied by a coloured box indicating which thread the instruction / dataflow is associated with and at what rate this thread runs.

The lowest priority thread, the graphics-thread, is shown by green boxes and runs at 15 Hz. This draws the states of the strings, connections and the plate on the screen.

Checking and retrieving the Sensel state happens at a rate of 150 Hz and is denoted by blue boxes. The parameters that the user controls by means of the Sensel, such as bowing position, force and velocity, will be updated in the models at this rate as well.

The highest priority thread is the audio-thread and runs at commonly-used sample rate 44,100 Hz. The main application gives an 'update' (u) instruction to the instrument,

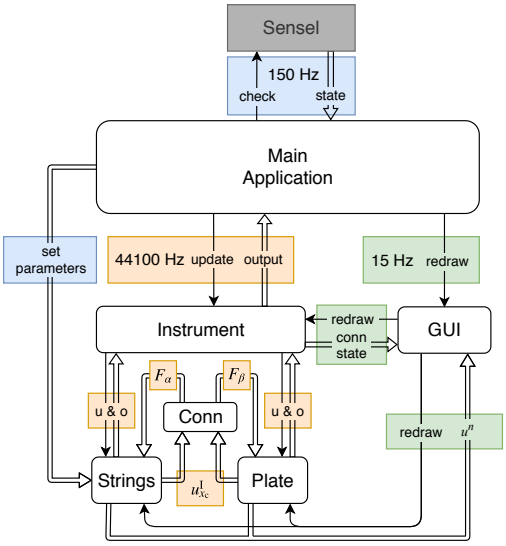

Figure 4. System architecture flowchart. See Section 4.4 for a thorough explanation.

which in turn updates the FDSs in its strings and plate. After the FDS update is done, the intermediate state at the connection points $u^{\mathrm{I}}_{x_c}$ (where $x_c = l_{c,\alpha}$ for the string or $x_c = (l_{c,\beta}, m_{c,\beta})$ for the plate) are sent to the connection (Conn) class which calculates the force-functions F_α and F_β. These values are then sent back to the string and plate classes and added to their respective states after which their outputs (o) (at arbitrary points) are sent back to the main application. See Algorithm 1 for this 'order of calculation'.

5. INSTRUMENTS AND USER INTERACTION

In this section, the Sensel Morph (or simply Sensel) and user interface will be described in more detail. Furthermore, several configurations of strings, plates and connections that are inspired by real-life instruments will be presented. A demonstration of one of the instruments can be found in [22].

5.1 Sensel Morph

The Sensel Morph is a highly accurate touch controller that senses position and force of objects [17] (see Figure 5). We use the Sensel as an expressive interface for interacting with the instrument configurations. Right above the touch-sensitive area, the Sensel contains an array of 24 LEDs that can be controlled from the application.

5.2 User interface

Strings are shown as coloured paths (see Figure 6 for a descriptive visualisation). The state $\mathbf{u}^n$ of the string is visualised using the vertical displacement of the paths. Bowed strings are shown in cyan on the top left. The bow is shown

while *application runs* **do**
  for *all elements* **do**
    calculate intermediate state $\mathbf{u}^{\mathrm{I}}$ using previous state values (as in Equation (11))
      $\mathbf{u}^{\mathrm{I}}_{\mathrm{s}} = 2\mathbf{u}^{n} - \mathbf{u}^{n-1} + K\boldsymbol{\mathcal{F}}$
  end
  if *element is excited/bowed* **then**
    calculate excitation term $\mathbf{E}$ and add to intermediate state of the element
      $\mathbf{u}^{\mathrm{I}}_{\mathrm{s+e}} = \mathbf{u}^{\mathrm{I}}_{\mathrm{s}} + \mathbf{E}$
  end
  for *all connections* **do**
    calculate connection forces and add connection term $\mathbf{C}$ to elements to obtain the state at the next time step
      $\mathbf{u}^{n+1}_{\mathrm{s+e+c}} = \mathbf{u}^{\mathrm{I}}_{\mathrm{s+e}} + \mathbf{C}$
  end
  update state vectors
    $\mathbf{u}^{n-1} = \mathbf{u}^{n}$
    $\mathbf{u}^{n} = \mathbf{u}^{n+1}_{\mathrm{s+e+c}}$
  increment time step
    n++
end

Algorithm 1: Pseudocode showing the correct order of calculation. The subscripts for state vector $\mathbf{u}$ shows what it consists of ('s' for previous state, 'e' for excitation and 'c' for connection).

as a yellow rectangle and moves on interaction. The fretting position is shown as a yellow circle. Plucked strings are shown in purple in the top right, underneath which the sympathetic strings are shown in light green. The plate is shown in the bottom using a grid of rectangles (clamped grid points are not shown). Its state is visualised using a grey-scale. Furthermore, connections are shown using orange circles/squares for the points of connection and dotted lines between these points. Lastly, all parameters that are controlled by the mouse such as output-level and plate-stiffness are located in a column on the right side of the application.

5.3 Instruments

We subdivide string-elements into three types: bowed, plucked and sympathetic strings. All strings will be connected to one plate acting as an instrument body of which the user can control the plate-stiffness. Furthermore, the user can change the output-level of each element type. Apart from these parameters, which are controlled by the mouse, the instruments are fully controlled by two Sensels. The instruments we have chosen as our inspiration are the sitar, the hammered dulcimer and the hurdy gurdy.

5.3.1 Bowed Sitar

The sitar is originally an Indian string instrument that has both fretted strings and sympathetic strings. Instead of plucking the fretted strings, we extended the model to bow them. Our implementation consists of 2 bowed strings

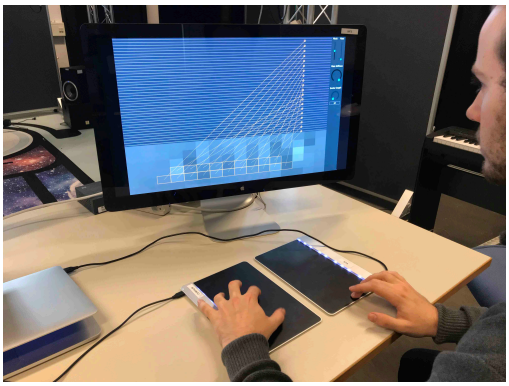

Figure 5. Player using the Sensel Morphs to interact with one of the instruments.

(tuned to A3 and E4), 13 sympathetic strings (tuned according to [23]) and 5 plucked strings (tuned A3-E4 following an A-major scale) as it is also possible to strum the sympathetic strings. See Figure 6 for a visual of the implementation. One Sensel is vertically subdivided into two sections; one for each bowed string. The first finger registered by the Sensel is mapped to a bow and the second is mapped to a fretting finger controlling pitch. The horizontal position of both fingers is visualised using the Sensel's LED array. The frets are not implemented as such (the pitch is continuous), but they are visualised for reference. The horizontal position of the first finger is mapped to the bowing position on the string, the vertical velocity to the bow velocity v_{B} and the finger force is linked to the excitation function F_{B} (both in Equation (3a)). The other Sensel is subdivided into 5 sections mapped to the plucked strings. These sections are visualised by the LED array for reference.

The mass ratio for the bowed/plucked string to plate connections has been set to $\mathcal{M} = 2$ and ratio for the sympathetic string to plate connections has been set to $\mathcal{M} = 0.5$ to increase the effect that the playable strings have on the sympathetic strings.

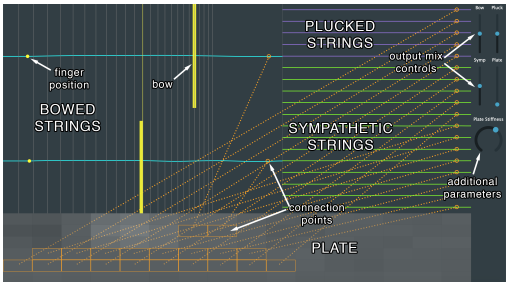

Figure 6. The bowed sitar application. The descriptions of the different elements and other objects are shown in the image, but will (naturally) not be visible in the application.

5.3.2 *Hammered Dulcimer*

The hammered dulcimer is an instrument that can be seen as an 'open piano' where the musician has the hammers in their hand. Just like the piano, the strings are grouped in pairs or triplets that are played simultaneously. In our implementation, we have 20 pairs of plucked strings. Even though most hammered dulcimers have more strings, we decided that this configuration has the highest number of strings while maintaining playability. One of each pair is connected to the plate which slightly detunes it, creating a desired 'chorusing' effect. See Figure 7 for a visual of the implementation. In order for the excitation to more resemble a strike of a hammer than a pluck, the contents of the cosine in (26) will be multiplied by 2 for the excitation to have a less abrupt ending, something desired for a hammered interaction. Moreover, the excitation-length can be changed to simulate short and long hammer-times.

The Sensels are placed vertically next to each other (see Figure 5). The pair with the lowest frequency will then be located in the bottom right and the highest in the top left, as in the real instrument. As with the plucked strings of the bowed sitar, the LED array is used to visualise the way that the Sensel is subdivided, which is especially useful here as one Sensel controls 10 string-pairs.

The mass ratio is set relatively high ($\mathcal{M} = 100$) to amplify the non-linear interaction between the strings and the detuning of the strings connected to the plate.

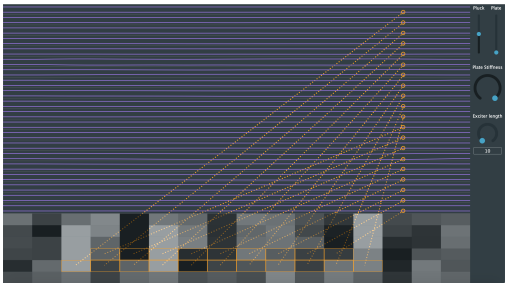

Figure 7. The hammered dulcimer application.

5.3.3 *Hurdy Gurdy*

The hurdy gurdy is an instrument that consists of bowed and sympathetic strings. The bowing happens through a rosined wheel attached to a crank and bows these strings as the crank is turned. It is possible to change the pitch of a few bowed strings - the melody strings - using buttons that press tangent pins on the strings at different positions. The other strings, referred to as drone strings, are mostly tuned lower than the melody strings and provide the bass frequencies of the instrument. The musician can place the bowed strings on rests that keep the wheel from interacting with it.

Our implementation consists of 5 bowed strings subdivided into 2 drone strings tuned to A2, E3 and 3 melody strings tuned to A3, E4 and A4 and 13 sympathetic strings tuned the same way as the sympathetic strings in bowed sitar. Furthermore, the mass ratios have been set the same as in the bowed sitar application. See Figure 8 for a visual of the implementation.

The Sensel is vertically subdivided into 5 rows that control whether the strings are placed on the wheel. The bowing velocity is mapped to the average pressure of the fingers. The fundamental frequency (in the model $\gamma/2$) of the melody-strings is changed by a Sensel with a piano-overlay acting as a midi controller. A demonstration of this instrument can be found in [22]. It is interesting to note here that the sympathetic strings that are in tune with the harmonics of the bowed strings resonate most, which is expected to happen in the real world as well.

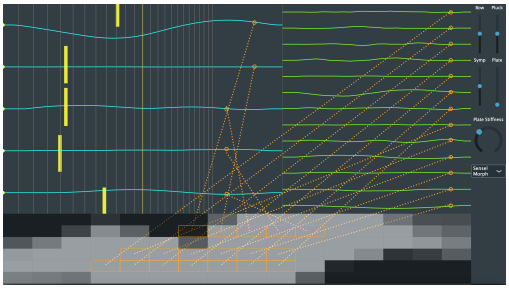

Figure 8. The hurdy gurdy application.

6. RESULTS AND DISCUSSION

Table 1 shows the CPU usage (on the same MacBook Pro 2.2 GHz i7 as described before) for the three instruments presented in the previous section. As the Sensel thread contributes a negligible amount to the CPU usage, this is not shown in the table.

Application	No Sound	No Graphics	Total
Bowed Sitar	32	63	85
Dulcimer	30	66	85
Hurdy Gurdy	28	58	78

Table 1. CPU usage (in %) for the instruments found in Section 5. Values show usage of one (virtual) thread and have been taken as an average (with a margin of ~5%) over a short period of time. The two middle columns show usage when the sound or graphics thread has been turned off.

As can be seen from the table, all instruments use about the same amount of CPU and none of them have audible dropouts (CPU $< 100\%$). It can be observed that the graphics use about 20% of the CPU, indicating that there is still much room to increase the complexity of the instrument-configurations before dropouts will occur. On the other hand, should the instruments be used in parallel with other audio applications or plug-ins, the CPU usage has to be greatly reduced. The first step towards this would be to vectorise the FDSs using AVX instructions.

While our instruments have been not formally evaluated yet, we have performed some qualitative evaluations with

sound and music computing experts. The goals of the evaluations were to explore the playability of the instrument, sonic quality and intuitiveness of control. These evaluations showed that especially the bowing interaction feels intuitive and creates a natural sound. The overall sound of the instruments was generally judged to be interesting, but not "sounding like a real-life instrument". This makes sense, as we did not seek to perfectly model each instrument, but rather used them as an inspiration for the configurations of the physical models. The next step for sound quality would be to replace the thin plate with a more realistic element, such as a wooden instrument body.

7. CONCLUSION

In this paper, a real-time modular physical modelling synthesis environment structured as a network of connected strings and plates has been presented. Several instruments have been created in the context of this environment which can be played by a pair of Sensel Morphs allowing for highly expressive control of these instruments. Informal evaluations with professional musicians have confirmed that the interaction is found natural and the output sound interesting. Further steps to improve this project are to optimise the algorithm and to replace the plate with a more realistic instrument body.

Acknowledgments

This work is supported by NordForsk's Nordic University Hub Nordic Sound and Music Computing Network NordicSMC, project number 86892.

8. REFERENCES

[1] C. Cadoz, "Synthèse sonore par simulation de mécanismes vibratoires," 1979, thèse de Docteur Ingénieur, I.N.P.G. Grenoble, France.

[2] C. Cadoz, A. Luciani, 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.

[3] 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.

[4] J. D. Morrison and J.-M. Adrien, "Mosaic: A framework for modal synthesis," *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, 1993.

[5] J. O. Smith, "Physical modeling using digital waveguides," *Computer music journal*, vol. 16, no. 4, pp. 74–91, 1992.

[6] P. Ruiz, "A technique for simulating the vibrations of strings with a digital computer," Master's thesis, University of Illinois, 1969.

[7] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I," *Journal of the Audio Engineering Society*, vol. 19, no. 6, pp. 462–470, 1971.

[8] ——, "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.

[9] R. Bacon and J. Bowsher, "A discrete model of a struck string," *Acustica*, vol. 41, pp. 21–27, 1978.

[10] 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.

[11] 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.

[12] S. Bilbao, *Numerical Sound Synthesis, Finite Difference Schemes and Simulation in Musical Acoustics*. John Wiley and Sons, Ltd, 2009.

[13] ——, "A modular percussion synthesis environment," *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, 2009.

[14] S. Bilbao, A. Torin, P. Graham, J. Perry, and G. Delap, "Modular physical modeling synthesis on gpu," in *Proceedings of the 40th International Computer Music Conference and the 11th Sound and Music Computing Conference*, 2013.

[15] C. Webb and S. Bilbao, "On the limits of real-time physical modelling synthesis with a modular environment," *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx-15)*, 2015.

[16] K. Franinović and S. Serafin, *Sonic interaction design*. Mit Press, 2013.

[17] Sensel Inc. (2018) Sensel morph. [Online]. Available: https://sensel.com/

[18] C. Desvages and S. Bilbao, "Two-polarisation physical model of bowed strings with nonlinear contact and friction forces, and application to gesture-based sound synthesis," *Applied Sciences*, 2016.

[19] K. Graff, *Wave Motion in Elastic Solids*. New York, New York: Dover, 1975.

[20] JUCE ROLI. (2019) JUCE. [Online]. Available: https://juce.com/

[21] J. Wallis, *A treatise of algebra, both historical and practical*. London, 1685.

[22] S. Willemsen. (2019) Hurdy gurdy demo. [Online]. Available: https://www.youtube.com/watch?v=BkxLji2ap1w

[23] Joe Rizzo. (2015) How to tune a sitar. [Online]. Available: http://www.joerizzo.com/sitar/

Paper C

Physical Models and Real-Time Control with the Sensel Morph

PHYSICAL MODELS AND REAL-TIME CONTROL WITH THE SENSEL MORPH

Silvin Willemsen
Multisensory Experience Lab, CREATE,
Aalborg University Copenhagen
Copenhagen, Denmark
sil@create.aau.dk

Stefan Bilbao
Acoustics and Audio Group
University of Edinburgh
Edinburgh, UK
s.bilbao@ed.ac.uk

Nikolaj Andersson, Stefania Serafin
Multisensory Experience Lab, CREATE,
Aalborg University Copenhagen
Copenhagen, Denmark
{nsa, sts}@create.aau.dk

ABSTRACT

In this demonstration we present novel physical models controlled by the Sensel Morph interface.

1. INTRODUCTION

The Sensel Morph is a highly accurate touch controller that senses position and force of objects [1]. Figure 1 shows one player interacting with two Sensel Morph devices to interact with the developed physical models. We use the Sensel as an expressive interface for interacting with different physical models described in a companion paper accepted to SMC 2019. Right above the touch-sensitive area, the Sensel contains an array of 24 LEDs that can be controlled from the application.

Strings are shown as coloured paths (see Figure 2 for a descriptive visualisation). The state of the string is visualised using the vertical displacement of the paths. Bowed strings are shown in cyan on the top left. The bow is shown as a yellow rectangle and moves while interacting. The fretting position is shown as a yellow circle. Plucked strings are shown in purple in the top right, underneath which the sympathetic strings are shown in light green. The plate is shown in the bottom using a grid of rectangles (clamped grid points are not shown). Its state is visualised using a grey-scale. Furthermore, connections are shown using orange circles/squares for the points of connection and dotted lines between these points. Lastly, all parameters that are controlled by the mouse such as output-level and plate-stiffness are located in a column on the right side of the application.

2. IMPLEMENTED INSTRUMENTS

We subdivide string-elements into three types: bowed, plucked and sympathetic strings. All strings will be connected to one plate acting as an instrument body of which the user can control the plate-stiffness. Furthermore, the user can change the output-level of each element type. Apart from these parameters, which are controlled by the mouse, the instruments are fully controlled by two Sensels. The instruments we have chosen as our inspiration are the sitar, the hammered dulcimer and the hurdy gurdy.

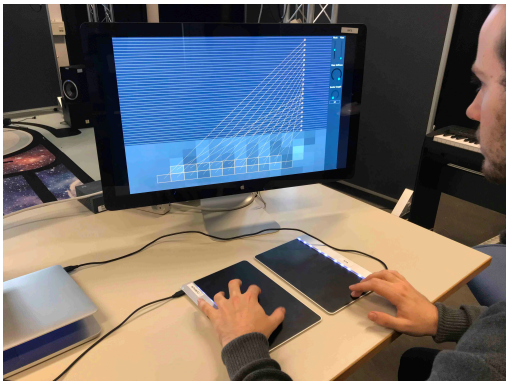

Figure 1. Player using the Sensel Morphs to interact with one of the instruments.

2.1 Bowed Sitar

The sitar is originally an Indian string instrument that has both fretted strings and sympathetic strings. Instead of plucking the fretted strings, we extended the model to bow them. Our implementation consists of 2 bowed strings (tuned to A3 and E4), 13 sympathetic strings (tuned according to [2]) and 5 plucked strings (tuned A3-E4 following an A-major scale) as it is also possible to strum the sympathetic strings. Figure 2 shows the visual interface of the implementation. One Sensel is vertically subdivided into two sections; one for each bowed string. The first finger registered by the Sensel is mapped to a bow and the second is mapped to a fretting finger controlling pitch. The horizontal position of both fingers is visualised using the Sensel's LED array. The frets are not implemented as such (the pitch is continuous), but they are visualised for reference.

2.2 Hammered Dulcimer

The hammered dulcimer is an instrument that can be seen as an 'open piano' where the musician has the hammers in their hand. Just like the piano, the strings are grouped in

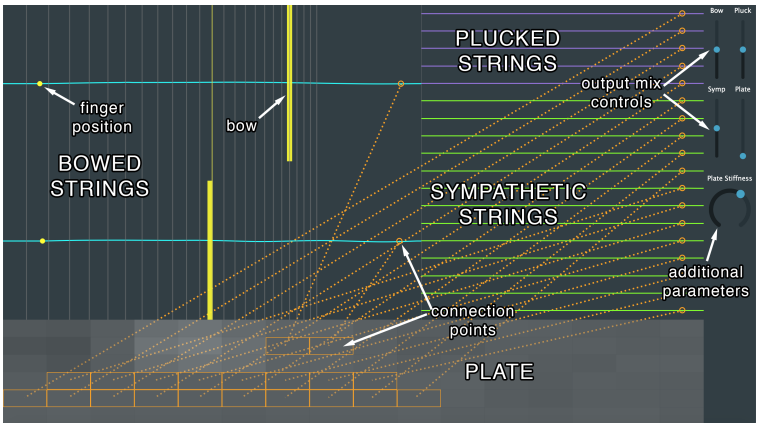

Figure 2. The bowed sitar application. The descriptions of the different elements and other objects are shown in the image, but will (naturally) not be visible in the application.

pairs or triplets that are played simultaneously. The interface for the hammered dulcimer can be seen in Figure 3. In our implementation, we have 20 pairs of plucked strings. Even though most hammered dulcimers have more strings, we decided that this configuration has the highest number of strings while maintaining playability. One of each pair is connected to a plate which slightly detunes it, creating a desired 'chorusing' effect. Two Sensel boards are placed vertically next to each other (see Figure 1). The pair with the lowest frequency is located in the bottom right and the highest in the top left, as in the real instrument. As with the plucked strings of the bowed sitar, the LED array is used to visualise the way that the Sensel is subdivided, which is especially useful here as one Sensel controls 10 string-pairs.

