
Real-Time Implementation of Bowed Instruments Using Static and Dynamic Friction Models

Master Thesis
Marius George Onofrei

Aalborg University
IT and Design



IT and Design
Aalborg University
<http://www.aau.dk>

AALBORG UNIVERSITY
STUDENT REPORT

Title:

Real-Time Implementation of Bowed Instruments Using Static and Dynamic Friction Models

Theme:

Physical Models for Sound Synthesis

Project Period:

Fall 2020 - Spring 2021

Project Group:

-

Participant(s):

Marius George Onofrei

Supervisor(s):

Silvin Willemsen, Stefania Serafin

Copies: 1

Page Numbers: 83

Date of Completion:

April 7, 2021

Abstract:

This thesis presents research on friction models of varying degrees of complexity and their application in physical modelling sound synthesis. For this aim, the focus is to develop physical models based on finite difference time-domain methods simulating two friction based instruments: a violin, modelled as a bowed stiff string connected to a resonant plate and a friction drum, modelled as a bowed membrane connected to an acoustic tube.

A comparison is desired of the behavior and sound using a static friction model, where the friction force is only dependent on the relative velocity of the interacting elements and a dynamic model, where the state of the system is dependent on its history, hysteresis.

The aim is to implement these models in a real-time sound synthesis software, with the main desire being not the faithful reproduction of the original acoustic instruments, but the development of new instruments based on the physics of their acoustic counterparts, which can be extended or modulated in ways that are not possible in the physical world.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Contents

1	Introduction	1
1.1	Violino Arpa	1
1.2	Friction Drum	2
1.3	Project Goals	4
2	Finite Difference Method	5
2.1	Terminology and Notation	5
2.2	Interpolation and Spreading Operators	8
2.3	Identities	10
3	Friction Models	11
3.1	Friction Mechanisms	11
3.1.1	Coulomb Friction	11
3.1.2	Viscous Friction	12
3.1.3	Stribeck Friction	12
3.2	Static Friction Model	13
3.3	Dynamic Friction Model	15
3.4	Elasto-Plastic Model	15
4	Violino Arpa	19
4.1	Continuous Model	19
4.1.1	Bowed Stiff String	19
4.1.2	Plate	20
4.1.3	Connection	20
4.1.4	Bowing Force	21
4.1.5	Complete System	21
4.2	Discretization	21
4.2.1	Bowed Stiff String	22
4.2.2	Plate	23
4.2.3	Connection	24
4.2.4	Bowing Force	25

4.2.5	Solving the Complete System	25
4.3	Implementation	27
4.3.1	Prototype	27
4.3.2	Real-Time Application	33
4.3.3	Real-Time Comparison of Friction Models	35
5	Friction Drum	39
5.1	Continuous Model	39
5.1.1	Membrane	39
5.1.2	Acoustic Tube	40
5.1.3	Connection	41
5.1.4	Bowing Force	41
5.1.5	Complete System	41
5.2	Discretization	41
5.2.1	Membrane	42
5.2.2	Acoustic Tube	43
5.2.3	Connection	44
5.2.4	Bowing Force	44
5.2.5	Solving the Complete System	45
5.3	Implementation	46
5.3.1	Prototype	48
5.3.2	Real-Time Application	49
5.3.3	Real-Time Comparison of Friction Models	52
6	Evaluation	55
6.1	Violino Arpa Sound Samples	55
6.2	Friction Drum App Demo	56
7	Conclusion	59
7.1	Summary	59
7.2	Future Work	59
	Bibliography	61
A	Violino Arpa Paper	65
B	Friction Drum Paper	75

Preface

This thesis was made as a completion of the Sound and Music Computing master's programme at Aalborg University Copenhagen.

The work presented in this project resulted in a paper which is currently under peer review for the 23rd International Conference on Digital Audio Effects, DAFx 2020in21 Vienna. The paper can be found in Appendix B.