The mass ratio is set relatively high ($\mathcal{M} = 100$) to amplify the non-linear interaction between the strings and the detuning of the strings connected to the plate.

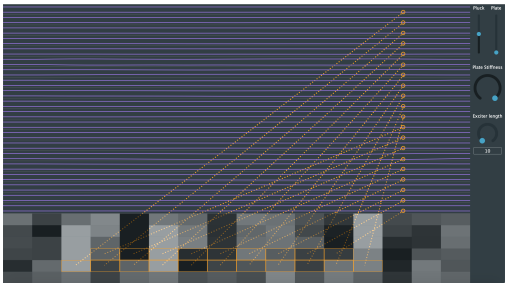

Figure 3. The hammered dulcimer application.

2.3 Hurdy Gurdy

The hurdy gurdy is an instrument that consists of bowed and sympathetic strings. The bowing happens through a rosined wheel attached to a crank and bows these strings as the crank is turned. It is possible to change the pitch of a few bowed strings - the melody strings - using buttons that press tangent pins on the strings at different positions. The other strings, referred to as drone strings, are mostly tuned lower than the melody strings and provide the bass frequencies of the instrument. The musician can place the bowed strings on rests that keep the wheel from interacting with it. The visual interface can be seen in Figure 4. Our implementation consists of 5 bowed strings subdivided into 2 drone strings tuned to A2, E3 and 3 melody strings tuned to A3, E4 and A4 and 13 sympathetic strings tuned the same way as the sympathetic strings in bowed sitar. Furthermore, the mass ratios have been set the same as in the bowed sitar application.

The Sensel is vertically subdivided into 5 rows that control whether the strings are placed on the wheel.

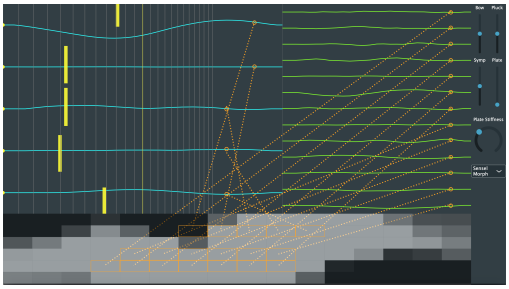

Figure 4. The hurdy gurdy application.

3. REFERENCES

[1] Sensel Inc. (2018) Sensel morph. [Online]. Available: https://sensel.com/

[2] Joe Rizzo. (2015) How to tune a sitar. [Online]. Available: http://www.joerizzo.com/sitar/

Paper D

Real-Time Implementation of an Elasto-Plastic Friction Model applied to Stiff Strings using Finite Difference Schemes

REAL-TIME IMPLEMENTATION OF AN ELASTO-PLASTIC FRICTION MODEL APPLIED TO STIFF STRINGS USING FINITE-DIFFERENCE SCHEMES

Silvin Willemsen
Multisensory Experience Lab, CREATE,
Aalborg University Copenhagen
Copenhagen, Denmark
sil@create.aau.dk

Stefan Bilbao
Acoustics and Audio Group
University of Edinburgh
Edinburgh, UK
s.bilbao@ed.ac.uk

Stefania Serafin
Multisensory Experience Lab, CREATE,
Aalborg University Copenhagen
Copenhagen, Denmark
sts@create.aau.dk

ABSTRACT

The simulation of a bowed string is challenging due to the strongly non-linear relationship between the bow and the string. This relationship can be described through a model of friction. Several friction models in the literature have been proposed, from simple velocity dependent to more accurate ones. Similarly, a highly accurate technique to simulate a stiff string is the use of finite-difference time-domain (FDTD) methods. As these models are generally computationally heavy, implementation in real-time is challenging. This paper presents a real-time implementation of the combination of a complex friction model, namely the elasto-plastic friction model, and a stiff string simulated using FDTD methods. We show that it is possible to keep the CPU usage of a single bowed string below 6 percent. For real-time control of the bowed string, the Sensel Morph is used.

1. INTRODUCTION

In physical modelling sound synthesis applications, the simulation of a bowed string is a challenging endeavour. This is mainly due to the strongly non-linear relationship between the bow and the string, through a model of friction. Such friction models can be categorised as static or dynamic; models of the latter type have only recently seen a major effort. As opposed to static friction models, where friction depends only on the relative velocity of the two bodies in contact, dynamic models describe the friction force through a differential equation.

A recently popular dynamic model is the elasto-plastic model, first proposed in [1]. The model assumes that the friction between the two objects in contact is caused by a large ensemble of bristles, each of which contributes to the total friction force. The average bristle deflection is used as an extra independent variable for calculating the friction force. As shown in [2], the elasto-plastic model can be applied to a bowed string simulation and it exhibits a hysteresis loop in the force versus velocity plane due to this multivariable dependency. This is consistent with measurements performed using a bowing machine in [3]. The elasto-plastic model has been thoroughly investigated in a musical context by Serafin et al. in [2, 4, 5].

Regarding bowed string simulations, the first musical non-linear systems, including bowed strings, were presented by McIntyre, et al. in [6]. Smith published the first real-time implementation of the bowed string using a digital waveguide (DW) for the string and a look-up table for the friction model in [7]. Simultaneously, Florens, et al. published a real-time implementation using mass-spring systems for the string and a static friction model for the bow in [8].

The dynamics of musical instruments are generally described by systems of partial differential equations (PDEs). Specialised synthesis methods such as DWs [9] and modal synthesis [10] are derived from particular solutions. Mainstream time-stepping methods such as finite-difference time-domain (FDTD) methods were first proposed in [11, 12, 13], and developed subsequently [14, 15]. In [16] the authors adapted the thermal model proposed by Woodhouse in [3] for real-time applications using a DW for the string implementation and a combination of the DW and FDTD methods for the bowing interaction. In [17, 18], Desvages used FDTD methods for the implementation of the string in two polarizations and a static double exponential friction model introduced in [19]. This was, however, not implemented in real-time. To the best of the authors' knowledge, the only known real-time implementation of any bow model applied to complete FDTD strings was presented in [20] where the soft exponential friction function presented in [14] was used. The current work can be considered an extension of this work.

We are interested in bridging the gap between highly accurate physical models and efficient implementations so that these models can be played in real-time. In this work, we present an implementation of the elasto-plastic friction model in conjunction with a finite-difference implementation of the damped stiff string. Furthermore, we show that it is possible to play the string in real-time using the Sensel Morph controller [21].

This paper is structured as follows. In Section 2, the elasto-plastic bow model in conjunction with a PDE model for a stiff string is described. Discretisation is covered in Section 3, and implementation details appear in Section 4. In Section 5, simulated results are presented and discussed. Some concluding remarks appear in Section 6.

2. ELASTO-PLASTIC BOW MODEL

Consider a linear model of transverse string vibration in a single polarization, where $u(x,t)$ represents string displacement as a function of time $t \geq 0$, in s, and coordinate $x \in [0,\ L]$ (in m) for some string length L (in m). Using the subscripts t and x to denote differentiation with respect to time and space respectively, a partial differential equation describing the dynamics of the damped stiff string is [14]

$$u_{tt} = c^2 u_{xx} - \kappa^2 u_{xxxx} - 2\sigma_0 u_t + 2\sigma_1 u_{txx}. \qquad (1)$$

Here, $c = \sqrt{T/\rho A}$ is the wave speed (in m/s) with tension T (in N), material density ρ (in kg·m^{-3}) and cross-sectional area A (in m^2). Furthermore, $\kappa = \sqrt{EI/\rho A}$ is the stiffness coefficient (in m^2/s) with Young's Modulus E (in Pa) and area moment of inertia I (in m^4). For a string of circular cross section we have radius r (in m), cross-sectional area $A = \pi r^2$ and area moment of inertia $I = \pi r^4/4$. Lastly, $\sigma_0 \geq 0$ (in s^{-1}) and $\sigma_1 \geq 0$ (in m^2/s) are coefficients allowing for frequency-independent and frequency-dependent damping respectively.

In our implementation we assume simply supported boundary conditions, which are defined as

$$u = u_{xx} = 0 \quad \text{where} \quad x = 0, L\,. \tag{2}$$

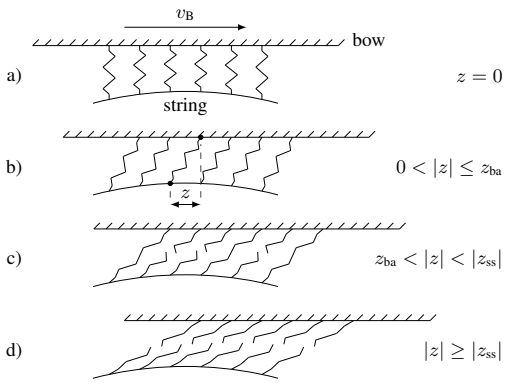

Figure 1: *Microscopic displacements of the bristles between the bow and the string. The bow moves right with a velocity of v_B. a) The initial state is where the average bristle displacement $z = 0$. b) The bow has moved right relative to the string. The purely elastic, or presliding regime is entered (stick). c) After the break-away displacement z_{ba}, more and more bristles start to 'break'. This is defined as the elasto-plastic regime. d) After all bristles have 'broken', the steady state (slip) is reached and the purely plastic regime is entered.*

As mentioned in the introduction, the elasto-plastic bow model assumes that the friction between the bow and the string is due to a large ensemble of bristles, each of which contributes to the total friction force. See Figure 1 for a graphical representation of this. The bristles are assumed to be damped stiff springs and can 'break' after a given break-away displacement threshold. An extra term can be added to (1) to include the bowing interaction

$$u_{tt} = \ldots - \delta(x - x_B) f(v, z)/\rho A. \tag{3}$$

Here, the spatial Dirac delta function $\delta(x - x_B)$ (in m^{-1}) allows for the pointwise application of the force f (in N) at externally supplied bowing position $x_B(t)$ (in m).

In the following we will use the definitions found in [1]. The force f is defined in terms of the relative velocity v (in m/s) and average bristle displacement z (in m) (see Figure 1) as

$$f(v, z) = s_0 z + s_1 \dot{z} + s_2 v + s_3 w, \tag{4}$$

where

$$v = u_t(x_B) - v_B, \tag{5}$$

where $v_B(t)$ is an externally supplied bow velocity, s_0 is the bristle stiffness (in N/m), s_1 is the damping coefficient (in kg/s), s_2 is the viscous friction (in kg/s) and s_3 is a dimensionless noise coefficient multiplied onto pseudorandom function $w(t)$ (in N) as done in [4] and adds noise to the friction force. Here, $\dot{z}$ indicates a time derivative of z, and is related to v through

$$\dot{z} = r(v, z) = v\left[1 - \alpha(v, z)\frac{z}{z_{ss}(v)}\right], \tag{6}$$

where z_{ss} is the steady-state function

$$z_{ss}(v) = \frac{\text{sgn}(v)}{s_0}\left[f_C + (f_S - f_C)e^{-(v/v_S)^2}\right], \tag{7}$$

with Stribeck velocity v_S (in m/s), Coulomb force $f_C = f_N\mu_C$ and stiction force $f_S = f_N\mu_S$ (both in N). Here μ_C and μ_S are the dynamic and static friction coefficient respectively and $f_N(t)$ is the normal force (in N) which is, like $v_B(t)$, externally supplied. See Figure 2 for a plot of (7).

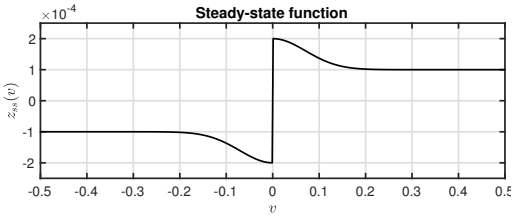

Figure 2: *A plot of the steady-state function $z_{ss}(v)$ with a force of 5 N.*

Furthermore, the adhesion map between the bow and the string is defined as

$$\alpha(v, z) = \begin{cases} 0 & |z| \leq z_{ba} \\ \alpha_m(v, z) & z_{ba} < |z| < |z_{ss}(v)| \\ 1 & |z| \geq |z_{ss}(v)| \end{cases} \text{if } \text{sgn}(v) = \text{sgn}(z)$$
$$\text{if } \text{sgn}(v) \neq \text{sgn}(z), \tag{8}$$

where the transition between the elastic and plastic behaviour is defined as

$$\alpha_m = \frac{1}{2}\left[1 + \text{sgn}(z)\sin\left(\pi\frac{z - \text{sgn}(z)\frac{1}{2}(|z_{ss}(v)| + z_{ba})}{|z_{ss}(v)| - z_{ba}}\right)\right], \tag{9}$$

with break-away displacement z_{ba}, i.e., where the bristles start to break (see Figure 1 c)). A plot of the adhesion map can be found in Figure 3.[1]

One of the difficulties in working with this model is that, due to the many approximations, the notion of an energy balance, relat-

[1]It is interesting to note is that in the literature on this topic such as [1, 2, 4, 5], a few inaccuracies can be found in the definition of $\alpha(v, z)$: 1) all uses of z_{ss} in (8) and (9) lack the absolute value operator, 2) the multiplications with $\text{sgn}(z)$ in (9) are excluded, 3) $\alpha(v, z)$ is undefined for $|z| = z_{ba}$ and $|z| = |z_{ss}(v)|$ (correct in the original paper by Dupont et al. [1]). It can be shown that only with the definitions presented here, is it possible to obtain the curve shown in Figure 3.

ing the rate of stored energy in the system to power input and loss is not readily available. Such energy methods are used frequently in the context of physical modeling synthesis and virtual analog modeling as a means of arriving at numerical stability conditions for strongly nonlinear systems, as is the present case. See, e.g., [14]. This means that we do not have a means of ensuring numerical stability in the algorithm development that follows. This does not mean, however, that an energy balance is not available.

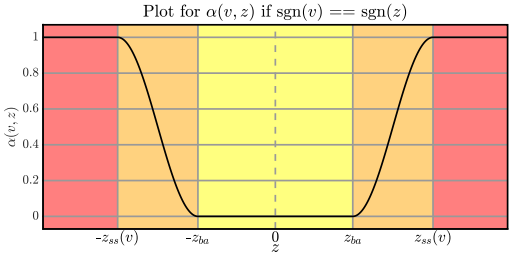

Figure 3: *A plot of the adhesion map $\alpha(v,z)$ plotted against z when the signs of v and z are the same. The different regions of the map are shown with the coloured areas and correspond to Figure 1 according to: yellow - a) & b), orange - c) and red - d).*

3. DISCRETISATION

Finite-difference schemes for the stiff string in isolation are covered by various authors [13, 14].

Equation (1) can be discretised at times $t = nk$, with sample $n \in \mathbb{N}$ and time-step $k = 1/f_\text{s}$ (in s) with sample-rate f_s (in Hz) and locations $x = lh$, where grid spacing h (in m) needs to abide the following condition [14]

$$h \geq h_\text{min} = \sqrt{\frac{c^2k^2 + 4\sigma_1 k + \sqrt{(c^2k^2 + 4\sigma_1 k)^2 + 16\kappa^2 k^2}}{2}} \tag{10}$$

and grid points $l \in [0, ..., N]$, where $N = \text{floor}(L/h)$ and $N + 1$ is the total number of grid points. It is important to note that the closer h is to h_min, the more accurate the scheme will be. Approximations for the derivatives found in (1) are described in the following way [14]:

$$u_t \approx \delta_{t\cdot} u_l^n = \frac{1}{2k}\big(u_l^{n+1} - u_l^{n-1}\big), \tag{11a}$$

$$u_{tt} \approx \delta_{tt} u_l^n = \frac{1}{k^2}\big(u_l^{n+1} - 2u_l^n + u_l^{n-1}\big), \tag{11b}$$

$$u_{xx} \approx \delta_{xx} u_l^n = \frac{1}{h^2}\big(u_{l+1}^n - 2u_l^n + u_{l-1}^n\big), \tag{11c}$$

$$u_{txx} \approx \delta_{t-}\delta_{xx} u_l^n = \frac{1}{hk^2}\big(u_{l+1}^n - 2u_l^n + u_{l-1}^n - u_{l+1}^{n-1} + 2u_l^{n-1} - u_{l-1}^{n-1}\big), \tag{11d}$$

$$u_{xxxx} \approx \delta_{xxxx} u_l^n = \frac{1}{h^4}\big(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n\big), \tag{11e}$$

with grid function u_l^n denoting a discretised version of $u(x,t)$ at the nth time step and the lth point on the string. Note that in (11d), the backwards time difference operator is used to keep (12) explicit and thus computationally cheaper to update. Using the approximations shown in (11), (3) can be discretised to

$$\begin{aligned}\delta_{tt} u_l^n = c^2\delta_{xx} u_l^n - \kappa^2\delta_{xxxx} u_l^n - 2\sigma_0\delta_{t\cdot} u_l^n \\ + 2\sigma_1\delta_{t-}\delta_{xx} u_l^n - J(x_\text{B}^n) f(v^n, z^n)/\rho A,\end{aligned} \tag{12}$$

where the relative velocity described in (5) can be discretised as

$$v^n = I(x_\text{B}^n)\delta_{t\cdot} u_l^n - v_\text{B}^n. \tag{13}$$

Here, $I(x_\text{B}^n)$ and $J(x_\text{B}^n)$ are weighting functions where the former interpolates the string displacement and velocity and the latter distributes the bowing term around time-varying bowing position x_B^n (see Figure 4 and [14] for more details on this). Furthermore,

$$f(v^n, z^n) = s_0 z^n + s_1 r^n + s_2 v^n + s_3 w^n \tag{14}$$

is the discrete counterpart of (4) where

$$r^n = r(v^n, z^n) = v^n\left[1 - \alpha(v^n, z^n)\frac{z^n}{z_\text{ss}(v^n)}\right] \tag{15}$$

is the discrete counterpart of (6).

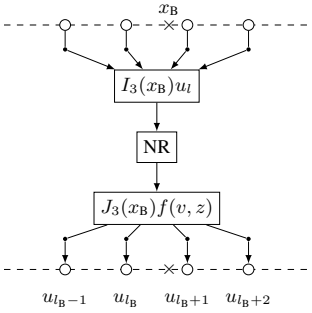

Figure 4: *Cubic interpolation at bowing point x_B. The interpolator I retrieves the values of four grid points which are then used in the Newton-Raphson (NR) solver. This outputs the force function $f(v,z)$ that the spreading function J in turn distributes over the same four grid points. This process happens every single sample.*

At the bowing point we need to iteratively solve for two unknown variables: the relative velocity between the bow and the string v^n and the mean bristle displacement z^n of the bow at sample n. We can solve (12) at x_B^n using (13) and identity [14]

$$\delta_{tt} u_l^n = \frac{2}{k}\big(\delta_{t\cdot} u_l^n - \delta_{t-} u_l^n\big) \tag{16}$$

resulting in

$$I(x_\text{B}^n)J(x_\text{B}^n) f(v^n, z^n)/\rho A + \Big(\frac{2}{k} + 2\sigma_0\Big)v^n + b^n = 0, \tag{17}$$

where

$$b^n = \frac{2}{k} v_{\mathrm{B}}^n - \frac{2}{k} I(x_{\mathrm{B}}^n)\delta_{t-} u_l^n - c^2 I(x_{\mathrm{B}}^n)\delta_{xx} u_l^n + \kappa^2 I(x_{\mathrm{B}}^n)\delta_{xxxx} u_l^n + 2\sigma_0 v_{\mathrm{B}}^n - 2\sigma_1 I(x_{\mathrm{B}}^n)\delta_{t-}\delta_{xx} u_l^n \tag{18}$$

and can be pre-computed as its terms are not dependent on v^n or z^n. Recalling (4), this can be rewritten to

$$I(x_{\mathrm{B}}^n) J(x_{\mathrm{B}}^n) \left(\frac{s_0 z^n + s_1 r^n + s_2 v^n + s_3 w^n}{\rho A} \right) + \left(\frac{2}{k} + 2\sigma_0 \right) v^n + b^n = 0. \tag{19}$$

To obtain the values of v^n and z^n, multivariate Newton-Raphson (NR) is used. If (19) is defined to be $g_1 = g_1(v^n, z^n)$ and

$$g_2(v^n, z^n) = r^n - a^n = 0, \tag{20}$$

with

$$a^n = (\mu_{t-})^{-1} \delta_{t-} z^n \tag{21}$$

(where the operators applied to z^n denote the trapezoid rule [14]) we obtain the following iteration

$$\begin{bmatrix} v_{(i+1)}^n \\ z_{(i+1)}^n \end{bmatrix} = \begin{bmatrix} v_{(i)}^n \\ z_{(i)}^n \end{bmatrix} - \begin{bmatrix} \frac{\partial g_1}{\partial v} & \frac{\partial g_1}{\partial z} \\ \frac{\partial g_2}{\partial v} & \frac{\partial g_2}{\partial z} \end{bmatrix}^{-1} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}, \tag{22}$$

where i is the iteration number capped by 50 iterations, and the convergence threshold is set to 10^{-7}.

4. IMPLEMENTATION

In this section, we will elaborate on the implementation; the parameters used and the system architecture. The real-time implementation of the discrete-time model shown in the previous section has been done using C++ together with the JUCE framework [22]. The application is shown in Figure 5. The parameters we used can be found in Table 1, most of which are based on implementations by Serafin in [4]. These parameters will be static, i.e., are not user-controlled (except for z_{ba} and s_3 which rely on f_{N}). A demonstrative video can be found in [23]. We use the passivity condition proposed by [24] for our choices of different parameter-values. As this condition applies to the LuGre model first proposed in [25, 26] from which the elasto-plastic model evolved, further investigation is required to conclude whether these conditions are identical for the elasto-plastic model.

4.1. Sensel Morph

As mentioned in Section 1, the Sensel Morph (or Sensel for short) is used as an interface to control the bowed string (see Figure 6). The Sensel is a highly sensitive touch controller containing ca. 20,000 pressure sensitive sensors that allow for expressive control of the implementation [21].

4.2. Interaction

The first finger the Sensel registers is linked to the following parameters: the normal force of the bow f_{N} (finger pressure), the bowing velocity v_{B} (vertical finger velocity) and bowing position x_{B} (horizontal finger position). The parameters are limited by the following conditions: $0 \leq f_{\mathrm{N}} \leq 10$, $-0.3 \leq v_{\mathrm{B}} \leq 0.3$ and

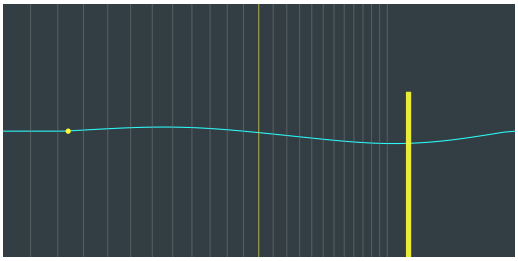

Figure 5: *The elasto-plastic bowed string application. The bow is shown as a yellow rectangle, moves on interaction and its opacity depends on the finger force. The state* $\mathbf{u}^n$ *is visualised using the cyan curve and stopping-finger position is shown as a yellow circle. The grey lines show the 'frets' corresponding to semi-tones as a visual reference for the stopping position and do not influence the model.*

Figure 6: *The Sensel Morph: an expressive touch sensitive controller used for controlling the real-time elasto-plastic bowed string implementation.*

$0 < x_{\mathrm{B}} < L$. The second finger acts as a stopping finger on the string. As done in [20], for a string stopped at location $x_{\mathrm{f}} \in [0, L]$ and $l_{\mathrm{f}} = \mathrm{floor}(x_{\mathrm{f}}/h)$ we use

$$u_l^n = \begin{cases} 0, & l = l_{\mathrm{f}} - 1 \vee l = l_{\mathrm{f}} \\ (1 - \alpha_{\mathrm{f}}^{\epsilon}) u_l^n, & l = l_{\mathrm{f}} + 1 \\ u_l^n, & \text{otherwise} \end{cases} \tag{23}$$

where $\alpha_{\mathrm{f}} = x_{\mathrm{f}}/h - l_{\mathrm{f}}$ and $\epsilon = 7$ is a heuristic value that has been found to most linearly alter pitch between grid points.

4.3. System Architecture

Implementation of the scheme shown in (12) starts by expanding the operators shown in (11) and solving for the state at the next sample $\mathbf{u}^{n+1}$ where $\mathbf{u}$ is a vector containing the values for all grid points $l \in [0, ..., N]$.

An overview of the system architecture can be found in Figure 7. The three main components of the application are the Sensel

Table 1: *Parameter values. Values for the fundamental frequency f_0 can be found in Section 5.*

Parameter	Symb. (unit)	Value (notes)
Material Density	ρ (kg·m^{-3})	7850
Radius	r (m)	$5 \cdot 10^{-4}$
String length	L (m)	1
Wave speed	c (m/s)	$2f_0/L$
Young's modulus	E (Pa)	$2 \cdot 10^{11}$
Freq. indep. damping	σ_0 (s^{-1})	1
Freq. dep. damping	σ_1 (m^2/s)	$5 \cdot 10^{-3}$
Coulomb friction	μ_C (-)	$0.3\ (< \mu_S)$
Static friction	μ_S (-)	$0.8\ (> \mu_C)$
Normal force	f_N (N)	10
Bow velocity	v_B (m/s)	0.1
Bow position	x_B (m)	0.25
Stribeck velocity	v_S (m/s)	0.1
Bristle stiffness	s_0 (N/m)	10^4
Bristle damping	s_1 (kg/s)	$0.001\sqrt{s_0}$
Viscous friction	s_2 (kg/s)	0.4
Noise coefficient	s_3 (-)	$0.02f_N$
Pseudorandom func.	w (N)	$-1 < w < 1$
Break-away disp.	z_{ba} (m)	$0.7f_C/s_0\ (< f_C/s_0)$
Sample rate	f_s (Hz)	44,100
Time step	k (s)	$1/f_s$

controlling the application, the violin string class that performs the simulation and the main application class that moderates between these and the auditory and visual outputs. The black arrows indicate instructions that one of these components can give to another and the hollow arrows indicate data flows. Moreover, the arrows are accompanied by coloured boxes, depicting what thread the instruction or data flow is associated with and at what rate this runs.

The graphics thread has the lowest priority, is denoted by the green boxes and runs at 15 Hz. The redraw instruction merely retrieves the current string state $\mathbf{u}^n$ and bow and finger position and visualises this as shown in Figure 5.

The thread checking and receiving data from the Sensel runs at 150 Hz and is denoted by the blue boxes. The parameters that the user interacts with (bowing force, velocity and position) are also updated at this rate.

The highest priority thread is the audio thread denoted by the orange boxes and runs at 44,100 Hz. The violin string class gets updated at this rate and performs operations in the order shown in Algorithm 1.

5. RESULTS AND DISCUSSION

Figure 8 shows the output waveforms for a string with $f_0 = 440$ Hz at different points along the string. The bowing parameters are $f_N = 5$ N and $v_B = 0.1$ m/s. The figure shows the traditional Helmholtz motion, which is the characteristic motion of a bowed string.

To test whether the implementation exhibits a hysteresis loop, the force vs. relative velocity plane was visualised. In Figure 9, this plot can be found for which the same parameters have been used. The figure shows values for 500 samples around $t = 0.5f_s$. As can be seen from the figure, the hysteresis loop is achieved and is similar to the one observed in [19]. The group of values around

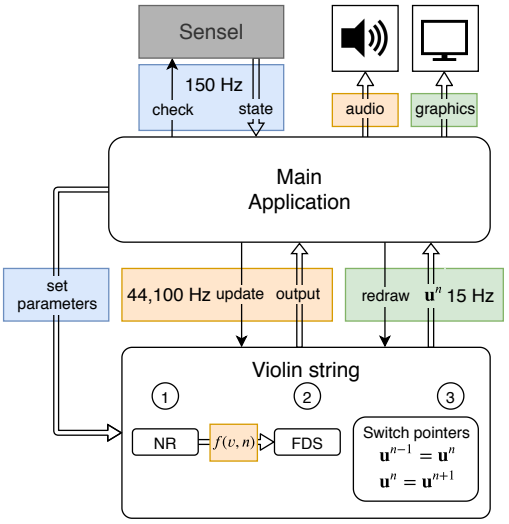

Figure 7: *The system architecture. See Section 4.3 for a thorough explanation.*

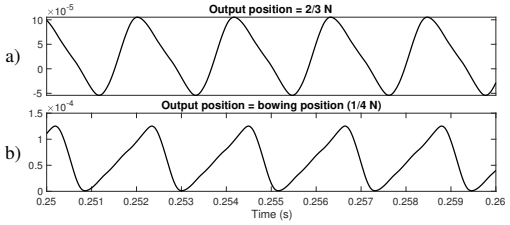

Figure 8: *Output waveforms of the simulation at different positions along the string where N denotes the number of points of the string ($f_0 = 440$ Hz, $f_N = 5$ N and $v_B = 0.1$ m/s).*

$v = 0$ are due to the sticking behaviour, and the others (the loop on the left) to the slipping behaviour.

For testing the speed of the algorithm, a MacBook Pro with a 2.2 GHz Intel Core i7 processor was used. The algorithm was tested using different frequencies according to the violin tuning of empty strings: $f_0 = 196.0$ (G3), 293.66 (D4), 440.0 (A4) and 659.26 (E5) Hz corresponding to $N = 95$, 71, 49, and 33 grid points respectively. The results can be seen in Table 2. When the total number of strings is smaller than 4, always the lowest frequency strings are used.

From Table 2 it can be observed that for one string, the CPU usage is $< 6\%$ with the graphics thread disabled. This is a great result, given the fact that both the bow and the string model are computationally complex. Empirical investigation shows that the NR algorithm converges after ca. 3-4 iterations and the capping of 50 iterations never has to be used. A single string (but also more) could thus safely be used as an audio plugin in parallel to others without the user having to worry about auditory dropouts.

for *t = 1:lengthSound* **do**
  calculate computable part b^n (Eq. (18))
  $\epsilon = 1$
  $i = 0$
  while $\epsilon < tol \wedge i < 50 \wedge f_C > 0$ **do**
    calculate..
    1. $z_{\text{ss}}(v^n_{(i)})$ (Eq. (7) in discrete-time)
    2. $\alpha(v^n_{(i)}, z^n_{(i)})$ (Eq. (8) in discrete-time)
    3. $r(v^n_{(i)}, z^n_{(i)})$ (Eq. (15))
    4. g_1, g_2 (Eqs. (19) and (20))
    5.–9. Compute derivatives of 1.–4. in the same order.
    10. Perform vector NR to obtain $v^n_{(i+1)}$ and $z^n_{(i+1)}$
    11. Calculate ϵ: $\epsilon = \left\| \begin{bmatrix} v^n_{(i+1)} \\ z^n_{(i+1)} \end{bmatrix} - \begin{bmatrix} v^n_{(i)} \\ z^n_{(i)} \end{bmatrix} \right\|$
    12. Increment i: $i = i + 1$
  end
  Repeat 1.–3. using the values for v^n and z^n from the NR iteration.
  Calculate $f(v^n, z^n)$ (Eq. (14))
  Calculate $\mathbf{u}^{n+1}$ (Eq. (12) expanded)
  $\mathbf{u}^{n-1} = \mathbf{u}^n$
  $\mathbf{u}^n = \mathbf{u}^{n+1}$
end

Algorithm 1: Pseudocode showing the order of calculations.

6. CONCLUSIONS

In this paper, we presented a real-time implementation of an elasto-plastic friction model with applications to a bow exciting a string, discretised using a finite-difference approach.

With a single string we are able to keep the CPU usage down to $< 6\%$ making for an efficient implementation that could be used in parallel with other virtual instruments or plugins.

Future work includes parameter design and including an instrument body for more realistic sounding results, as well as listening tests to verify the perceivable differences between simpler friction models versus the elasto-plastic model.

7. ACKNOWLEDGMENTS

Many thanks to the anonymous reviewers for giving their valuable input. This work is supported by NordForsk's Nordic University Hub Nordic Sound and Music Computing Network NordicSMC, project number 86892.

8. REFERENCES

[1] P. Dupont, V. Hayward, B. Armstrong, and F. Altpeter, "Single state elastoplastic friction models," *IEEE Transactions on Automatic Control*, vol. 47, no. 5, pp. 787–792, 2002.

[2] S. Serafin, F. Avanzini, and D. Rocchesso, "Bowed string simulation using an elasto-plastic friction model," *SMAC 03*, pp. 95–98, 2003.

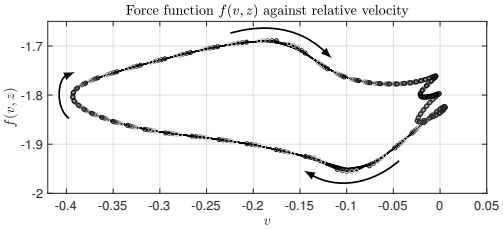

Figure 9: *Hysteresis loop showing 500 values. The values around $v = 0$ are due to sticking behaviour and the loop on the left is due to slipping behaviour.*

Table 2: *CPU usage for different amounts of strings. The values are averages over a 10 s period both for the enabled and disabled graphics thread. All strings are bowed simultaneously (polyphonically).*

Amount of strings	Graphics (%)	No graphics (%)
1	44.8	5.95
2	47.7	9.54
3	52.8	12.1
4	60.9	17.9

[3] J. Woodhouse, "Bowed string simulation using a thermal friction model," *Acta Acustica united with Acustica*, vol. 89, pp. 355–368, 2003.

[4] S. Serafin, *The Sound of Friction: Real-time Models, Playability and Musical Applications*, Ph.D. thesis, CCRMA, 2004.

[5] F. Avanzini, S. Serafin, and D. Rocchesso, "Interactive simulation of rigid body interaction with friction-induced sound generation," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 5, pp. 1073–1080, 2005.

[6] M.E. McIntyre, R.T. Schumacher, and J. Woodhouse, "On the oscillations of musical instruments," *Journal of the Acoustical Society of America*, vol. 74, no. 5, pp. 1325–1345, 1983.

[7] J.O. Smith, "Efficient simulation of the reed bore and bow string mechanics," *ICMC 86*, pp. 275–280, 1986.

[8] J.-L. Florens, A. Razafindrakoto, A. Luciani, and C. Cadoz, "Optimized real time simulation of objects for musical synthesis and animated image synthesis," *ICMC 86*, pp. 65–70, 1986.

[9] J.O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, vol. 16, no. 4, pp. 74–91, 1992.

[10] J.D. Morrison and J.-M. Adrien, "Mosaic: A framework for modal synthesis," *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, 1993.

[11] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I," *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 and A. Askenfelt, "Numerical simulations of struck strings. I. a physical model for a struck string using finite difference methods," *JASA*, vol. 95, no. 2, pp. 1112–1118, 1994.

[14] S. Bilbao, *Numerical Sound Synthesis*, J. Wiley & Sons, 2009.

[15] S. Bilbao, R. Hamilton, B and. Harrison, and A. Torin, "Finite-difference schemes in musical acoustics: A tutorial," in *Springer Handbook of Systematic Musicology*, chapter 19, pp. 349–384. Springer, 2018.

[16] 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.

[17] C. G. M. Desvages, *Physical Modelling of the Bowed String and Applications to Sound Synthesis*, Ph.D. thesis, The University of Edinburgh, 2017.

[18] C. Desvages and S. Bilbao, "Two-polarisation finite difference model of bowed strings with nonlinear contact and friction forces," *Proceedings of the International Conference on Digital Audio Effects*, 2015.

[19] J.H. Smith and J. Woodhouse, "The tribology of rosin," *Journal of the Mechanics and Physics of Solids*, vol. 48, pp. 1633–1681, 2000.

[20] S. Willemsen, N. Andersson, S. Serafin, and S. Bilbao, "Real-time control of large-scale modular physical models using the sensel morph," *Proc. of the 16th Sound and Music Computing Conference*, pp. 275–280, 2019.

[21] Sensel Inc., "Sensel Morph," Available at https://sensel.com/, accessed April 01, 2019.

[22] JUCE ROLI, "JUCE," Available at https://juce.com/, accessed April 04, 2019.

[23] S. Willemsen, "Elasto-Plastic Bow Model acting on a Finite-Difference Stiff String," Available at https://www.youtube.com/watch?v=-5bDebCW1Qg, accessed April 06, 2019.

[24] K. J. Åström and C. Canudas de Wit, "Revisiting the lugre friction model," *IEEE Control Systems Magazine*, vol. 28, no. 6, pp. 101–114, 2008.

[25] C. Canudas de Wit, H. Olsson, K. J. Åström, and P. Lischinsky, "Dynamic friction models and control design," *Proceedings of the 1993 American Control Conference*, pp. 1920–1926, 1993.

[26] C. Canudas de Wit, H. Olsson, K. J. Åström, and P. Lischinsky, "A new model for control of systems with friction," *IEEE Transactions on Automatic Control*, vol. 40, no. 3, pp. 419–425, 1995.

Paper E

Real-time Implementation of a Physical Model of the Tromba Marina

REAL-TIME IMPLEMENTATION OF A PHYSICAL MODEL OF THE TROMBA MARINA

Silvin Willemsen and Stefania Serafin
Multisensory Experience Lab, CREATE
Aalborg University Copenhagen
{sil, sts}@create.aau.dk

Stefan Bilbao and Michele Ducceschi
Acoustics and Audio Group
University of Edinburgh
{s.bilbao, michele.ducceschi}@ed.ac.uk

ABSTRACT

The tromba marina is a medieval bowed monochord instrument. The string of the instrument rests on a rattling bridge that, due to the collision with the body, creates a trumpet-like sound. This paper presents a real-time implementation of a physical model of the tromba marina. The goal of the simulation is to make the instrument accessible to a larger audience. The physical model is implemented using finite-difference time-domain (FDTD) methods and non-iterative collision methods. A real-time implementation of the instrument is also presented. The simulation exhibits brass-like qualities and sounds similar to a real tromba marina, but requires further testing to validate the realism.

1. INTRODUCTION

The tromba marina (see Figure 1) is a medieval bowed monochord instrument with a long quasi-trapezoidal body and a uniquely fashioned bridge (often called a shoe, because of its shape – see Figure 2). The name of the instrument derives from the fact that *tromba* means *trumpet* in Italian. A peculiarity of the instrument is that a foot of the bridge is free to rattle against the soundboard in sympathy with the vibrating string. This unusual bridge creates a trumpet-like sound. The frequency produced by the instrument is varied by placing the side of the knuckle of the non-dominant hand, lightly, at specific nodal points on the string, in order to select various harmonics of the open string. The dominant hand controls the bow, which is drawn across the string above the non-dominant hand [1].

In this paper, we present a real-time implementation of a physical model of the tromba marina. One of the ultimate goals is the emulation of an instrument that, due to its rarity, is not accessible to a large audience.

Physical modelling for sound synthesis has a long history. Various techniques have been developed to simulate real-world instruments, including mass-spring systems [2], digital waveguides [3] and modal synthesis [4]. Finite-difference time-domain (FDTD) methods were first used for sound synthesis by Hiller and Ruiz in [5–7], later by Chaigne et al. in [8, 9] and elaborated upon by Bilbao and colleagues in [10, 11]. Compared with other techniques,

Figure 1. The tromba marina from the Danish Music Museum in Copenhagen.

Figure 2. The bridge of the tromba marina from the Danish Music Museum in Copenhagen. The right side is pressed against the body by the string while the left side is free and can rattle against the body.

FDTD methods are more computationally expensive, but easily generalisable and flexible—no assumptions of linearity of travelling wave solutions are employed. Our goal is to implement these techniques in real time and thereby make the simulations playable for the users. For this purpose, we use the expressive Sensel Morph controller [12]. Other work in real-time control of FDTD methods using this controller includes [13].

The emulation of nonlinear collision interactions in musical instruments normally requires the use of iterative solvers (such as the Newton-Raphson algorithm) [14]. For the nonlinear collisions present in the instrument, a method recently proposed in the field of audio by Lopes and Falaize in [15–17] and later by Ducceschi and Bilbao in [18] allows such iterative methods to be sidestepped. It is thus suited to creating a real-time implementation of the tromba marina.

This paper is structured as follows: Section 2 presents the models used and Section 3 shows the discretisation of these. Section 4 provides information about implementation, parameter choices, the graphical user interface and control and mapping. Section 5 shows the results and discusses these. Concluding remarks and future work are presented in Section 6.

2. MODELS

The tromba marina can be subdivided into three main components: the string, the bridge and the body. In this section, the partial differential equations (PDEs) of the different components in isolation, under zero-input conditions, will be of the form

$$\mathcal{L}q = 0. \tag{1}$$

Here, $q = q(\boldsymbol{x}, t)$ represents the state of the component at time t and spatial coordinate $\boldsymbol{x} \in \mathcal{D}$, where the dimensions of domain $\mathcal{D}$ depend on the component at hand. Furthermore, $\mathcal{L}$ is a partial differential operator. (Subscripts 's', 'm' and 'p' used subsequently indicate that (1) applies to the string, bridge (mass) or body (plate), respectively.)

2.1 Bowed Stiff String

Consider a damped stiff string of length L (m), with domain $\mathcal{D} = \mathcal{D}_\mathrm{s} = [0, L]$ and state variable $q = u(\chi, t)$. With reference to (1), we define the operator $\mathcal{L} = \mathcal{L}_\mathrm{s}$ as [10]

$$\mathcal{L}_\mathrm{s} = \rho_\mathrm{s} A \partial_t^2 - T\partial_\chi^2 + E_\mathrm{s} I \partial_\chi^4 + 2\rho_\mathrm{s} A \sigma_{0,\mathrm{s}} \partial_t - 2\rho_\mathrm{s} A \sigma_{1,\mathrm{s}} \partial_t \partial_\chi^2. \tag{2}$$

Here, ∂_t and ∂_χ indicate partial differentiation with respect to t and χ. The various parameters appear as: material density ρ_s (kg·m^{-3}), cross-sectional area $A = \pi r^2$ (m^2), radius r (m), tension $T = (2f_{0,\mathrm{s}}L)^2 \rho_\mathrm{s} A$ (N), [1] fundamental frequency $f_{0,\mathrm{s}}$ (s^{-1}), Young's modulus E_s (Pa), area moment of inertia $I = \pi r^4/4$ (m^4), and loss coefficients $\sigma_{0,\mathrm{s}}$ (s^{-1}) and $\sigma_{1,\mathrm{s}}$ (m^2/s). We set the boundary conditions to be simply supported so that

$$u = \partial_\chi^2 u = 0 \quad \text{for} \quad \chi = 0, L. \tag{3}$$

[1] Even though this definition for T from the fundamental frequency $f_{0,\mathrm{s}}$ is only valid for a simply supported string without stiffness, the effect of the stiffness eventually chosen for $f_{0,\mathrm{s}}$ is negligible.