Also some of the work related to this project has been included in a collaborative project at Aalborg University, where the experience of playing the Violino Arpa, a musical instrument exhibited at the Danish Music Museum, has been recreated using augmented reality. This resulted in a paper which has been submitted to the 18th Sound and Music Computing Conference, SMC2021, and is currently under peer review. The paper can be found in Appendix A.

I would like to express my sincere thanks to my supervisors for this thesis: Silvin Willemsen and Stefania Serafin who have always been within reach to answer any of my questions related to the project, as well as providing excellent guidance and support along the way.

Aalborg University, April 7, 2021



Marius George Onofrei
<monofr11@student.aau.dk>

Chapter 1

Introduction

The behavior of musical instruments can be characterized mathematically according to the most basic laws of physics by systems of equations, which can be solved to produce simulations of their sound. Bilbao et al. [3] give a detailed review on the state of physical modelling sound synthesis (PMSS). Solutions to this problem typically involve splitting the system into an excitation part and a resonator part, where the former can be strongly non-linear, like a frictional mechanism, which is the focus of this current project. The latter is typically modelled as a linear system which throughout the history of PMSS lead to the interpretation of such systems to descriptions in terms of modes of vibration (modal synthesis) [15] or travelling wave solutions (digital waveguides) [29]. Another idea is using more direct time-stepping methods, where the components of the system are represented over grids and then the solution is propagated over discrete time intervals [2]. One such family of possible solutions are finite difference methods. The drawback of using such methods is the higher demand on computational resources, but as the performance of computer processors continues to increase keeping Moore's law alive and well, [14], their use for simulating the systems of equations which describe the physics of music instruments fast enough for real-time audio becomes a possibility.

This project deals with the real-time implementation of two "exotic" music instruments driven by friction by means of numerical simulations using finite difference time domain (FDTD) methods: a bowed string instrument, the Violino Arpa and a friction drum, where a 2-D membrane is bowed.

1.1 Violino Arpa

The first instrument tackled is the Violino Arpa (also known as Amoeba Violin), which is a bowed string instrument found at the Danish Music Museum, which stands out due to the peculiar shape of its body. Figure 1.1 shows a picture of the instrument. Due to its age and the desire for preservation, the instrument



Figure 1.1: A photograph of the Violino Arpa owned by the Danish Music Museum.

cannot be played by the general public, which makes the simulation of its sound an interesting task.

Bowed string instruments are perhaps some of the most famous of the friction musical instrument category and are a great starting point for the investigation of friction as an excitation mechanism in the context of physical modelling sound synthesis. This is due to the fact that the string can be modeled as a 1-D element and its transverse displacement can be described starting from a widely investigated partial differential equation (PDE) in physics, namely the 1-D wave equation.

Being often encountered in popular culture, the sound of bowed strings has been thoroughly investigated by many authors. McIntyre et al. [13] tackled the simulation of bowed string sounds in some of the first musical non-linear systems. A digital waveguide (DW) method was used by Smith [31] who published the first real-time implementation of a bowed string, while another real-time implementation of the sound was proposed by Florens et al. [11] using a mass-spring system to model the string coupled a static friction model for the bow.

More complex friction models like the elasto-plastic model described in Section 3.4 has been implemented by Serafin et al. [28] using a DW approach, while Willemsen et al. [36] implemented a FDTD based real-time implementation, which will be used as a benchmark in this project.

1.2 Friction Drum

Bowed string instruments are the perhaps most distinguished in the category of frictional musical instruments, as they form the backbone of the symphony orchestra. However, other interesting instruments with a frictional based excitation mechanism exist and they can be modelled to produce some interesting audio applications using PMSS methods. Developing a real-time audio application sim-



Figure 1.2: The friction drum exhibited at the Danish music museum.

ulating one such instrument is the second part of this project. The instrument in question is the friction drum, which has been described as a peculiar music instrument or even a noisy toy, [1] is one such instrument. Throughout different cultures around the world it has many different names: like *cuica* in Brazil, *putipù* in Italy, *zambomba* in Spain, *buhai* in Romania and variations on *rummelpot* in Germanic countries like Denmark, Germany or the Netherlands. It is an instrument often used as an accompaniment in folk traditions, for example being played by children during Christmas holiday songs in 19th century Denmark [16].

Although its design might differ slightly between cultures, it is essentially a membrane tensioned on top of a cylinder (which acts as a sound box), with a stick inserted in the middle. It is played by either rubbing the stick up and down the membrane or in other cases by rubbing the stick itself with a cloth and having the vibrations transferred to the membrane. Essentially it uses a frictional excitation, hence the name friction drum. Figure 1.2 shows a picture of the friction drum present at the Danish Music Museum.

What makes the friction drum a very interesting instrument in terms of physical modelling sound synthesis is the fact that it could allow for the possibility of engaging with a virtual version of the real instrument in ways which may be impossible in the real world. Particularly, the bowing position of a friction drum is fixed. The membrane is tensioned around the stick and moving it in the plane of the membrane would be impossible. This limitation would not be an issue in a virtual model. Additionally, other physical parameters could be adjusted dynamically, like the tension of the membrane or its damping characteristics, which again

would not be possible with a real friction drum. In essence it would allow for creating a "new" instrument based on the physics of its acoustic counterpart.

Another interesting aspect is in terms of the engineering exercise alone. Based on the author's investigations, not a lot of research has been carried out in terms of coupling a friction excitation model (bowing model for short) with a 2-D element, for sound synthesis. Huynh [9] has made use of the sound of a bowed plate in a perception study in her thesis. For sound generation she used an implementation available in Modalys, [25], which uses modal synthesis techniques. In terms of FDTD methods Bilbao shows results of bowed plate simulations using a static friction model in his essential book on numerical sound synthesis [2]. However, it is not clear if a real-time implementation was used. No previous work has been found, using a more advanced friction model like the elasto-plastic dynamic model presented in Chapter 3. This is perhaps because of the much greater interest in bowed string instruments and the fact that the bowed membrane is fundamentally the same problem but just extended in 2-D.

1.3 Project Goals

The detailed description of the development of the audio apps, which was the main goal of the project, is given in Chapter 4 and Chapter 5. The FDTD methods used for the simulation are described in Chapter 2. A focus is placed on different friction models, which are described in Chapter 3 and their effect on the resulting sound. Chapter 6 presents the work related to evaluation. Finally, Chapter 7 presents the conclusion and highlights possible future work related to this project. Demonstrative videos for the two audio applications can be found at [23] and [24].

Chapter 2

Finite Difference Method

Physical phenomena can often be described by evaluating the rates of change of the state variables of the considered process with respect to time and/or space parameters. This can be represented in mathematical formulation as either ordinary differential equations (ODEs) in the case of single-variable functions or partial differential equations (PDEs) for multi-variable functions. The vibrational state of many elements of musical instruments can be characterized by these types of equations. An example being the motion of a string of a violin or the motion of its resonant plate. Consider u [m] the transverse displacement of a vibrating string. This displacement will vary both with respect to time t [s] as well as with respect to its position along the length of the string, denoted x [m]. Hence $u(x, t)$ is a multi-variable function. In the case of the plate, as it spans two spatial dimensions, say x [m] describes the length of the plate and y [m] its width, its displacement w [m] will be dependent on both of these in addition to time, i.e. $w(x, y, t)$.

Systems of PDEs describing real-world phenomena are typically difficult to solve analytically and rarely have a closed-form solution. However, using numerical analysis methods, these systems of equations can be discretized into grid points and converted to a system of equations which can be solved using linear algebra techniques. Thus an approximation to the continuous solution can be achieved. One such numerical analysis technique is the Finite Difference Method, which is used in the current project. This chapter presents an overview of the building blocks of this method: the finite difference operators and identities used throughout this project.

2.1 Terminology and Notation

The notations and terminology used in the finite difference schemes (FDSs), i.e. a discretization structures of continuous differential equations, which are presented throughout the report are taken from [2].