As the string is excited using a bow, Equation (1) may be augmented as [10]

$$\mathcal{L}_\mathrm{s} u = -\delta(\chi - \chi_\mathrm{b}) F_\mathrm{b} \Phi(v_\mathrm{rel}), \tag{4}$$

with externally supplied downward bow force $F_\mathrm{b} = F_\mathrm{b}(t)$ (N), spatial Dirac delta function $\delta(\chi - \chi_\mathrm{b})$ (m) selecting the bow position $\chi_\mathrm{b} = \chi_\mathrm{b}(t) \in \mathcal{D}_\mathrm{s}$ (m) and dimensionless friction characteristic

$$\Phi(v_\mathrm{rel}) = \sqrt{2a} v_\mathrm{rel} e^{-a v_\mathrm{rel}^2 + 1/2}, \tag{5}$$

with free parameter a. The relative velocity between the string at bow location χ_b and the externally supplied bow velocity $v_\mathrm{b} = v_\mathrm{b}(t)$ (m/s) is defined as

$$v_\mathrm{rel} = \partial_t u(\chi_\mathrm{b}, t) - v_\mathrm{b}. \tag{6}$$

2.2 Bridge

The bridge is modelled as a simple mass-spring-damper system. As this system is point-like, or zero-dimensional, the state variable $q = w(t)$ and the definition of domain $\mathcal{D}$ is unnecessary. The operator $\mathcal{L} = \mathcal{L}_\mathrm{m}$ is defined as

$$\mathcal{L}_\mathrm{m} = M\frac{d^2}{dt^2} + M\omega_0^2 + MR\frac{d}{dt}, \tag{7}$$

with mass M (kg), linear angular frequency of oscillation $\omega_0 = 2\pi f_{0,\mathrm{m}}$, (s^{-1}), fundamental frequency $f_{0,\mathrm{m}}$ (s^{-1}) and damping coefficient R (s^{-1}).

2.3 Body

The body is simplified to a two-dimensional plate with side-lengths L_x and L_y, domain $\mathcal{D} = \mathcal{D}_\mathrm{p} = [0, L_x] \times [0, L_y]$ and state variable $q = z(x, y, t)$. Using the 2D Laplacian

$$\Delta \triangleq \partial_x^2 + \partial_y^2, \tag{8}$$

the operator $\mathcal{L} = \mathcal{L}_\mathrm{p}$ can be defined as [10]

$$\mathcal{L}_\mathrm{p} = \rho_\mathrm{p} H \partial_t^2 + D\Delta\Delta + 2\rho_\mathrm{p} H \sigma_{0,\mathrm{p}} \partial_t - 2\rho_\mathrm{p} H \sigma_{1,\mathrm{p}} \partial_t \Delta, \tag{9}$$

with material density ρ_p (kg·m^{-3}), plate thickness H (m), stiffness coefficient $D = E_\mathrm{p} H^3/12(1-\nu^2)$, Young's modulus E_p (Pa), dimensionless Poisson's ratio ν, and loss coefficients $\sigma_{0,\mathrm{p}}$ (s^{-1}) and $\sigma_{1,\mathrm{p}}$ (m^2/s). The boundary conditions of the plate are set to be clamped so that

$$z = \mathbf{n} \cdot \nabla z = 0. \tag{10}$$

where ∇z is the gradient of z, and where $\mathbf{n}$ indicates a normal to the plate area at the boundary.

2.4 Collisions

It can be argued that the greatest contributor to the characteristic sound of the tromba marina is the rattling bridge colliding with the body. A diagram of the bridge with important parts highlighted can be found in Figure 3. A collision can be modelled by including a term to the PDEs mentioned above describing the potential energy of the system

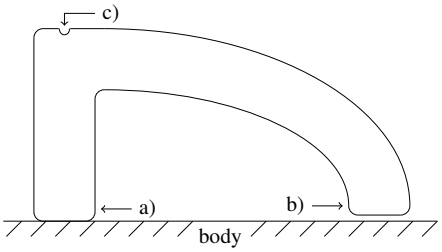

Figure 3. Diagram of the bridge while rattling (view from top of the tromba marina). Indicated are: a) the pivoting point always in contact with the body, b) the rattling point colliding with the body (currently not colliding), and c) the string cavity straight above the middle of the pivoting point.

(further referred to as *the potential*) [18]. For the bridge-body (mass-plate) interaction this potential is defined as follows

$$\phi_{\text{mp}}(\eta_{\text{mp}}) = \frac{K_{\text{mp}}}{\alpha_{\text{mp}}+1}[\eta_{\text{mp}}]_+^{\alpha_{\text{mp}}+1}, \tag{11}$$

$$K_{\text{mp}} > 0, \quad \alpha_{\text{mp}} \geq 1, \quad \eta_{\text{mp}} \triangleq z(x_{\text{mp}}, y_{\text{mp}}, t) - w(t)$$

where K_{mp} is the collision stiffness (N/m if $\alpha_{\text{mp}} = 1$), α_{mp} is the dimensionless nonlinear collision coefficient, and $\eta_{\text{mp}} = \eta_{\text{mp}}(t)$ is the distance between the rattling part of the bridge and the body at the point of collision (m). Furthermore, $[\eta_{\text{mp}}]_+ = 0.5(\eta_{\text{mp}} + |\eta_{\text{mp}}|)$ is the positive part of η_{mp}. Note that penalty methods are employed here, where a positive η_{mp}, i.e., interpenetration of the colliding objects, is intended [19]. The term which can then be included in the PDEs is $\phi'_{\text{mp}} = d\phi_{\text{mp}}/d\eta_{\text{mp}}$. As described in [15–18], using this form of the potential requires using iterative methods for solving its discrete counterpart. In [18], the authors propose to rewrite the potential to

$$\psi = \sqrt{2\phi}, \tag{12}$$

and the term included in the PDEs to

$$\phi' = \psi\psi' = \psi\frac{d\psi}{d\eta} \xrightarrow{\text{chain rule}} \psi\frac{\dot{\psi}}{\dot{\eta}}, \tag{13}$$

where the dot above ψ and ϕ denotes a single time derivative. Equation (13), as can be seen in Section 3, leads to guaranteed stable and explicitly computable simulation algorithms without the need for iterative solvers.

As the string rests on the bridge, the interaction between these components needs to be modelled as well. Even though the bridge-body interaction is perpendicular to the string-bridge interaction, we can model them as being parallel, assuming that a "horizontal" movement of the string causes a "vertical" movement of the rattling part of the bridge. We can use an alternative version of the potential in Equation (11) described in [20] to make the collision two-sided acting as a connection:

$$\phi_{\text{sm}}(\eta_{\text{sm}}) = \frac{K_{\text{sm}}}{\alpha_{\text{sm}}+1}|\eta_{\text{sm}}|^{\alpha_{\text{sm}}+1}, \tag{14}$$

$$K_{\text{sm}} > 0, \quad \alpha_{\text{sm}} \geq 1, \quad \eta_{\text{sm}} \triangleq w(t) - u(\chi_{\text{sm}}, t)$$

where $\eta_{\text{sm}} = \eta_{\text{sm}}(t)$ is the distance between the string at the location of the bridge and the bridge itself.

2.5 Complete System

A complete system for the tromba marina may be written, in continuous-time as:

$$\begin{cases} \mathcal{L}_{\text{s}} u & = -\delta(\chi - \chi_{\text{b}})F_{\text{b}}\Phi(v_{\text{rel}}) \\ & \quad + \delta(\chi - \chi_{\text{sm}})\psi_{\text{sm}}\psi'_{\text{sm}} & \text{(15a)} \\ \mathcal{L}_{\text{m}} w & = -\psi_{\text{sm}}\psi'_{\text{sm}} + \psi_{\text{mp}}\psi'_{\text{mp}}, & \text{(15b)} \\ \mathcal{L}_{\text{p}} z & = -\delta(x - x_{\text{mp}}, y - y_{\text{mp}})\psi_{\text{mp}}\psi'_{\text{mp}}, & \text{(15c)} \\ \eta_{\text{sm}} & = w(t) - u(\chi_{\text{sm}}, t), & \text{(15d)} \\ \eta_{\text{mp}} & = z(x_{\text{mp}}, y_{\text{mp}}, t) - w(t), & \text{(15e)} \end{cases}$$

where $\chi_{\text{sm}} \in \mathcal{D}_{\text{s}}$ is the location of the bridge along the string and $(x_{\text{mp}}, y_{\text{mp}}) \in \mathcal{D}_{\text{p}}$ is the location on the body with which the bridge collides.

3. DISCRETISATION

System (15) is discretised using FDTD methods. These methods subdivide the continuous system in grid points in space and samples in time. Before going into the discretisation of the models, and collision and connection terms in the system described in (15), some finite difference operators are introduced.

3.1 Operators

The identity and temporal shift operators are defined as

$$1\eta^n = \eta^n, \quad e_{t+}\eta^n = \eta^{n+1}, \quad e_{t-}\eta^n = \eta^{n-1}. \tag{16}$$

Using these, the operators for the forward, backward and centered time differences can be defined as

$$\delta_{t+} = \frac{e_{t+} - 1}{k}, \ \delta_{t-} = \frac{1 - e_{t-}}{k}, \ \delta_{t\cdot} = \frac{e_{t+} - e_{t-}}{2k}, \tag{17}$$

and are all approximations to a first-order time derivative. Furthermore, forwards and backwards averaging operators are defined as

$$\mu_{t+} = \frac{e_{t+} + 1}{2}, \quad \mu_{t-} = \frac{1 + e_{t-}}{2}. \tag{18}$$

and can be used to describe interleaved grid points $n+1/2$ and $n-1/2$ respectively.

3.2 Discrete Models

To approximate the state of a system in isolation we use

$$q(\boldsymbol{x}, t) \approx q_{\boldsymbol{l}}^n, \tag{19}$$

where grid function $q_{\boldsymbol{l}}^n$ is a discrete approximation to $q(\boldsymbol{x}, t)$ at $t = nk$ with time step k (s), time index $n \geq 0$ and grid location $\boldsymbol{l}$ that depends on domain $\mathcal{D}$ of the system at hand. In the case of the string, we use $\chi = lh_{\text{s}}$ with grid spacing h_{s} (m), $\boldsymbol{l} = l \in [0, \ldots, N]$ and total number of grid points $N = L/h_{\text{s}}$ to yield $u(\chi, t) \approx u_l^n$.

In the case of the body, we use $x = lh_\text{p}$ and $y = mh_\text{p}$ to get $z(x, y, t) \approx z^n_{(l,m)}$ where $\boldsymbol{l} = (l, m)$ with $l \in [0, \ldots, N_x]$ and $m \in [0, \ldots, N_y]$. Here, $N_x = L_x/h_\text{p}$ and $N_y = L_y/h_\text{p}$ are the horizontal and vertical number of grid points respectively with grid spacing h_p (m).

The discretisation of and expansion of operator $\mathcal{L} \approx \ell$ in the case of stiff strings, mass-spring systems and plates using FDTD methods are well covered in the literature [10] and will not be described in detail in this paper. To obtain the highest accuracy possible while keeping the system explicit (except for the bow), centered differences – which are second-order accurate – have been chosen where possible.

For stability, grid spacings h_s and h_p should satisfy the conditions below. In the case of the damped stiff string,

$$h_\text{s} \geq \sqrt{\frac{c^2k^2 + 4\sigma_{1,\text{s}}k + \sqrt{(c^2k^2 + 4\sigma_{1,\text{s}}k)^2 + 16\kappa_\text{s}^2k^2}}{2}}, \tag{20}$$

with wave speed $c = \sqrt{T/\rho_\text{s}A}$ and stiffness coefficient $\kappa_\text{s} = \sqrt{E_\text{s}I/\rho_\text{s}A}$ and in the case of the plate,

$$h_\text{p} \geq 2\sqrt{k\left(\sigma_{1,\text{s}} + \sqrt{\kappa_\text{p}^2 + \sigma_{1,\text{s}}^2}\right)}, \tag{21}$$

with stiffness coefficient $\kappa_\text{p} = \sqrt{D/\rho_\text{p}H}$. The closer the grid spacings are to these conditions, the higher the accuracy of the approximation.

In order to discretise the Dirac delta functions found in system (15) we introduce a spreading operator $J(\boldsymbol{x}_\text{c})$ that applies a force to coordinate $\boldsymbol{x}_\text{c}$, which, in the simplest case, is defined as [10]

$$J(\boldsymbol{x}_\text{c}) = \begin{cases} \frac{1}{h^d}, & \boldsymbol{l} = \boldsymbol{l}_\text{c} = \text{round}(\boldsymbol{x}_\text{c}/h) \\ 0, & \text{otherwise} \end{cases} \tag{22}$$

Here, d is the number of dimensions of domain $\mathcal{D}$ that $\boldsymbol{x}$ is defined for, i.e., $d = 0$ for the bridge, $d = 1$ for the string, and $d = 2$ for the plate. For finer control, a cubic spreading operator J_3 can be introduced [10]. This is used for the bowing term in Equation (4), which is discretised as follows

$$\ell_\text{s} u^n_l = -J_3(\chi_\text{b})F^n_\text{b}\phi(v^n_\text{rel}) \tag{23}$$

where, using the centered difference operator from Equation (17),

$$v^n_\text{rel} = \delta_{t\cdot}u^n_{l_\text{b}} - v^n_\text{b}, \tag{24}$$

with coordinate $l_\text{b} = \chi_\text{b}/h_\text{s}$. Equation (24) needs to be calculated using iterative methods.

3.3 Collisions using Non-Iterative Methods

For the discrete-time definitions of the potential in (13) we can use

$$\psi \approx \mu_{t+}\psi^{n-1/2} \quad \text{and} \quad \psi' \approx \frac{\delta_{t+}\psi^{n-1/2}}{\delta_{t\cdot}\eta^n}, \tag{25}$$

where ψ at interleaved grid point $n - 1/2$ is defined as

$$\psi^{n-1/2} = \mu_{t-}\psi^n. \tag{26}$$

Note that applying a forward or backward difference operator to an interleaved grid – such as $\delta_{t+}\psi^{n-1/2}$ in Equation (25) – is second-order accurate.

For a system that has a single (upward) collision we get

$$\ell q^n_{\boldsymbol{l}} = J(\boldsymbol{x}_\text{c})\left(\mu_{t+}\psi^{n-1/2}\right)\frac{\delta_{t+}\psi^{n-1/2}}{\delta_{t\cdot}\eta^n}. \tag{27}$$

Here, we use the identity

$$\mu_{t+}\psi^{n-1/2} = \frac{k}{2}\delta_{t+}\psi^{n-1/2} - \psi^{n-1/2} \tag{28}$$

and define

$$g^n = \frac{\delta_{t+}\psi^{n-1/2}}{\delta_{t\cdot}\eta^n}, \tag{29}$$

which can be rewritten to

$$\delta_{t+}\psi^{n-1/2} = g^n\delta_{t\cdot}\eta^n. \tag{30}$$

Then, inserting (30) into (28) and this together with (29) into (27) we get

$$\ell q^n_{\boldsymbol{l}} = J(\boldsymbol{x}_\text{c})\left(\frac{k}{2}g^n\delta_{t\cdot}\eta^n - \psi^{n-1/2}\right)g^n \tag{31}$$

where g^n may be explicitly calculated using the analytic expressions for ψ and ϕ [18]:

$$g^n = \psi'\Big|_{\eta=\eta^n} = \frac{\phi'}{\sqrt{2\phi}}\Big|_{\eta=\eta^n}. \tag{32}$$

Numerical stability of this scheme is shown in [18]. When writing out (32) we can obtain definitions for g^n_sm using (14)

$$g^n_\text{sm} = \text{sgn}(\eta^n_\text{sm})\sqrt{\frac{K_\text{sm}(\alpha_\text{sm}+1)}{2}}|\eta^n_\text{sm}|^{\frac{\alpha_\text{sm}-1}{2}}, \tag{33}$$

and g^n_mp using (11)

$$g^n_\text{mp} = \sqrt{\frac{K_\text{mp}(\alpha_\text{mp}+1)}{2}}[\eta^n_\text{mp}]_+^{\frac{\alpha_\text{mp}-1}{2}}. \tag{34}$$

3.4 Complete Discrete System

Introducing for brevity,

$$\xi^n = \frac{k}{2}g^n\delta_{t\cdot}\eta^n - \psi^{n-1/2}, \tag{35}$$

the discrete counterpart of the complete system described in (15) will be

$$\begin{cases} \ell_\text{s}u^n_l &= -J_3(\chi^n_\text{b})F_\text{b}\Phi(v^n_\text{rel}) + J(\chi_\text{sm})\xi^n_\text{sm}g^n_\text{sm}, & \text{(36a)} \\ \ell_\text{m}w^n &= -\xi^n_\text{sm}g^n_\text{sm} + \xi^n_\text{mp}g^n_\text{mp}, & \text{(36b)} \\ \ell_\text{p}z^n_{(l,m)} &= -J(x_\text{mp}, y_\text{mp})\xi^n_\text{mp}g^n_\text{mp}, & \text{(36c)} \\ \eta^n_\text{sm} &= w^n - u^n_{l_\text{sm}}, & \text{(36d)} \\ \eta^n_\text{mp} &= z^n_{(l_\text{mp}, m_\text{mp})} - w^n, & \text{(36e)} \end{cases}$$

where discrete counterparts of connection and collision locations in Equations (36d) and (36e) are described as $l_\text{sm} = \chi_\text{sm}/h_\text{s}$ and $(l_\text{mp}, m_\text{mp}) = (x_\text{mp}/h_\text{p}, y_\text{mp}/h_\text{p})$. This leaves

us with two different types of update equations, one where $q_{\boldsymbol{l}}^{n+1}$ is calculated and one where $\psi^{n+1/2}$ is calculated.

One might think that due to the centered differences $\delta_{t\cdot}\eta^n$ still present in Equation (35), our system remains implicit, but as we can insert the definitions for Equations (36d) and (36e) evaluated at the next time index $n+1$, which are already present in $\ell_{\mathrm{s}}u_l^n$, $\ell_{\mathrm{m}}w^n$ and $\ell_{\mathrm{p}}z_{(l,m)}^n$, the Equations in (36) reduce to a system of linear equations that can be solved by a single division.

4. IMPLEMENTATION

The real-time implementation of the system has been done in C++ using the JUCE framework [21] and will be controlled using the Sensel Morph (or simply Sensel) – an expressive touch controller. A demo of the application can be found in [22]. This section will first elaborate some important considerations regarding the setup of the system. Then, the algorithm together with the parameter design will be presented. Finally, the graphical user interface (GUI) will be detailed together with the Sensel and its mapping to the application.

4.1 Introducing an Offset

Firstly, for more realistic and expressive sounds, we model the bridge – and with that, the string – to rest slightly above the body. Expanding $\ell_{\mathrm{m}}w^n$ in (36b) and including the offset yields

$$\ell_{\mathrm{m}}w^n \Rightarrow M\delta_{tt}w^n + M\omega_0^2(w^n - w_{\mathrm{off}}) + MR\delta_{t\cdot}w^n \quad (37)$$

where $w_{\mathrm{off}} \geq 0$ is a predefined offset between the body and the bridge. Furthermore, the second-order time derivative can be defined from the definitions in (17) as

$$\delta_{tt} = \delta_{t+}\delta_{t-}\,. \quad (38)$$

The boundary condition of the string defined in Equation (3) will also change depending on the bridge offset:

$$u = w_{\mathrm{off}} \quad \text{and} \quad \partial_x^2 u = 0. \quad (39)$$

4.2 Pitch Control

Secondly, as briefly mentioned in Section 1, the way that different pitches are played on the tromba marina, is to slightly rest a knuckle or finger on nodal points along the string to induce harmonics. Thus, a damping finger is implemented. Using the cubic interpolation operator I_3 [10], Equation (36a) can be extended to

$$\ell_{\mathrm{s}}u_l^n = \ldots - J_3(\chi_{\mathrm{f}})I_3(\chi_{\mathrm{f}})\sigma_{\mathrm{f}}(u_l^n - w_{\mathrm{off}}), \quad (40)$$
$$\text{where} \quad 0 \leq \sigma_{\mathrm{f}} \leq 1,$$

which essentially subtracts its own state at location $\chi_{\mathrm{f}} \in [0, 0.5\chi_{\mathrm{sm}}]$ according to the damping coefficient σ_{f} (kg$\cdot$s^{-2}) applied. As done in [13], the fractional part used in the spreading operator ($\alpha_{\mathrm{i}} = \chi_{\mathrm{f}}/h - \mathrm{floor}(\chi_{\mathrm{f}}/h)$) is raised to the 7th power as it has been found to scale finger position to pitch more properly in the context of FDTD. As the string is bowed above the damping finger (at the other side of the rattling bridge) it is essential that the energy from the bow reaches the rattling bridge, which is still the case for lower values of σ_{f}. A more realistic approach that could be investigated is to model the finger as a mass colliding with the string, rather than imposing the damping directly to the state of the string as presented here.

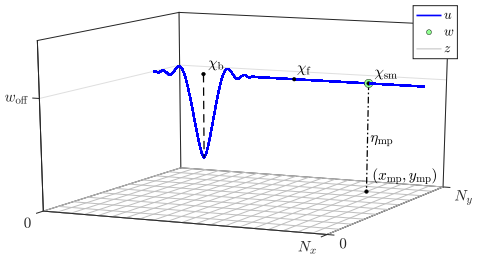

Figure 4. The virtual system in (36) including the offset in Equation (37) and the damping finger in Equation (40), with different important coordinates highlighted. Note that η_{sm} (Equation (36d)) is not shown as it is close to 0 at all times.

A schematic plot of the full system, including the offset described in Equations (37) and (39) and the damping finger from Equation (40) can be found in Figure 4.

4.3 Other Considerations

Realistic initialisation of both η_{sm} and η_{mp} is essential. In this case (at $n = 0$) $\eta_{\mathrm{sm}}^0 = 0$ and $\eta_{\mathrm{mp}}^0 \leq 0$ so that no collision is present at initialisation.

After h_{p} is calculated in Equation (21), we check whether it is smaller than a set value $h_{\mathrm{p,min}} = 0.01$. This reduces the quality of the model, but increases the speed, ultimately allowing for real-time implementation.

4.4 Order of Calculation

The order of calculation is shown in the pseudocode in Algorithm 1. In theory, in order to iteratively calculate the bow force, the collision and connection forces should be included in this. However, as the string is practically never bowed at the bridge position χ_{sm}, these can be calculated independently.

4.5 Parameter Design

The list of parameters used in the implementation can be found in Table 1. As the authors had a real (recreated) tromba marina (presented in [23]) at their disposal, some parameters have been measured in accordance to the real instrument. The others have been tuned by ear by one of the authors.

Regarding the output of the system, through informal testing it was decided to retrieve the output from the state of the plate right at the point of collision $z_{\mathrm{out}} = (l_{\mathrm{mp}}, m_{\mathrm{mp}})$ combined with the sound of the string at $u_{\mathrm{out}} = L - \chi_{\mathrm{sm}}$ at a lower volume. It can be argued that the loudest sound comes from the collision between the bridge and the body

while *application is running* **do**

1. calculate schemes	(ℓq in Eqs. (36a-c))
2. apply bow to string	(Eq. (36a))
3. apply damping finger	(Eq. (40))
4. calculate g^n_{sm} and g^n_{mp}	(Eqs. (33) and (34))
5. calculate collision and connection forces and add to schemes	(Eqs. (36a-c))
6. Update states	$\boldsymbol{q}^{n-1} = \boldsymbol{q}^n$ $\boldsymbol{q}^n = \boldsymbol{q}^{n+1}$ $\boldsymbol{\psi}^{n-1/2} = \boldsymbol{\psi}^{n+1/2}$

end

Algorithm 1: Pseudocode showing the order of calculation after initialisation. Bold symbols denote the collection of states of the entire system ($\boldsymbol{q}$) and potentials ($\boldsymbol{\psi}$).

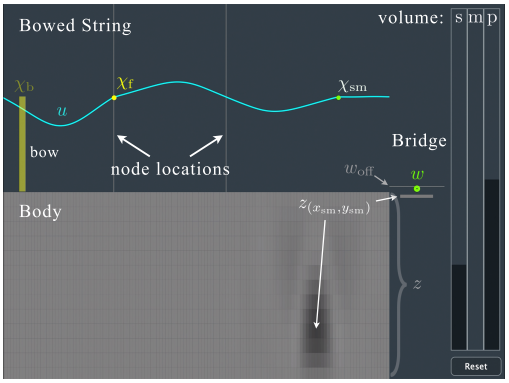

Figure 5. The GUI showing the excited system with components highlighted. A more detailed description can be found in Section 4.6.

making it logical to select this point as the main sound source.

4.6 Graphical User Interface

A screenshot of the GUI is shown in Figure 5. The GUI is divided in four sections, three showing the states of the string, bridge and body respectively and one control section.

Firstly, the string section shows the state of the string u as a cyan-coloured path and the bow as a yellow rectangle with bow position χ_{b} and its opacity depending on the bow force F_{b}. Furthermore, the bridge state w^n is shown as a green circle at location (of the bridge along the string) χ_{sm}. Finally, the position of the damping finger χ_{f} is displayed as a yellow circle, the size of which depends on damping coefficient σ_{f}. The position of the finger triggers lines showing the locations of the closest nodes along the

Name	Symbol (unit)	Value
String		
Length	L (m)	1.90*
Material density	ρ_{s} (kg·m^{-3})	7850
Radius	r (m)	0.0005
Fundamental freq.	f_0 (s^{-1})	32*
Young's modulus	E_{s} (Pa)	$2 \cdot 10^{11}$
Freq. indep. loss	$\sigma_{0,\text{s}}$ (s^{-1})	0.1
Freq. dep. loss	$\sigma_{1,\text{s}}$ (m^2/s)	0.05
Bow		
Bow force	F_{b} (N)	$0 \leq F_{\text{b}} \leq 0.1$
Bow velocity	v_{b} (m/s)	$-0.5 \leq v_{\text{b}} \leq 0.5$
Free parameter	a (-)	100
Bridge		
Mass	M (kg)	0.001
Fundamental freq.	$f_{0,\text{m}}$ (s^{-1})	500
Damping	R (s^{-1})	0.05
Body		
Length	L_x (m)	1.35*
Width	L_y (m)	0.18*
Material density	ρ_{p} (kg·m^{-3})	50
Thickness	H (m)	0.01
Young's modulus	E_{p} (Pa)	$2 \cdot 10^5$
Poisson's ratio	ν (-)	0.3
Freq. indep. loss	$\sigma_{0,\text{p}}$ (s^{-1})	2
Freq. dep. loss	$\sigma_{1,\text{p}}$ (m^2/s)	0.05
Min. grid spacing	$h_{\text{p,min}}$ (m)	0.01
String-bridge connection		
Stiffness coefficient	K_{sm} (N/m)	$5 \cdot 10^6$
Nonlin. col. coeff.	α_{sm} (-)	1
Bridge location	χ_{sm} (m)	1.65*
Bridge-body collision		
Stiffness coefficient	K_{mp} (N/m)	$5 \cdot 10^8$
Nonlin. col. coeff.	α_{mp} (-)	1
Bridge location	$(x_{\text{mp}}, y_{\text{mp}})$ (m,m)	$(1.08, 0.135)$*
Other		
Offset	w_{off} (m)	$5 \cdot 10^{-6}$
Damp. finger coeff.	σ_{f} (kg·s^{-2})	$0 \leq \sigma_{\text{f}} \leq 1$
Output loc. string	u_{out} (m)	$L - \chi_{\text{sm}}$
Output loc. body	z_{out} (m, m)	$(x_{\text{mp}}, y_{\text{mp}})$

Table 1. List of parameter values used for the simulation. *These values have been taken from a real (recreated) tromba marina [23].

string according to the following equation

$$\chi^i_{\text{node}} = \frac{i \cdot \chi_{\text{sm}}}{n} \quad \text{for} \quad i = [1, \ldots, n-1], \tag{41}$$

where $n = \text{round}(\chi_{\text{sm}}/\chi_{\text{f}})$ is an integer closest to the ratio between the string length until the bridge location and the damping finger position. These lines are drawn to help the user place the damping finger at nodes along the string.

Secondly, the bridge section shows the displacement of the bridge w as a green circle, the state of the body at the collision location $z_{(l_{\text{sm}}, m_{\text{sm}})}$, both moving vertically according to their respective displacements and finally, a static grey horizontal line denoting the offset w_{off}, i.e., the resting position of the bridge.

Thirdly, the body section shows the state of the body z as a grid of rectangles changing (grey-scale) colour according to their displacement.

Finally, the control section contains three sliders that control the volume-levels of the string (s), bridge (m) and body

(p) respectively (for experimentation of volume ratios between the components) and a reset button to re-initialise the system.

4.7 Sensel Morph and Mapping

The Sensel is an expressive touch controller using ~20,000 pressure-sensitive sensors laid out in an hexagonal grid [12]. It retrieves x and y-positions and pressure at a rate of 150 Hz from which velocities and accelerations can be obtained.

The first finger registered by the Sensel is mapped to the bow: x-position is mapped to bow position χ_{b}, y-velocity to bow velocity v_{b} (y-position is shown in the GUI but does not influence the model directly) and pressure to bow force F_{b}. The second finger is mapped to the damping finger: x-position is mapped to finger location x_{f} and pressure to damping coefficient σ_{f}.

5. RESULTS AND DISCUSSION

Informal listening by the authors has confirmed that the sound has brass-like qualities and comparison with the recreated tromba marina showed that the sound exhibited similar qualities. Naturally, formal listening tests need to be conducted to verify this.

Disabling the graphics of the application, its CPU usage is 68.9% on a MacBook Pro with a 2.2 GHz Intel i7 processor, easily allowing it to work in real-time. As the heaviest part of the algorithm is the calculation of the body, the minimum grid spacing $h_{\mathrm{p,min}}$ could be set to a higher value to decrease the CPU usage. However, as mentioned, this will decrease the quality of the output sound.

Through using the application, the authors found some odd behaviour, where the bridge 'gets stuck' behind the plate, i.e., values for ψ_{mp} would be negative for a short period of time (one to several samples). The explicit technique used in this work allows for this to happen (and can be proven to still be stable in this case [18]), but it is 'unphysical' to have a negative potential as this implies a 'pulling' collision. As can be seen from Table 1, the nonlinear collision coefficients α_{sm} and α_{mp} are set to 1. When increasing these values, this behaviour would arise much more often, and even occur for a prolonged period of time (several seconds to indefinitely). This is also the reason why the reset button presented in Section 4.6 has been implemented. As mentioned in [18], oversampling increases the accuracy of the explicit collision method, and could be a solution to this issue. However, in order for the application to run in real time, this solution can not be afforded without decreasing the quality of the implementation, e.g. increasing $h_{\mathrm{p,min}}$. Further investigation will be necessary to solve this issue without oversampling.

Lastly, it has been found that when $|z^n_{(l,m)}| \lessapprox 10^{-306}$ (but non-zero) for any coordinate (l, m) (which happens when the body has not been collided with for a prolonged period of time), the CPU usage increases considerably. This could be explained by the fact that calculations with extremely small values are handled differently by the application. This is solved by implementing a limit to how small a value for $z^n_{(l,m)}$ can be. If the value of $z^n_{(l_{\mathrm{sm}},m_{\mathrm{sm}})}$ is lower than this limit, the total plate state is set to 0.

6. CONCLUSION AND FUTURE WORK

In this paper, a real-time implementation of a simulation of the tromba marina has been presented. The output sound has been found natural and brass-like by the authors and exhibited similar qualities when compared to a real (recreated) tromba marina.

Future work includes a comparison between the non-iterative methods used in this paper and iterative methods (such as and Newton-Raphson) both regarding algorithm speed and sound quality.

Lastly, for a more physical implementation of the damping finger, it would be good to model it as another mass colliding with the string rather than directly imposing damping onto the string state.

Acknowledgments

The authors would like to thank Peter Williams for his valuable feedback on our application.

This work is supported by NordForsk's Nordic University Hub Nordic Sound and Music Computing Network NordicSMC, project number 86892. Ducceschi's work was supported by an Early Career Fellowship from the Leverhulme Trust.

7. REFERENCES

[1] D. Munrow, *Instruments of the Middle Ages and Renaissance*. Oxford University Press, USA, 1976.

[2] C. Cadoz, "Synthèse sonore par simulation de mécanismes vibratoires," Ph.D. dissertation, Grenoble INP, 1979.

[3] J. O. Smith, "Physical modeling using digital waveguides," *Computer music journal*, vol. 16, no. 4, pp. 74–91, 1992.

[4] J. D. Morrison and J.-M. Adrien, "Mosaic: A framework for modal synthesis," *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, 1993.

[5] P. Ruiz, "A technique for simulating the vibrations of strings with a digital computer," Master's thesis, University of Illinois, 1969.

[6] L. Hiller and P. Ruiz, "Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I," *Journal of the Audio Engineering Society*, vol. 19, no. 6, pp. 462–470, 1971.

[7] ——, "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.

[8] 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.

[9] 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.

[10] S. Bilbao, *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics*. John Wiley & Sons, 2009.

[11] S. Bilbao, B. Hamilton, R. Harrison, and A. Torin, "Finite-difference schemes in musical acoustics: A tutorial." *Springer handbook of systematic musicology*, 2018.

[12] Sensel Inc. (2020) Sensel morph. [Online]. Available: https://sensel.com/

[13] S. Willemsen, N. Andersson, S. Serafin, and S. Bilbao, "Real-time control of large-scale modular physical models using the sensel morph," *Proc. of the 16th Sound and Music Computing (SMC) Conference*, pp. 275–280, 2019.

[14] S. Bilbao, A. Torin, and V. Chatziioannou, "Numerical modeling of collisions in musical instruments," *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2015.

[15] N. Lopes, T. Hélie, and A. Falaize, "Explicit second-order accurate method for the passive guaranteed simulation of port-hamiltonian systems," *Proc. 5th IFAC*, 2015.

[16] A. Falaize and T. Hélie, "Passive guaranteed simulation of analog audio circuits: A port-hamiltonian approach," *Applied Sciences*, vol. 6, pp. 273–273, 2016.

[17] A. Falaize, "Modélisation, simulation, génération de code et correction de systèmes multi-physiques audios: Approche par réseau de composants et formulation hamiltonienne à ports," Ph.D. dissertation, Université Pierre et Marie Curie, Paris, 2016.

[18] M. Ducceschi and S. Bilbao, "Non-iterative solvers for nonlinear problems: The case of collisions," *Proc. of the 22th Int. Conf. on Digital Audio Effects (DAFx-19)*, 2019.

[19] S. Bilbao, A. Torin, and V. Chatziioannou, "Numerical modeling of collisions in musical instruments," *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2014.

[20] S. Bilbao and M. Ducceschi, "Large-scale real-time modular physical modeling sound synthesis," *Proc. of the 22th Int. Conf. on Digital Audio Effects (DAFx-19)*, 2019.

[21] JUCE ROLI. (2020) JUCE. [Online]. Available: https://juce.com/

[22] S. Willemsen. (2020) Virtual tromba marina - sensel morph. [Online]. Available: https://www.youtube.com/watch?v=x72Xh-nUoVc

[23] A. Baldwin, T. Hammer, E. Peciulis, P. Williams, D. Overholt, and S. Serafin, "Tromba moderna: A digitally augmented medieval instrument," *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, vol. 16, pp. 14–19, 2016.

Paper F

Resurrecting the Tromba Marina: A Bowed Virtual Reality Instrument using Haptic Feedback and Accurate Physical Modelling

RESURRECTING THE TROMBA MARINA: A BOWED VIRTUAL REALITY INSTRUMENT USING HAPTIC FEEDBACK AND ACCURATE PHYSICAL MODELLING

Silvin Willemsen, Razvan Paisa and Stefania Serafin
Multisensory Experience Lab, CREATE
Aalborg University Copenhagen
{sil, rpa, sts}@create.aau.dk

ABSTRACT

This paper proposes a multisensory simulation of a tromba marina – a bowed string instrument in virtual reality. The auditory feedback is generated by an accurate physical model, the haptic feedback is provided by the PHANTOM Omni, and the visual feedback is rendered through an Oculus Rift CV1 head-mounted display (HMD). Moreover, a user study exploring the experience of interacting with a virtual bowed string instrument is presented, as well as evaluating the playability of the system. The study comprises of both qualitative (observations, think aloud and interviews) and quantitative (survey) data collection methods. The results indicate that the implementation was successful, offering participants realistic feedback, as well as a satisfactory multisensory experience, allowing them to use the system as a musical instrument.

1. INTRODUCTION

The tromba marina is a bowed monochord from medieval Europe [1] (see Figure 1). The string rests on a loose bridge that rattles against the body. This rattling mechanism creates a sound with brass- or trumpet-like qualities. Unlike other bowed string instruments, different frequencies are created by slightly damping the string with a finger of the non-bowing hand as opposed to pressing the string fully against the neck. This interaction at different locations along the string triggers the different harmonics of the open string. Furthermore, the tromba marina is bowed closer to the nut, and the finger determining the frequency is closer to the bridge (below the bow). As the tromba marina is a rare instrument which can be merely found in museums, very few have the opportunity to play it and discover its interesting timbral possibilities. We wish to recreate the feeling of playing this instrument by using physics based multisensory simulations [2].

In the context of musical applications, physics based multisensory simulations have shown some interest in the sound and music computing community. As stated in [3], the combination of haptics and audio visual content has its own specific challenges worth investigating. Sile O'Modhrain is one of the pioneers that noticed the tight connection between auditory and haptic feedback and investigated how haptic feedback can improve the playability of virtual instruments [4]. At the same time, Charles Nichols developed the vBow, a haptic human computer interface for bowing [5]. For several years, researchers from ACROE in Grenoble have developed multisensory instruments based on the mass-spring-system paradigm, with custom-made bowing interfaces [6, 7]. Such multisensory simulations have recently been made open source [8]. Haptic feedback has also been combined with digital waveguide models for simulating bowed string interactions [9].

Figure 1. A tromba marina owned by *Nationalmuseet* in Copenhagen, Denmark.

Simulating the feeling of string-instrument vibrations is particularly important since it has been shown how vibrations' level can be strongly perceived [10]. We use democratized VR technologies controlled by a commercial device called the PHANTOM Omni (or simply Omni) by SenseAble Technologies (now 3D Systems) [11]. The Omni is a six-degrees-of-freedom system providing the tracking and haptic feedback (up to 3.3 N) in our application. Using the same device, Avanzini and Crosato tested the influence of haptic and auditory cues on perception of material stiffness [12]. Auditory stimuli were obtained using a physically-based audio model of impact, in which the colliding objects are described as modal resonators that interact through a non-linear impact force [13]. Auditory stiffness was varied while haptic stiffness was kept constant. Results show a significant interaction between auditory stiffness and haptic stiffness, the first affecting the perception of the second. Passalenti et. al's also used the Omni to simulate the act of plucking a virtual guitar string [14–16].

The goal of this project is to explore the experience of interacting with virtual bowed instrument by using physics based simulations and haptic feedback, together with a visual virtual reality (VR) experience. This effectively makes the implementation a virtual reality musical instrument (VRMI)

[17]. The tromba marina is used solely as inspiration because it affords itself to being a solid starting point by having only one string. Besides that, the rarity of the instrument ensures that the participants do not have prior experience playing a tromba marina, nullifying possible comparisons between a real instrument and the virtual one. At no point the system was evaluated as an alternative to the real tromba marina. The system (and its evaluation) is targeted towards musicians in order to avoid discouragement frequently encountered when non-musicians interact with musical instruments. It is assumed that musicians acknowledge that mastering any instrument require extended study, therefore it is expected that they will not evaluate this system exclusively based on its difficulty to play.

We start by describing the implementation of the system, both from the hardware and software perspective in Section 2, followed by presenting a study that evaluates the setup in Section 3. Section 4 shows the results of the evaluation and Section 5 discusses these. Finally, concluding remarks appear in 6.

2. IMPLEMENTATION

The virtual tromba marina consists of three main components: auditory, visual and haptic feedback, all of which will be elaborated on in this section. For visuals, the Oculus Rift CV1 setup was used [18]. The setup consists of a head-mounted display (HMD) and a pair of of wireless controllers that provide tracking information and user input through several buttons and a joystick. A diagram showing the full setup of the system can be found in Figure 2. The controls, their mapping to the system and the final setup of the system will also be presented. A video showing the implementation can be found in [19].

2.1 Auditory Feedback

The audio is generated by a physical model of the tromba marina presented in a companion paper [20]. Some parameters of the model are exposed and can be controlled by the user. These are the velocity, force and position of the bow and the position of the finger inducing the harmonics. The algorithm will not be discussed in detail here, but the mapping to the various parameters of the model will be described in Section 2.4.

2.2 Visual Feedback

The application was built using the cross-platform game engine Unity3D (or simply Unity) [21] which can be used to build VR applications. Even though the visual feedback is not the focus of the implementation and eventual evaluation, it was used to guide the users' movements and give them a sense of where the virtual instrument was located. Figure 3 shows a screenshot of the view from the HMD, depicting the virtual instrument, the bow and the damping finger indicator. A 3D model of the tromba marina was made inspired by a real-life instrument (presented in [22]) available to the authors. The overall environment resembled a medieval room, providing context to tromba marina's historical nature.

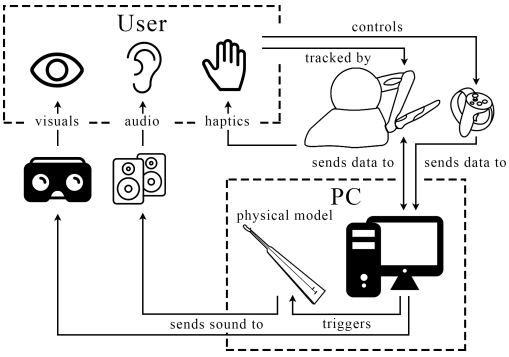

Figure 2. Diagram showing the system layout of the application. The user interacts with the system using the Omni – which in turn provides haptic feedback – and the Oculus Touch controller. These trigger the physical model of the tromba marina. Auditory feedback then comes from speakers and visual feedback from the Oculus Rift headset. A detailed explanation can be found in Section 2.5.

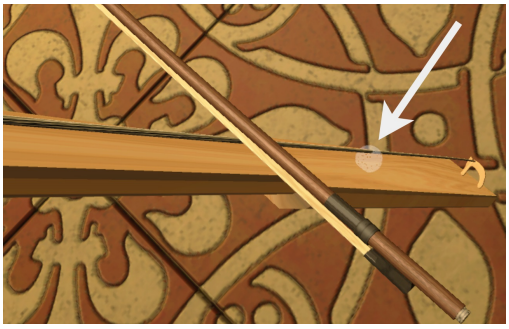

Figure 3. The view from the head-mounted display (HMD). The damping finger is highlighted and shown as a transparent white sphere.

2.3 Haptic Feedback

The PHANTOM Omni (or simply Omni) is a six-degrees-of-freedom tracking and haptic system developed by Sens-Able Technologies (see Figure 4). The device has a pen-shaped arm that a user interacts with.

The raw data provided by the Omni are 1) the absolute position of pivot point B2 (three degrees of freedom), 2) the rotation (three degrees of freedom), and 3) the pressure (touching depth). The latter is calculated from the absolute euclidean distance between the virtual collision point of the object (in our case the bow) and the virtual position of the pen.

The axes are labelled as follows in relation to the virtual tromba marina (also see global coordinate system in Figure 5): x-axis (width): horizontally across the soundboard (the common interaction direction), y-axis (height): floor to ceiling, and z-axis (depth): perpendicular to the soundboard. The orientation of the Omni with respect to

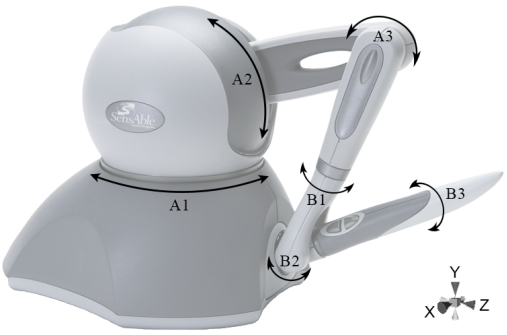

Figure 4. The PHANTOM Omni has six axes of rotation, three of which provide force feedback (A1-3), and three only tracking position (B1-3). Together, these axes provide six degrees of freedom: x, y and z positions of B2 (according to the shown coordinate system) and rotations of the pen.

the aforementioned axis can be seen from the coordinate system in Figure 4.

The fact that pivot points B1-3 do not provide force feedback gives rise to an issue in our application. The virtual bow's frog (where it is held by the player) has been placed at the pivot point B2, whereas the interaction between the virtual bow and string happens at an offset as seen in Figure 5. To solve this issue, we created a separate game object with which the bow (pivot point B2 to be exact) will interact with in the virtual world Figure 5. This '(hidden) collision block' lives in a local coordinate system and its x and y-position exactly follow that of the Omni-pen. The y-rotation will change the rotation of the local coordinate system and uses the virtual string as the center point. If, for any reason, the bow ends up behind the string, the collision block will be offset to the left along the (local) x-axis so that no collision occurs when trying to return the bow to the normal playing area.

Through a list of pseudophysical parameters, the collision forces computed by Unity's physics engine are mapped to the haptic feedback produced by the Omni. Through empirical testing, the following pseudophysical parameters have been found: *Stiffness*: 0.003, *Damping*: 0.0071, *Static Friction*: 0, *Dynamic Friction*: 0.109, and *Pop-through*: 0. For more information, please refer to [23].

Throughout implementation, it was considered to actuate the Omni's pen with the output of the physical model used for the auditory feedback, in order to replicate the stick-slip interaction encountered in a real bowing scenario. This was deemed unnecessary, as the Omni's internal gearing systems provide a similar, though uncorrelated, haptic feedback, which satisfied the authors.

2.4 Controls and Mapping

As most people are right-handed, it was chosen to also have the bow in the right hand in the application. The (now-local) x-velocity of the Omni is mapped to the bow velocity, pressure to bow force and y-position (including

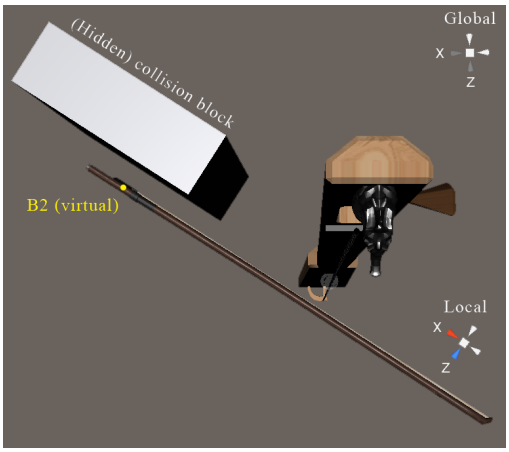

Figure 5. Top-down view of the global and local coordinate system (x-z–plane). The rotation of the local coordinate system around the (global) y-axis is determined by the y-rotation of the bow. The (normally hidden) collision block lives in the local coordinate system. Its (local) x and y-position follows the (local) x and y-position of B2.

rotation around the local z-axis) to bow position. The left hand is used to control the pitch by changing the position of the damping finger along the string. This position is defined as

$$x_{\mathrm{f}} = L \cdot n^{-1}, \tag{1}$$

where L is the length of the string and $n \in [2, 8]$. If n is an integer, it is the number of the harmonic we want to induce. The lowest harmonic has been set at half the string length $L/2$, meaning that the string is never completely open. The highest harmonic (8 in this case) has been chosen to be the one that can still be (comfortably) reached. The location of the damping finger x_{f} is controlled using the 'X' and 'Y' buttons and the joystick on the left Oculus Touch controller. The buttons are used for "discrete harmonic" control of the damping finger, i.e. integer values of n in Equation (1), where 'Y' increases n and 'X' decreases it. The joystick allows for fine pitch control, i.e., decimal values of n, and moves the damping finger up and down the string. The latter could potentially create pitch glides in the output sound of the application, but make it harder to 'hit' a perfect harmonic according to Equation (1). If a button is pressed while the current finger position is between two discrete points, the position will move to the next or previous discrete position, depending on the button pressed.

2.5 Physical Setup

The physical setup is shown in Figure 6. The Omni is mounted on a stand at ~125 cm to match the approximate bowing height of the real instrument. As can be seen in Figure 6, the right Oculus Touch controller is mounted right underneath the Omni. This is used to align the physical setup with the virtual tromba marina, both in the x-

Figure 6. User interacting with the physical setup. The Omni is mounted on a ~125cm stand together with the right Oculus Touch controller used for location and tilting information.

z–plane but also the height of the bow in the application. After the scene is initialised the controller is used for a tilting interaction so that the instrument can rest on the user's body, as is done with the real instrument. The aforementioned alignment came with a drawback – as the center of the x-axis range of the Omni was aligned with the tromba marina and B2 was aligned with one end of the bow, only half of the range of the Omni could be used for bowing.

The setup shown in Figure 2 is implemented as follows: the user controls the application using the Omni (for tracking) and the left Oculus Touch controller which sends data to the computer running the application. The Omni produces haptic feedback based on Unity's physics engine calculating the interaction force between the '(hidden) collision block' and the virtual bow as shown in Figure 5. This data simultaneously triggers the physical model which sends its output to a pair of speakers. The user wears a HMD that gives visual information about the location of the tromba marina (and medieval scene). The user's position in the VR environment is controlled by the HMD, but this dataflow is not visualised in the diagram. Lastly, the right Oculus Touch controller is attached to the stand the Omni is attached to, and sends position and tilting data to the application.

3. EVALUATION

The goal of the study was to (1) evaluate the general experience of bowing in a VR environment using haptic feedback and accurate physical modelling and (2) to evaluate the playability of a VR monochord instrument. This was done by exploring the quality of the software, the acoustic model, the interface and the mapping, as proposed by [24] and implemented previously in a similar study [25]. To meet this aim, an investigative study was performed through which feedback on the virtual instrument was collected. In order to ensure a high level of validity and reliability, a triangulation of methods has been used: think aloud protocol [26] throughout the interaction, observation and post-study self report through a modified Usability Metric for User Experience survey [27]. The study concluded with an semi-structured interview based on the observed actions, noted comments and questions loosely revolving around *goals, operators, methods and selection* method [28].

3.1 Participants

A total of 14 people (12 male, 2 female), 23-48 years old (M=29.5, SD=7.65) participated in the study. All participants were students or staff at Aalborg University Copenhagen. The selection of participants was based on the single criterion that one had to have experience playing a musical instrument. Over 70% of the participants have been playing an instrument for more than 5 years, guitar being the most common occurrence (25%). There was only one participant experienced in playing bowed instruments (violin). The same participant mentioned playing the tromba marina briefly before, but the majority of the other participants had never heard (of) it. All but one participant have had tried VR experiences before joining the study.

3.2 Procedure and Task

The experiment started with the participant reading an introduction about the experiment and completing a questionnaire covering several demographic questions (age, gender, musical experience, familiarity with the tromba marina and VR experience). They were then introduced to the setup and task, and controls were explained. The participants were informed that the study is exploring the experience of bowing in a VR environment. It was emphasised that the most important part of the experiment was for the participant to talk aloud with the phrase: "anything positive, negative, basically anything that comes to mind, please speak out loud". Furthermore, the user was instructed to bow above the damping finger (visualised as a white sphere) at all times, as this is also the interaction with the real instrument.

The interaction part was divided into two phases. Firstly, the participants were asked to freely explore the instrument on their own. Then, when they felt they are ready to move on, an audio recording made by the authors using the application was played, showcasing the system's capabilities, aiming to inspire the second phase of free exploration. It was stressed that the participants did not have to recreate what they heard, but to merely use it as inspiration. The experiment concluded with participants completing a questionnaire covering usability and playability of the system. Finally, a semi-structured interview was held which lasted 5 minutes on average.

Throughout the interaction phase, the participants' actions were observed and noted by the authors, and their comments written down. Most participants were encouraged again to think aloud during their exploration. The full experiment lasted ~30 minutes for all participants.

3.3 Measurements

Because the goal of the study was to investigate the overall experience of bowing in VR, as well as evaluate the playability of the instrument, self-reporting measurements were used in combination with the observations, interview and *think aloud* notations. Specifically, after exposed to the instrument, the participants were asked to fill out a questionnaire containing 20 items related to the experience of interacting with the VRMI. The items can be broadly segmented into four categories: overall experience, haptic feedback, auditory feedback and visual feedback. Table 1 presents the questions.

Questionnaire items:
Overall experience:
(1) It was easy to understand how to play the instrument.
(2) I felt the instrument was hard to play.
(3) I felt the instrument was expressive.
(4) The instrument's capabilities did not match my expectations.
(5) I felt I could easily achieve my goals.
(6) I made many errors playing the instrument.
(7) I am satisfied with the instrument.
(8) I felt the instrument was boring.
(9) Interacting with the instrument was frustrating.
Haptic feedback:
(10) I felt the haptic feedback was realistic.
(11) I felt the haptic feedback was too strong.
(12) I felt the haptic feedback was natural.
Auditory feedback:
(13) I felt I was in control of the sound.
(14) I felt the audio was matching my actions.
(15) I felt the sound was matching the haptic feedback.
(16) I felt the sound was matching the visuals.
(17) I felt the sound was static.
Visual feedback:
(18) I felt the visual feedback was helping me play.
(19) I felt the visuals were confusing.
(20) I felt the visuals were matching my actions.

Table 1. The questionnaire items and corresponding anchors of the 5 point (1 – 5) rating scales (Strongly disagree – Strongly agree).

4. RESULTS

This section presents the results obtained from the self-reported measure regarding the participants' experience as well as the qualitative findings from interview, observations and think aloud.

4.1 Quantitative Data

The data obtained for the questionnaire items was treated as ordinal and analysed in terms of central tendency (medians and mode), interquartile ranges, minimum and maximum ratings. Figure 7 visualises the collected data. The mode was considered only when different from the median, specifically question 8, 13 and 14. It is worth noting that most of the items show a skewed normal distribution.

Question 1-9 paint a picture of how the instrument was perceived by the users. Questions 1, 2, 4, 5, 6 and 9 cover the perceived difficulty of using the system as a musical instrument. The answers to these questions show that even though participants generally found the instrument easy to understand, they had difficulty playing it and reaching their goals. Questions 3, 7 and 8 cover their general opinion about the instrument. Participants generally felt satisfied and not bored with the instrument. Questions 10-12 cover exclusively the impressions about haptic feedback. It can be seen that most participants found the haptic feedback to be realistic and generally natural and the force to be not too strong. The questions 13-17 approach the auditory aspect of the instrument, focusing on its perceived characteristics. As can be seen from questions 13, 14 and 17, the participants felt a high level of command over the sound, and were satisfied with mapping between the haptic and auditory feedback. The same thing can be said about the visual mapping, as indicated by question 16. Items 18-20 investigate the perceived visual quality. It can be seen that the visuals helped the participants play and were implemented well, i.e., not confusing and matching their actions.

4.2 Qualitative Data

In order to present an accurate representation of the findings, this section will be split into two categories: actions – covering the observed activities during the interaction phase, and oral feedback – presenting the findings from the think aloud protocol and interviews.

4.2.1 Observed Actions

Since there were no tasks given to the participants, all actions were noted and analysed. That said, most users performed similar actions in their interaction phase. All participants experimented with bowing at different heights, but only a few of them tried to explore bowing heights for all discrete pitches. Most of them were satisfied with trying different heights on whatever pitch they found themselves at that time. In a similar fashion, all participants experimented with playing different pitches, both using the discrete buttons as well as the joystick. It is worth mentioning that many users tried to investigate the limits of the pitches they could play. Higher pitches usually resulted in little or no sound which was commented on by most. This behaviour is true to a real tromba marina, where higher harmonics are harder to excite than lower ones. The majority tried to perform some form of glissando, as well as bowing with different velocities, usually commenting on the findings. Due to the non-intrusive nature of observation, it was impossible to notice the pressure applied with the bow, but some participants explicitly mentioned that they tried to experiment with different forces. This was especially true in the second phase of interaction, when they experimented with a higher dynamic range of sounds. Another common occurrence was the attempt to play some sort of melody or riff. Simple melodies like *Mary had a little lamb*, or *Twinkle twinkle little star* were attempted, with various degrees of success. One participant tried to play a Mozart segment. The last commonality was found in the attempt to perform a sustained tone, with a constant bowing speed and a back-and-forth motion.

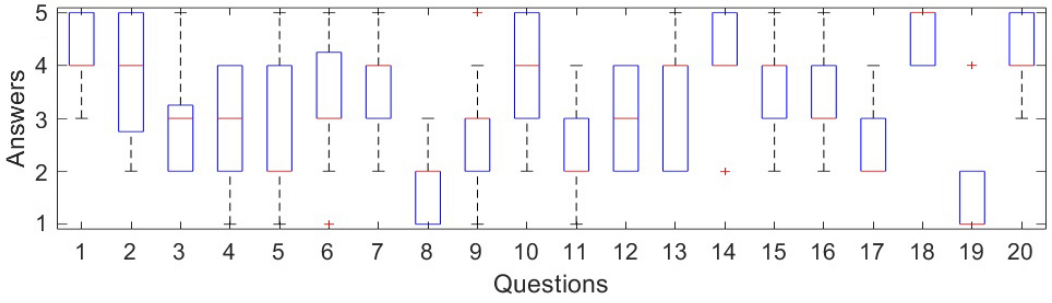

Figure 7. Boxplots visualizing the results related to the 20 questionnaire items (shown in Table 1) in terms of medians (red lines), interquartile ranges (blue rectangles), minimum and maximum ratings (dashed lines), and outliers (red crosses). The y-axis maps "Strongly disagree – Strongly agree" to a 1 – 5 interval.

When it comes to seldom or individual actions, a great variance in experimentation was observed. Participants tried to hit the string with the bow, rotate the bow upwards to the point of it being parallel to the string, move the bow in an up-down (y-axis) motion, bow on the damping finger indicator and underneath it or play some form of vibrato or staccato. No one tried to tilt the stand supporting the Omni.

4.2.2 Oral Feedback

Generally the overall impression of the instrument was positive, described with words like: *cool, fun, interesting, weird*, as well as *hard* or *difficult to play*. One participant's answer encapsulates this very well by saying: "I got to express my ideas, but not perfect them".

Just as described in the previous section, there was a general consensus on several reported characteristics. All participants that attempted to play the highest harmonic said it is hard to play, and that it felt frustrating. At the other end, several participants expressed their preference towards lower pitches, where some said that they prefer the sound produced when bowing under the damping finger (essentially playing the lower-pitched open string). Besides that, many reported that is was hard to maintain a sustained tone, regardless of the pitch. Another sound-related report was the inability to re-create the buzzing sound heard in the recording; one of the participants familiar with the tromba marina's mechanism even mentioned specifically that he "couldn't get the bridge to rattle". When it comes to the pitch selection interface, the reports are very polarised between the joystick and the buttons. On one hand some describe the buttons as being more, *fun, musical, melodic, useful* or *easier*, while describing the joystick as *useless, unrealistic, too hard* or *meaningless*. On the other hand some participants clearly preferred the joystick describing it as *natural, intuitive, interesting, expressive* or *humane*, but everyone mentioned that the sensitivity of the joystick is too high, making it hard to land on the desired pitches. Due to the incremental nature of the damping fingers' position, it was impossible to skip over notes, a fact that was mentioned in different forms by several participants. Some noted that the control of the damping finger ('Y' for up the string and 'X' for down) should have been inverted. Furthermore, some would have liked a more physical interaction for the damping finger, such as moving the controller up and down rather than using buttons.

When it comes to the haptic feedback, the majority was satisfied with it, mentioning that "it feels nice", "it feels good", "is great", "impressive - it felt natural", or "it feels real", while one participant found it to be "wild and a bit too powerful". A special case related to the haptic feedback was the bounce obtained by hitting the virtual string with the bow. Most subjects found it pleasing and were intrigued by its realistic feel, but the violin player repeatedly mentioned that it is "unrealistic and way to powerful". One participant explicitly mentioned that the haptic feedback matches the auditory one, and his expectations.

Several participants noticed that the bow could rotate along its axis and asked whether it made a sonic difference or not, to which they were answered negatively. Besides that, there were very few comments regarding the visual aspect of the system, but most of these were positive. One participant mentioned that sometimes there's a gap between the string and the bow, and that it would be nice to observe one's hands. No one mentioned anything related to the visual indication or the damping finger seen in Figure 3.

The overall interaction was described offering a high degree of freedom on the bowing hand, but the pitch selecting hand was either not mentioned, or described as *disconnected* several times. Some agreed that the instrument is hard to play, mentioning that it is *frustrating*. However, most people estimated that they can perform better after practising more.

5. DISCUSSION

In this section, both the evaluation procedure itself and the results from Section 4 will be discussed.

5.1 Procedure

It is acknowledged that the cognitive load of speech and playing an instrument are overlapping [29], therefore the *think aloud protocol* might have not generated in the most

abundant data possible. Most participants alternated between playing and speaking. This resulted in occasionally long breaks in either activities, and required participants to be encouraged to think aloud. Retrospectively, a structured activity schedule allocating time for playing and feedback could have been more productive. Similarly, using self report through Likert scales require large sample sizes to achieve a high level of accuracy [29]. Therefore the interpretation of results rooted into the qualitative data, and then validated using the quantitative data.

Furthermore, as the data we obtained was purely through non-intrusive methods, it would have been useful to log the raw data provided by the Omni (such as the bowing pressure). This could then have been analysed to obtain a better understanding of the user's feedback.

Lastly, the audio did not fully match the sounds that were possible to create with the application. As the recording was quite distorted, the volume of the audio plugin was turned down during the test, but the recording was not remade. This will be elaborated on below.

5.2 User Feedback

The generally positive oral feedback about the overall experience is backed up by the quantitative data which showed that participants were satisfied and not bored with the instrument. They attempted to perform fundamental tasks as producing a sustained tone or playing simple melodies with various degrees of success, and when exposed to the example recording, some tried to recreate the sounds heard from the audio clip. Several participants mentioned that it was difficult to achieve this particular goal, a problem that finds its explanation in the difference in volume between the recording and the experiment scenario as mentioned above. This would be a point of improvement for future testing, as it could have impacted the answers for question 5 – the lowest scoring question regarding the overall experience. Another reason for this question's answers could be linked to the inability to play the higher notes, or the limited pitch range, as presented in Section 4, but these characteristics are inherited from the physical characteristics of the real instrument, so could be expected.

Interestingly, many participants believed that it was easy to understand how to play the instrument, but that they could become better after some more practice. This indicates that the setup has a low "entry-level", with an envisioned high virtuosity ceiling. This is believed something desirable when creating computer based instruments [30]. Even though the implementation was inspired by a real instrument with a possibly different learning curve, as mentioned, it was not our goal to recreate it.

The haptic feedback was considered positive and generally having an appropriate level of resistance to movements. The answers to questions 10 (realistic haptic feedback) and 12 (natural haptic feedback) correlate positively and question 15 (sound matching the haptic feedback) was also answered positively, giving a strong indication that the participants considered the haptic feedback real and according to their expectations. It can be understood that the realism of the haptic feedback is estimated considering the multisensory experience and this result reassures that bowing in VR with our setup is possible.

Furthermore, many participants noticed that they could 'bounce' the bow onto the string. Even though this behaviour was a byproduct of the implementation, participants generally liked this interaction and found it to be realistic and exciting.

Many users noted that the full range of the bow could not be used. As mentioned in Section 2.5, the virtual tromba marina was aligned with the physical position of the Omni and as the bow is held at one end, about half of the range could not be used for bowing. This was a commonly reported issue, and it could have impacted the answers for questions 4, 5, and 9. The reason for aligning the physical setup with the virtual tromba marina was the tilting interaction, so that if people wanted to interact with the entire instrument, they would be able to grab the physical setup. As none of the participants used this, we could discard the aforementioned alignment to be able to account for the entire range of the bow.

The polarisation of the participants' opinion on the pitch control – joystick versus buttons – was backed up by the answers individuals gave on question 9 (interaction was frustrating). It could be argued that users preferring the joystick over the buttons had a harder time interacting with the instrument than the people preferring the buttons. As mentioned, all participants who mentioned the joystick interaction said it was too fast, explaining the above.

6. CONCLUSIONS

This paper presents a virtual reality implementation of the tromba marina and its evaluation. Our goal was to evaluate the general experience of bowing in VR and to evaluate the playability of our implementation. The results show that the implementation was successful with participants finding the haptic feedback realistic and the general experience enjoyable and interesting on one hand, and difficult and frequently frustrating on the other hand. Nevertheless, all sensory modalities we focused on (auditory, haptic and visual) seemed to reinforce each other, inspiring participants to attempt to play melodies with the instrument. This was considered to be an important achievement. Improvements on our application include the pitch control, which should either be more physical, i.e., moving the pitch hand physically up and down the virtual string, or simply slower continuous control. Besides that, a better physical setup, allowing the users to utilise the entire bow is desired. The findings of this paper prove that it is possible to create a satisfactory bowed VRMI using off-the-shelf hardware and accurate physical modelling.

Acknowledgments

We would like to thank Peter Williams for his valuable insights on playing the tromba marina, feedback on our application and providing access to his instrument replica.

This work is supported by NordForsk's Nordic University Hub Nordic Sound and Music Computing Network NordicSMC, project number 86892.

7. REFERENCES

[1] The Editors of Encyclopædia Britannica, *Encyclopædia Britannica*. Encyclopædia Britannica, Inc., 2020.

[2] D. K. Pai, "Multisensory interaction: Real and virtual," *Robotics Research. The Eleventh International Symposium*, pp. 489–498, 2005.

[3] F. Danieau, A. Lécuyer, P. Guillotel, J. Fleureau, N. Mollet, and M. Christie, "Enhancing audiovisual experience with haptic feedback: a survey on hav," *IEEE transactions on haptics*, vol. 6, no. 2, pp. 193–205, 2012.

[4] M. S. O'Modhrain, "Playing by feel: incorporating haptic feedback into computer-based musical instruments," Ph.D. dissertation, Stanford University, 2001.

[5] C. Nichols, "The vbow: development of a virtual violin bow haptic human-computer interface," pp. 1–4, 2002.

[6] J.-L. Florens and C. Cadoz, "Modèles et simulation en temps réel de corde frottée," *Le Journal de Physique Colloques*, vol. 51, no. C2, pp. C2–873–C2–876, 1990.

[7] A. Luciani, J.-L. Florens, and N. Castagné, "From action to sound: a challenging perspective for haptics," pp. 592–595, 2005.

[8] J. Villeneuve and J. Leonard, "Mass-interaction physical models for sound and multi-sensory creation: Starting anew," *Proc. Int. Conf. Sound and Music Computing*, 2019.

[9] S. Sinclair, G. P. Scavone, and M. M. Wanderley, "Audio-haptic interaction with the digital waveguide bowed string," *ICMC*, 2009.

[10] A. Askenfelt and E. V. Jansson, "On vibration sensation and finger touch in stringed instrument playing," *Music Perception: An Interdisciplinary Journal*, vol. 9, no. 3, pp. 311–349, 1992.

[11] 3D Systems, Inc, "3D Systems Touch Haptic Device," accessed February 12, 2020, available at https://www.3dsystems.com/haptics-devices/touch.

[12] F. Avanzini and P. Crosato, "Haptic-auditory rendering and perception of contact stiffness," *Lecture Notes in Computer Science*, vol. 4129, pp. 24–35, 2006.

[13] F. Avanzini and D. Rocchesso, "Physical modeling of impacts: theory and experiments on contact time and spectral centroid," *Proc. Int. Conf. Sound and Music Computing*, pp. 287–293, 2004.

[14] A. Passalenti, R. Paisa, N. Nilsson, N. Andersson, F. Fontana, R. Nordahl, and S. Serafin, "No strings attached: Force and vibrotactile feedback in a virtual guitar simulation," *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1116–1117, 2019.

[15] ——, "No strings attached: Force and vibrotactile feedback in a guitar simulation," *Proc. Int. Conf. Sound and Music Computing*, 2019.

[16] F. Fontana, R. Paisa, R. Ranon, and S. Serafin, "Multisensory plucked instrument modeling in Unity3D: From keytar to accurate string prototyping," *Applied Sciences*, vol. 10, 2020.

[17] S. Serafin, C. Erkut, J. Kojs, N. Nilsson, , and R. Nordahl, "Virtual reality musical instruments: State of the art, design principles, and future directions," *Computer Music Journal*, 2016. [Online]. Available: http://www.mitpressjournals.org/doi/pdfplus/10.1162/COMJ_a_00372

[18] Facebook Technologies, LLC, "Oculus Rift: VR Headset for VR-ready PCs — Oculus," accessed March 6, 2020, available at https://www.oculus.com/rift/.

[19] Razvan P, "Resurrecting the Tromba Marina," accessed March 6, 2020, available at https://www.youtube.com/watch?v=SlHqvaaPCyU.

[20] S. Willemsen, S. Serafin, S. Bilbao, and M. Ducceschi, "Real-time implementation of a physical model of the tromba marina," *submitted to the 17th Sound and Music Computing (SMC) Conference*, 2020.

[21] Unity Technologies, "Unity Real-Time Development Platform — 3D, 2D VR & AR Visualizations," accessed February 10, 2020, available at https://unity.com/.

[22] A. Baldwin, T. Hammer, E. Peciulis, P. Williams, D. Overholt, and S. Serafin, "Tromba moderna: A digitally augmented medieval instrument," *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, vol. 16, pp. 14–19, 2016.

[23] 3D Systems Inc., "Openhaptics unity plugin user guide (toolkit version 3.5.0)," 2018, available at http://s3.amazonaws.com/dl.3dsystems.com/binaries/Sensable/OH/3.5/OpenHaptics_Toolkit_ProgrammersGuide.pdf.

[24] J. Barbosa, J. Malloch, M. Wanderley, and S. Huot, "What does "evaluation" mean for the NIME community?" *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2015.

[25] D. Young and S. Serafin, "Playability evaluation of a virtual bowed string instrument," *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pp. 104–108, 2003.

[26] M. Someren, Y. Barnard, and J. Sandberg, *The Think Aloud Method - A Practical Guide to Modelling Cognitive Processes*. London: Academic Press, 1994.

[27] K. Finstad, "The usability metric for user experience," *Interacting with Computers*, vol. 22, pp. 323–327, 2010.

[28] S. K. Card, A. Newell, and T. P. Moran, *The Psychology of Human-Computer Interaction*. New Jersey: Lawrence Erlbaum Associates Publishers, 1983.

[29] D. Stowell, A. Robertson, N. Bryan-Kinns, and M. Plumbley, "Evaluation of live human–computer music-making: Quantitative and qualitative approaches," *International Journal of Human-Computer Studies*, vol. 67, pp. 960–975, 2009.

[30] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer Music Journal*, 2002.

Paper G

DigiDrum: A Haptic-based Virtual Reality Musical Instrument and a Case Study

DIGIDRUM - A HAPTIC-BASED VIRTUAL REALITY MUSICAL INSTRUMENT AND A CASE STUDY

Silvin Willemsen
Multisensory Experience Lab
CREATE, Aalborg University
Copenhagen, Denmark
sil@create.aau.dk

Anca-Simona Horvath
Research Laboratory for Art and Technology
KOM, Aalborg University
Aalborg, Denmark
ancah@hum.aau.dk

Mauro Nascimben
Augmented Cognition Lab
CREATE, Aalborg University
Copenhagen, Denmark
mana@create.aau.dk

ABSTRACT

This paper presents DigiDrum – a novel virtual reality musical instrument (VRMI) which consists of a physical drum augmented by virtual reality (VR) to produce enhanced auditory and haptic feedback. The physical drum membrane is driven by a simulated membrane of which the parameters can be changed on the fly. The design and implementation of the instrument setup are detailed together with the preliminary results of a user study which investigates users' haptic perception of the material stiffness of the drum membrane. The study tests whether the tension in the membrane simulation and the sound damping (how fast the sound dies out) changes users' perception of drum membrane stiffness. Preliminary results show that higher values for both tension and damping give the illusion of higher material stiffness in the drum membrane, where the damping appears to be the more important factor. The goal and contribution of this work is twofold: on the one hand it introduces a musical instrument which allows for enhanced musical expression possibilities through VR. On the other hand, it presents an early investigation on how haptics influence users' interaction in VRMIs by presenting a preliminary study.

1. INTRODUCTION

Virtual Reality (VR) is described as an immersive environment provided by technology and experienced through sensory stimuli [1]. Different types of technologies are available for creating VR experiences, and head-mounted displays (HMDs) are among the most popular. VR has been used as a platform for the creation of perceptual illusions, and much research has gone into producing realistic or otherwise compelling visual and auditory experiences. By comparison, the sense of touch has been neglected in spite of its obvious potential to increase a sense of presence in a simulated world [1].

Virtual musical instruments (VMIs) are defined as software simulations or extensions of existing musical instruments with a focus on sonic emulation. Virtual reality musical instruments (VRMIs), are those which also include a simulated visual component [2].

The design and evaluation of DigiDrum – a novel VRMI where a physical darbuka (a djembe-like drum) is enhanced by VR is presented. The user wears a HMD which puts them in a recording studio where a virtual drum is aligned with the physical drum, so that both drums can be played at the same time. Interaction with the (physical + virtual) drum triggers a virtually simulated sound of a drum membrane. This sound is sent to the user through sound-isolating headphones for auditory feedback, and a vibration motor (haptuator) attached to the inside of the physical drum's membrane creating a vibrotactile response in the physical drum – similar to the haptic response a real membrane would produce. As the drum's sound is being simulated, its properties can be changed on the fly, something which is impossible to do in the physical world.

An initial user study was conducted on DigiDrum with a twofold goal. On the one hand, in order to study how users interact with the installation and use this feedback to improve the drum, and on the other hand, for trying to understand whether there is a correlation between material stiffness perception and the way users interact with the drum. More specifically, the study investigated which parameters influence the perception of the material stiffness of the drum membrane. Different combinations of values for: (1) tension in the virtual membrane and (2) damping, or how quickly the sound dies out were used. The initial hypothesis was that higher values for both tension and damping would influence the perception of stiffness positively. In other words, higher tension and higher damping (sound dying out faster) will result in users perceiving the drum membrane as being more stiff. It was suspected that tension would be the most important parameter in the perception of stiffness. In the test, the auditory and haptic cues were linked, or matching.

The research question which guides this work is:

Can a user's perception of material stiffness in an enhanced drum membrane change by using auditory and haptic cues?

The ultimate goal of the paper is to (1) present an installation which helps to enhance musical expression possibilities through a novel VRMI, and (2) investigate users' interaction with a VRMI focused not only on the visual and auditory experience, but also on haptics.

The paper is structured as follows: Section 2 presents a selection of related work. Section 3 is an introduction to haptic perception. Section 4 describes the design criteria used in for DigiDrum. In Section 5 we describe the system overview and Section 6 details the implementation of the visual virtual environment. In Section 7, the physical model sound algorithm is described. In Section 8 a user study looking at the interaction with the setup is presented and preliminary results are shown and discussed in Section 9. Conclusive remarks and future development are made in Section 10.

2. RELATED WORK

Several investigations on the connection between haptic and auditory cues and perception of material stiffness have been done. Some look specifically at playing percussion musical instruments as the music community has long had a strong interest in haptic technology [3], where others investigate haptics, visuals and sound in relation to human computer interaction.

In [4], Sile O'Mohrain describes a series of studies where experienced musicians played VMIs with both haptic and auditory feedback with the aim of finding out whether adding haptic feedback to these instruments would improve their playability. The results indicate that the presence of haptic feedback can improve a player's ability to learn the behavior of a VMI.

In [5], Dahl gives a detailed analysis of four experienced drummers performing the same musical sequence using drumsticks on drums with three striking surfaces (soft, medium and hard). The study finds that the main parameter influencing the preparatory movement and the striking velocity was the dynamic level and, to a lesser extent, the striking surface.

The work of Avanzini and Crosato's [6], that of Passalenti *et al.* [7], and that of Liu et. al. use the haptic device PHANTOM® Omni™ (now Touch) [8]. Avanzini and Crosato test the influence of haptic and auditory cues on perception of material stiffness separately, in an experiment where subjects had to tap on virtual surfaces, and were presented with audio-haptic feedback. In each condition the haptic stiffness had the same value while the acoustic stiffness was varied. The study indicates that subjects consistently ranked surfaces according to the auditory stimuli. Passalenti *et al.*'s experiment focuses on haptics and guitar strings. In [9], a multimodal interface that synchronizes visual, haptic and auditory stimuli to give users a feeling of presence of virtual objects is presented and thoroughly detailed. The study notices that although the stiffness parameters of different materials were set to be the same, the sound effects biased user's judgment of the hardness of surfaces.

The preliminary experiment conducted in relation to DigiDrum takes inspiration from these works, but looks at the specificity of a VR-enhanced drum.

In [10], the design of a physically intuitive haptic drumstick is presented. The paper suggests that physically intuitive new musical instruments may help performers transfer motor skills from familiar, traditional musical instruments.

3. HAPTICS

The sense of touch is the first to develop in humans – a sense we cannot shut down. Vision is the last sense to develop, a sense we are able to "turn off" [11] by closing our eyes. Despite this, tactile awareness generally receives less attention than other sensory modalities when it comes to technological development [12]. We live in a world oversaturated by visuals, and VR is a technology where this has been the case notably. In this section, we describe in further detail haptic perception and how it works from a neurophysiological point of view as well as the basis for subjective decision making on tactile sensation.

3.1 Haptic Perception

The peripheral nervous system gathers environmental stimuli in form of visual, audible, tactile, olfactory (smell) and gustatory (taste) inputs and transfers them to the central nervous system for further elaboration and integration. Tactile information is collected in the skin, muscles, and joints and sent to an area in the brain called the primary somatosensory cortex [13]. This cortical area is the first stage for the tactile awareness occurring across the surface of the body. Several other structures of the central nervous system take part in the generation of tactile feedback, as generally, a single brain area is never responsible for information awareness [14]. Light touch and tactile attention are processed in the secondary somato-sensory cortex – an area directly connected with the primary somato-sensory cortex [15]. Literature reports that people undergoing tactile training improve their perception but also strengthen the connections and cortical representations of the stimulated body area [16]. There is a direct relationship between size of cortical region and haptic performance.

A specific area of the central parietal lobe, placed in the back of the primary somato-sensory cortex, integrates the information from the visual and haptic regions to help locate objects in space.

The sense of hearing is connected to the sense of touch and touching objects in different ways produce abundant sounds which convey information about the object and the interaction, such as material, shape, roughness, stiffness, the gesture, rate and strength of our actions. In VR systems, users may immediately notice the unnaturalness if the interface has no sound or provides mismatched sound [9].

As Cao *et al.* explain in [17], skilled interactions with sounding objects, such as drumming, rely on resolving the uncertainty in the acoustical and tactual feedback signals generated by vibrating objects.

3.2 Notes on Experiments Involving Haptics

Conducting experiments on haptics can prove difficult because there are no proper technological devices for delivering controlled and reliable tactile stimuli [12]. When users interact with a physical object, uncertainty may arise from

mis-estimation of the objects' geometry-independent mechanical properties, such as surface stiffness. How multisensory information feeds back into the fine-tuning of sound-generating actions remains unexplored [17].

In virtual environments (as used in VR) and using hand tracking devices such as Leap Motion [18] (see Section 5), subjects are able to move their hands freely, which could confound somato-sensory processing with activations related to motor planning and movement [19]. These uncontrolled motor activities result in uncontrolled somatic stimulation. There is an anatomical explanation of this close somato-motor functional relationship: areas involved in the perception of touch on the hands in the primary somato-sensory cortex are located mostly in front of the areas responsible for hand movements [20]. Another problem with haptics is the subjective quantification of the stimuli. Contents of tactile consciousness vary between individuals and a common lexicon to evaluate haptic sensation through surveys still seems far to be conceived [21].

3.3 Interaction between Visual Information and Tactile Feedback

In a famous experiment, Pavani *et al.* [22], asked a group of participants to detect the position of vibro-tactile stimuli on their arm. The participant's own arm was placed under a table (out of sight) while a fake rubber hand was laid in front of them. The rubber hand was laid out in a position that was anatomically compatible with participant's real hand. When seeing the mannequin hand being touched, all participants reported that their own hand was being touched, even though that was not the case. In short, the perception of tactile stimulation was simulated through visuals. A similar experiment was conducted by [23] asking subjects to watch a video of a hand being touched on the first finger while their own hand was stimulated synchronously. Brain activity during synchronous stimulation showed an improved tactile acuity. Taking into account previous literature findings, we can conclude that in virtual environments hand manipulations and interactions are important factors that enhance realism and user experience.

4. DESIGN CRITERIA FOR DIGIDRUM

As explained by [10], a new musical instrument is physically intuitive if the physics of haptic interaction are similar to those supported by a traditional musical instrument. Physically intuitive new musical instruments may help performers transfer motor skills from familiar, traditional musical instruments. This is why we choose to augment an existing drum, instead of suggesting a completely new musical instrument – seeing a physical drum will invite users to play the new instrument in an intuitive way and as a regular drum. The mechanics of a musical instrument's interface – what the instrument feels like – determines much of its playability [24].

In creating DigiDrum, the design criteria for VRMIs suggested by Serafin *et al.* were used as guidelines [2]. The setup integrates visuals, audio and haptics and extends an existing musical instrument using VR seeking to create a

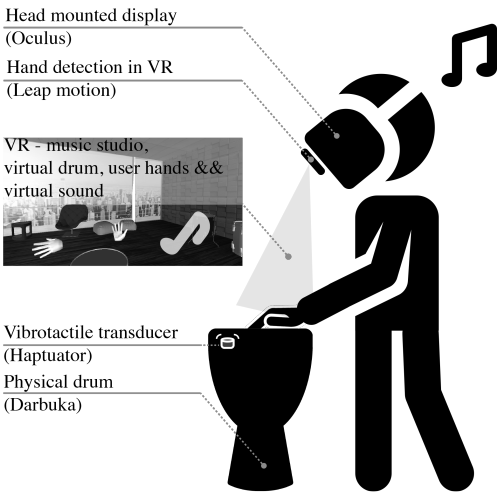

Figure 1: The physical setup of the system. The Leap Motion is mounted to the front of the HMD.

Figure 2: A user interacting with the setup.

"magical interaction". Creating a sense of presence is attempted by mapping the virtual drum's location to that of the physical one, and by representing the user's hands in the simulated world. DigiDrum was designed to create three types of illusions: (1) a place illusion – users should feel like they are in a music production studio, (2) a plausibility illusion – users should feel like the experience is really happening, and (3) virtual body ownership – users should see their own body in the virtual world and feel ownership of their virtual body.

5. SYSTEM OVERVIEW

Figure 1 shows the overall design of DigiDrum and its setup and Figure 2 shows a user interacting with the setup. For hand-tracking, the Leap Motion [18], which is an infrared-sensor-based camera that allows for accurate hand tracking is used. It is mounted to the front of an Oculus Rift HMD

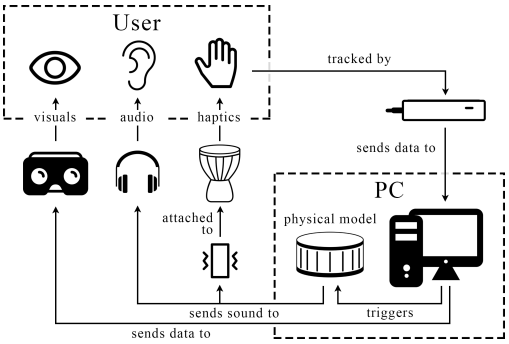

Figure 3: Detailed system layout. The user interacts with the system using their hands and gets haptic feedback from the haptuator attached to the drum membrane, auditory feedback from closed headphones and visual feedback from the Oculus Rift headset. A detailed explanation can be found in Section 5.

so that the user's hands are in the field of view when they look at the virtual drum. The drum is fixed in-place and played like a djembe. In the application, the virtual drum was placed slightly higher than the physical drum to make sure the physical model was triggered when the physical drum was hit.

A detailed overview of the system is given in Figure 3. The hand movement data is retrieved by a PC which runs the cross-platform game engine Unity [25]. The Unity 'scene' contains the virtual environment (see Section 6) that the user will see through the HMD and the physical model used for the sound and haptics (see Section 7). The HMD also sends data back to the PC regarding location and head rotation. Once the tracked hand touches (or collides with) the virtual drum, the physical model is triggered and its output sound is sent to a haptuator which is attached to the inside of the drum membrane. This effectively causes the physical membrane to be actuated by a virtual membrane. To accommodate for the plausability illusion mentioned in Section 4, the chosen haptuator has a very high fidelity, i.e. can play realistic audio signals as opposed to non-realistic 'buzzes'. Other forms of haptic feedback have been considered, but – according to the authors – the use of this actuator attached to the drum membrane had the highest potential of resembling realistic drum membrane vibration in the end.

Finally, the same sound that is sent to the haptuator is also sent to sound-isolating headphones. Sound-isolation is important as the sound coming from the physical drum should not interfere with the audio coming from the simulated drum.

6. UNITY IMPLEMENTATION

The virtual environment was created using Unity. All the hardware drivers and software components were linked together using this platform. Here, a virtual drum playable with hand motion using Leap Motion was created. The user enters the VR environment (rendered as a recording studio) and Leap Motion reconstructs (in VR) the subject's own hands. In the virtual recording studio a drum was placed at the center and programmed to detect collision with the reconstructed hands. When a collision was detected, a C# script, in which a physical model of a drum membrane was programmed, was activated to reproduce the beating sound of the drum through an actuator placed inside the drum skin.

7. PHYSICAL MODEL

The behaviour of musical instruments can be well described by partial differential equations (PDEs) [26]. In this section, the continuous-time PDE for a drum-membrane is given and explained. This is followed by an explanation of the discretisation method used. Finally, the parameter values used for the implementation are given.

7.1 Continuous Time

A rectangular (stiff) membrane with dimensions L_x (m) and L_y (m) can be described by the following equation [27]:

$$\rho H \frac{\partial^2 u}{\partial t^2} = T\Delta u - D\Delta\Delta u - 2\sigma_0 \frac{\partial u}{\partial t} + 2\sigma_1 \Delta \frac{\partial u}{\partial t}. \quad (1)$$

Here, state variable, $u = u(x, y, t)$ is a function of horizontal coordinate $x \in [0, L_x]$, vertical coordinate $y \in [0, L_y]$ and time $t \geq 0$ and is parameterised in terms of material density ρ (kg/m^3), membrane thickness H (m), tension T (N) and frequency independent and dependent damping coefficients σ_0 (s^{-1}) and σ_1 (m^2/s). Furthermore, $D = EH^3/12(1-\nu^2)$ with Young's modulus E (Pa) and Poisson's ratio ν. Lastly, Δ represents the 2D Laplacian [27]:

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}. \quad (2)$$

Furthermore, clamped boundary conditions – i.e., the state u at all plate edges and their gradients are 0 – have been chosen for simplicity:

$$u = \nabla u = 0 \quad \text{with} \quad \nabla = \frac{\partial}{\partial x} + \frac{\partial}{\partial y}. \quad (3)$$

7.2 Discretisation

For implementing the physical model, finite-difference time-domain (FDTD) methods were used [27]. These methods were chosen over others, such as the 2D waveguide mesh [28], as they allow parameters and real-time changes of these to be better controlled. FDTD methods discretise $u(x, y, t)$ shown in Equation (1) to $u^n_{(l,m)}$ using $t = nk$ with sample n and time step k (s), $x = lh$ where $l \in [0, ..., N_x - 1]$ and $y = mh$ where $m \in [0, ..., N_y - 1]$ where N_x and N_y are the number of horizontal and vertical grid points respectively. Furthermore, grid spacing h (m) can be calculated using

$$h \geq h_{\min} = 2\sqrt{\frac{c^2k^2 + 4\sigma_1 k + \sqrt{(c^2k^2 + 4\sigma_1 k)^2 + 4\kappa^2 k^2}}{2}}, \quad (4)$$

Parameter	Symbol (unit)	Value
Membrane width	L_x (m)	0.3
Membrane length	L_y (m)	0.3
Material density	ρ (kg/m^3)	10
Thickness	H (m)	0.001
Tension	T (N)	$\{15, 40, 80\}$
Young's modulus	E (Pa)	$2 \cdot 10^3$
Poisson's ratio	ν (-)	0.3
Freq. indep. damping	σ_0 (s^{-1})	$\{0.5, 2, 5\}$
Freq. dep. damping	σ_1 (m^2/s)	$[0, 0.005]$
Time step	k (s)	$1/44100$
Grid spacing	h (m)	$4h_{\text{min}}$

Table 1: Table showing parameter values.

where $c = \sqrt{T/\rho H}$ and $\kappa = \sqrt{D/\rho H}$. The closer h is to h_{min}, the higher the accuracy of the implementation.

7.3 Parameters

Most parameters used in the simulation were chosen empirically and can be found in Table 1. With these parameters a small (30×30 cm) membrane with a low density and stiffness is simulated. For the purpose of getting the model to work in real time, the minimum grid spacing h_{min} in Equation (4) is multiplied by 4 (h_{min} in (4) is calculated based on the highest value of T and $\sigma_1 = 0.005$). The values for T and σ_0 correspond to the cases used in the experiment. The frequency dependent damping σ_1 follows an exponentially decaying curve,

$$\sigma_1(t) = 0.005e^{-0.01t}, \tag{5}$$

where $t = 0$ at the time of excitation. This allows for very low damping, i.e., very long sound, while taking away some of the high frequency content present immediately after excitation. This ultimately results in a more natural drum sound, even when σ_0 is set low.

8. USER STUDY

This work hopes to add to the corpus of design guidelines for VRMIs, more specifically those VRMIs which involve a touch based stroking movement. In [2], Serafin et al. describe three layers of evaluation for VRMIs, namely: (1) investigating modalities of interaction, (2) evaluating VR specific aspects, with engagement being the most interesting from a VRMI perspective, and (3) looking at quality and goals of interaction.

An initial user study was conducted with a towfold goal: on the one hand - to study how users interact with DigiDrum and create guidelines for improving the setup, and on the other hand to investigate the relationship between tension and frequency independent damping coefficient (T and σ_0 respectively in Section 7) and user's perception of material stiffness.

As shown in Section 7, there are 3 different cases for both tension T and frequency independent damping σ_0. All combinations were tested, resulting in 9 different cases. Sound examples of each individual case can be found in [29]. Participants' experiences were evaluated through both qualitative and quantitative methods, namely by: (1) asking them during the test how they rate the stiffness of the material in each of the 9 cases, (2) a questionnaire including questions about their relationship and experience with playing a musical instrument, virtual body ownership and whether they thought their interaction patterns changed between the different cases and (3) observation while the participants interacted with the setup to retrieve data on engagement and stroke patterns which possibly correlate to the haptics and sound.

8.1 Process for the User Study

Before the experiment, participants were told that they would be "drumming in VR", that their perception of the stiffness of the material they were interacting with was tested and that their performances did not need to be musical in any way. Furthermore, participants were told they would hear 9 different cases in between which the "parameters of the experience" would be changed and that for each of these cases they would have to rate the stiffness of the material they were interacting with on a scale of 1 to 7, 1 being "extremely soft or loose", 7 being "extremely stiff or hard". The order in which the cases were presented was randomised to reduce bias. Between cases, the participants did not take off the headset or headphones, and the authors noted their answers. After the test, the participants filled out a questionnaire with the following questions (the last two taken from [6]):

- I felt like the hands in the simulation were my own. (1-7 rating)
- In order to express your judgements to the questions during the simulation, you relied mainly on... (multiple answers possible: visuals|audio|haptics)
- In your opinion what was varying between each condition? (multiple answers possible: visuals|audio|haptics)

From participant-observation during the experiment and the the final two questions of the questionnaire, "Did your behaviour change between different cases, and if so what did you do differently?" and "Anything you would like to add?", information on the user interaction and the quality of the setup was collected.

The experiment was done on 16 participants, 9 of which were experienced musicians ($>$ 5 years of instrument practice). Three participants were drummers.

9. RESULTS AND DISCUSSION

This section will give the results of the user study and discuss these. Due to the small sample size and some issues regarding interaction described at the end of this section, the presented results should be considered preliminary.

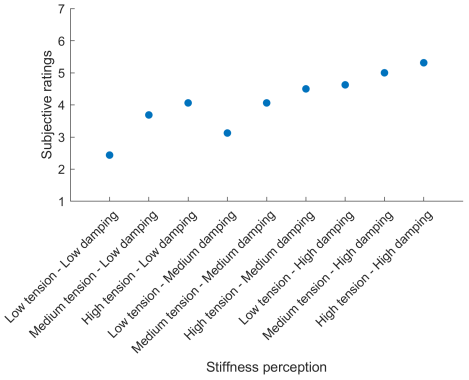

Figure 4: Relation between stiffness perception and subjective ratings.

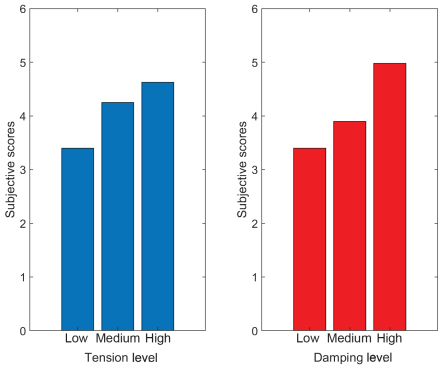

Figure 5: Subjective ratings grouped by different tension and damping levels.

9.1 Statistical Analysis

The results of the stiffness ratings can be found in Figure 4. Intriguingly, there was a significant correlation between the cases sorted by damping first and then by tension (both sorted from low to high) and the subjective ratings (ρ = 0.9372, p $<$ 0.01) using Spearman correlation. The Spearman methodology was used because the low number of values did not allow modelling a normal distribution [30]. A quasi-linear relationship between subjective stiffness perception and the values for tension and damping used by the simulation can be observed.

Figure 5 shows the average participant scores for each level of damping and tension both grouped in levels (low, medium and high). As previously hypothesised, the ratings of material stiffness increases with tension and damping.

A statistical analysis was run on each single level based on non-parametric Mann-Whitney U-test with the results reported in Table 2. This test helps to identify significant differences between groups in presence of small samples made by ordinal variables. Abbreviations are T for "tension" and D for "damping" while letters L, M and H mean the levels "low", "medium" and "high". It is important to take into account the multi-comparison problem and in this case the threshold level for significance should be equal to 0.0056 following the Bonferroni correction.

Mann-Whitney U-test [p-values]									
	LT LD	MT LD	HT LD	LT MD	MT MD	HT MD	LT HD	MT HD	HT HD
LT LD		.0642	.0163	.1268	.0049	.0041	.0034	.0004	.0002
MT LD	.0642		.6463	.3465	.6582	.1802	.1584	.034	.0091
HT LD	.0163	.6463		.2123	.8933	.5413	.4455	.1737	.0915
LT MD	.1268	.3465	.2123		.0791	.0254	.0283	.0012	.0009
MT MD	.0049	.6582	.8933	.0791		.3191	.3114	.0449	.0174
HT MD	.0041	.1802	.5413	.0254	.3191		.7732	.5214	.1383
LT HD	.0034	.1584	.4455	.0283	.3114	.7732		.7725	.3424
MT HD	.0004	.034	.1737	.0012	.0449	.5214	.7725		.2991
HT HD	.0002	.0091	.0915	.0009	.0174	.1383	.3424	.2991	
Note: Bonferroni adjusted significance threshold for multi-comparison p$<$0.0056									

Table 2: Mann-Whitney U-test (p-values).

Different levels of tension/damping		p-values
Low Damping	Medium Damping	0.2560
Medium Damping	High Damping	0.0078
Low Damping	High Damping	6.5071e-04
Low Tension	Medium Tension	0.0950
Medium Tension	High Tension	0.4268
Low Tension	High Tension	0.0297

Table 3: Comparison between different levels of tension and damping.

As we can observe from Table 2, there isn't a significant difference between "high tension – low damping" and "low tension – medium damping" (p = 0.2123) suggesting that the linear relation shown in Figure 5 holds despite the discontinuity between points as seen on the scatterplot. However, we should consider it similar to a monotonically increasing function rather than a pure linear trend.

Lastly, Table 3 shows group comparisons between the three different values of damping and tension. A significant difference in participant's ratings between medium-high damping and low-high damping levels can be noticed while tension shows significance only between low to high tension. It can be deducted from the results that damping is a more important factor than tension in material stiffness perception. This result was unexpected, as it was hypothesised that tension would be the most dominant factor in stiffness perception. Additionally, it appears difficult for participants to evaluate low to medium levels of both damping and tension. In a future test, the values could be chosen differently, or more alternatives for the parameter values could be investigated to better see the perceptual differences between these values.

9.2 Statistical Analysis: Reliability

Individual ratings were initially analysed with Cronbach's alpha [31] to test the internal consistency of the responses. This measure is generally known as a metric to validate a questionnaire with higher values of alpha as those more desirable. The non-standardised Cronbach's alpha value was 0.6348 while the standardised value reached 0.6589. According to [32], a value between 0.6 to 0.7 is questionable (questionnaire scale is not fully reliable) with 0.7 as

Question	Result
I felt like the hands in the simulation were my own. (1-7 rating)	$\mu = 5.44$, $\sigma = 1.26$
In order to express your judgements to the questions during the simulation, you relied mainly on... (visuals|audio|haptics)	visuals: 0, audio: 15, haptics: 5
In your opinion what was varying between each condition? (visuals|audio|haptics)	visuals: 0, audio: 14, haptics: 10

Table 4: Questionnaire results. The last two questions were taken from [6].

the threshold for an acceptable test. Despite the outcomes being slightly below threshold (probably caused by subjective difficulties in evaluating stiffness), it appears that in future a good reliability can be reached by increasing the sample size. Moreover, if we don't consider all factors loadings as evenly distributed, we could assume that the Cronbach's alpha underestimates the true reliability.

9.3 Questionnaire

The questionnaire results in Table 4 show that the participants generally found that the hands in the simulation were their own. This proves that the Leap Motion is a good way to track the hands and that it was well implemented. The visuals had no influence on participants' judgement, probably because they were unchanged. The audio seemed to be the most predominant feature the participants focused on when expressing their judgements (93.8%). Haptics for expressing judgements was only chosen by 5 participants (31.3%). In the future, removing the audio, only leaving the haptics might be a better way to force the participants to use their sense of touch and test the influence of this modality on perception.

9.4 Qualitative Observations

From participant observation during the experiment, comments they gave during and after the test, and the two last (open) questions of the questionnaire (see Section 8) additional findings were compiled.

Due to the fact that the virtual drum was placed slightly higher than the physical drum (see Section 5), many participants interacted with the air above the drum rather than finishing their stroke to actually hit the drum. This was an issue, as the haptic sensation would not be felt in that case. This might also explain the result of the second question in Table 4. Either before or during the test, the participants were instructed to finish their stroke to actually physically interact with the drum.

The interaction was programmed in such a way, that when a tracked hand collides with the virtual drum, this hand would not be able to trigger the physical model until it was completely out of the "collision zone". Due to the misalignment mentioned above, many interactions were not captured. Again, either before or during the experiment, the participants were instructed to make longer movements to ensure that their hands were completely outside of this "collision zone" before interacting with the drum again.

Another technical issue was that sometimes participants would look forward rather than down to the hands. This caused the hands not to be tracked anymore as the Leap Motion was mounted on the HMD. A solution for this would be to mount it at a lower angle rather than straight forward (as is the current case).

The experiment could be improved by addressing the above interaction issues to yield stronger data. The issues could potentially be solved by adding a more precise and reliable sensor to the setup, such as a contact microphone placed on the drum membrane. Even though a feedback loop could occur due to the haptuator being present on the same membrane, there is a potential to filter out its vibrations and only use the transients due to the interaction with the membrane for control.

Some participants commented that they would have liked to have reference points for "the stiffest" and "the softest" cases before testing as they said they would have judged the first few cases differently if they had known these references in advance. This could, however, bias the participants' answers.

The movements of participants were observed during the test and sporadically noted. There was a small tendency towards slower and longer movements in the case of lower tension and faster and shorter movements in the opposite case, but as these observations were not done systematically, to be able to say anything about this, this should be properly tested, possibly using raw data from the hand tracking.

10. CONCLUSION

In this paper, we presented and evaluated a novel VRMI where a physical drum was enhanced by VR. The physical drum was augmented by a vibration motor and the sound was simulated using a physical model of a drum membrane. In an experiment run during the study, preliminary results show that higher values for both tension and damping increase the perception of material stiffness of the drum membrane, as hypothesised. However, the damping appeared to be a more important factor in this perception than the tension, which was contrary to expectations.

In future work, improving the experiment by, for example, adding a contact microphone to the membrane for more accurate control and re-conducting the experiment with a larger sample size will be necessary to validate or improve the results presented in this paper.

Other future work includes decoupling the audio and the haptics, to test the perceptual influence of each individual modality separately. More alternatives of the parameter values could be presented in a future test to more deeply investigate the connection between parameter values and stiffness perception.

Additionally, the tracking of the user's hands should be improved by mounting the Leap Motion more downwards on the HMD. Furthermore, the virtual and physical drum should be better aligned in space as to make the interaction less confusing and more intuitive. Lastly, in order to test

whether the interaction patterns change depending on the changes in parameters, the raw data from the hand tracking should be analysed.

Acknowledgments

The authors wish to thank Stefania Serafin, Cumhur Erkut, Niels Nilsson, Rolf Nordahl, and Michele Geronazzo – our teachers of the "Virtual, Augmented, Mixed realities" PhD course at Aalborg University Copenhagen – for their help, and all the participants who were kind enough to participate in our experiment.

11. REFERENCES

[1] S. Serafin, N. Nilsson, C. Erkut, and R. Nordahl, "Virtual reality and the senses - whitepaper," *Danish Sound Innovation Network*, 2017.

[2] S. Serafin, C. Erkut, J. Kojs, N. Nilsson, , and R. Nordahl, "Virtual reality musical instruments: State of the art, design principles, and future directions," *Computer Music Journal*, 2016.

[3] E. Berdahl, H.-C. Steiner, and C. Oldham, "Practical hardware and algorithms for creating haptic musical instruments," in *NIME*, 2008.

[4] M. S. O'Modhrain, *Playing by Feel: Incorporating Haptic Feedback into Computer-Based musical Instruments*. Ph.D. Dissertation, Stanford University, 2001.

[5] S. Dahl, "Playing the accent - comparing striking velocity and timing in an ostinato rhythm performed by four drummers," *Acta Acustica united with Acustica*, vol. 90, pp. 762 – 776, 2004.

[6] F. Avanzini and P. Crosato, "Haptic-auditory rendering and perception of contact stiffness," *Lecture Notes in Computer Science*, vol. 4129, pp. 24–35, 2006.

[7] A. Passalenti, R. Paisa, N. Nilsson, N. Andersson, F. Fontana, R. Nordahl, and S. Serafin, "No strings attached: Force and vibrotactile feedback in a virtual guitar simulation," *IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 1116–1117, 2019.

[8] 3D Systems, Inc, "3D Systems Touch Haptic Device," accessed November 4, 2019, available at https://www.3dsystems.com/haptics-devices/touch.

[9] J. Liu and H. Ando, "Hearing how you touch: Real-time synthesis of contact sounds for multisensory interaction," *Conference on Human System Interactions*, pp. 275–280, 2008.

[10] E. Berdahl, B. Verplank, J. O. Smith, and G. Niemeyer, "A physically intuitive haptic drumstick," in *ICMC*, 2007.

[11] K. Barnett, "A theoretical construct of the concepts of touch as they relate to nursing," *Nursing Research*, vol. 21, pp. 102–110, 1972.

[12] A. Gallace, M. K. Ngo, J. Sulaitis, and C. Spence, "Multisensory presence in virtual reality: possibilities & limitations," *Multiple sensorial media advances and applications: New developments in MulSeMedia*, pp. 1–38, 2012.

[13] M. Blatow, E. Nennig, A. Durst, K. Sartor, and C. Stippich, "fMRI reflects functional connectivity of human somatosensory cortex," *Neuroimage*, vol. 37, pp. 927–936, 2007.

[14] T. Manzoni, F. Conti, and M. Fabri, "Callosal projections from area SII to SI in monkeys: Anatomical organization and comparison with association projections," *Journal of Comparative Neurology*, vol. 252, pp. 245–263, 1986.

[15] S. B. Eickhoff, A. Schleicher, K. Zilles, and K. Amunts, "The human parietal operculum. I. Cytoarchitectonic mapping of subdivisions," *Cerebral cortex*, vol. 16, no. 2, pp. 254–267, 2005.

[16] D. N. Saito, T. Okada, M. Honda, Y. Yonekura, and N. Sadato, "Practice makes perfect: The neural substrates of tactile discrimination by Mah-Jong experts include the primary visual cortex," *BMC Neuroscience*, vol. 7, no. 79, 2007.

[17] Y. Cao, B. L. Giordano, F. Avanzini, and S. McAdams, "The dominance of haptics over audition in controlling wrist velocity during striking movements," *Experimental Brain Research*, vol. 234, pp. 1145—1158, 2016.

[18] UltraLeap Ltd, "Leap Motion," accessed November 4, 2019, available at https://www.leapmotion.com/.

[19] A. Bodegard, S. Geyer, C. Grefkes, K. Zilles, and P. Roland, "Hierarchical processing of tactile shape in the human brain," *Neuron*, vol. 31, pp. 317–328, 2001.

[20] W. Penfield and T. L. Rasmussen, *The cerebral cortex of man: A clinical study of localization of function*. London: Macmillan, 1950.

[21] A. Gallace and C. Spence, *Touch and the body: The role of the somatosensory cortex in tactile awareness*. Psyche: An Interdisciplinary Journal of Research on Consciousness, 2010.

[22] F. Pavani, C. Spence, and J. Driver, "Visual capture of touch: Out-of-the-body experiences with rubber gloves," *Psychological Science*, vol. 11, pp. 353–359, 2000.

[23] M. Schaefer, H. Flor, H. J. Heinze, and M. Rotte, "Dynamic modulation of the primary somatosensory cortex during seeing and feeling a touched hand," *Neuroimage*, vol. 29, pp. 587–592, 2006.

[24] S. O'Modhrain and R. B. Gillespie, *Once More, with Feeling: Revisiting the Role of Touch in Performer-Instrument Interaction*, 2018.

[25] Unity Technologies, "Unity Real-Time Development Platform — 3D, 2D VR & AR Visualizations," accessed November 4, 2019, available at https://unity.com/.

[26] N. H. Fletcher and T. D. Rossing, *The Physics of Musical Instruments*. Springer, 1998.

[27] S. Bilbao, *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics*. John Wiley & Sons, 2009.

[28] S. Van Duyne and J. O. Smith, "The 2-D digital waveguide mesh," *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 177–180, 1993.

[29] S. Willemsen, "Sound Files (MembraneOutputs.zip)," accessed April 26, 2020, available at https://github.com/SilvinWillemsen/Membrane/blob/Paper/Sound_Files/MembraneOutputs.zip.

[30] R. Kirk, *Statistics: an introduction*. Nelson Education, 2007.

[31] L. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297—-334, 1951.

[32] P. Kline, *The handbook of psychological testing (2nd ed.)*. London: Routledge, 2000.

Paper H

Dynamic Grids for Finite-Difference Schemes in Musical Instrument Simulations

Paper I

A Physical Model of the Trombone using Dynamic Grids for Finite-Difference Schemes