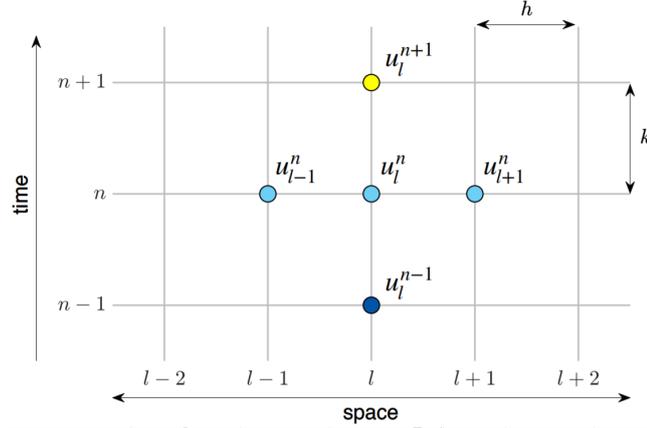


Figure 2.1: Illustration of the discretization of a continuous function $u = u(x, t)$ as a grid function, u_l^n . From [34].

Consider a continuous time state variable $u(x, t)$ dependent on one spatial dimension x and on time, t . This multi-variable function can be discretized and approximated as a grid function u_l^n across a time-space grid given by $x = lh$ and $t = nk$, where h is a spatial step and k is a time step. Then, l and n are both integers counting the the number of steps in space and in time with $l \in [0, N_x]$ and $n \in \mathbb{N}$. Here, N_x is the number of grid intervals the spatial domain is divided in. Figure 2.1 shows such a grid, highlighting the values of the grid function u_l^n at the "next" time step u_l^{n+1} and the "previous" time step u_l^{n-1} and similarly the neighbours in space, i.e. u_{l+1}^n and u_{l-1}^n .

For the case of a state variable with multi-dimensional space, the same type of discretization occurs which would be added as another grid layer in Figure 2.1. A y dimension for instance is approximated by $y = mh_y$ with h_y being the spatial step in y direction and $m \in [0, N_y]$ being the step count, with N_y being the number of grid intervals the spatial domain is divided in y direction.

With this framework in place, operators can be introduced, which denote actions which can be applied to the discrete grid function u_l^n . The most basic of these are shift operators which can be forward, backward or stationary (described as the identity operation "1"):

$$e_{t+}u_l^n = u_l^{n+1} \quad e_{t-}u_l^n = u_l^{n-1} \quad e_{x+}u_l^n = u_{l+1}^n \quad e_{x-}u_l^n = u_{l-1}^n \quad 1u_l^n = u_l^n \quad (2.1)$$

Starting from these shift operators, now difference operators can be introduced. These are of particular importance within the finite difference method as they are discrete approximations to the continuous derivatives which are the building blocks of differential equations. An example of such difference operators with respect to time t are:

$$\delta_{t+} \triangleq \frac{1}{k} (e_{t+} - 1) \approx \frac{d}{dt} \quad (2.2a)$$

$$\delta_{t-} \triangleq \frac{1}{k} (1 - e_{t-}) \approx \frac{d}{dt} \quad (2.2b)$$

$$\delta_t \triangleq \frac{1}{2k} (e_{t+} - e_{t-}) \approx \frac{d}{dt} \quad (2.2c)$$

In the case of spatial difference operators, the denominator will be the relevant spatial step, h for the case of u_l^n , as opposed to k .

Other operators of interest which are used to build more accurate FDSs, are averaging operators. An example for averaging operators with respect to time t is given below:

$$\mu_{t+} = \frac{1}{2} (e_{t+} + 1) \approx 1 \quad (2.3a)$$

$$\mu_{t-} = \frac{1}{2} (1 + e_{t-}) \approx 1 \quad (2.3b)$$

$$\mu_t = \frac{1}{2} (e_{t+} + e_{t-}) \approx 1 \quad (2.3c)$$

The partial derivatives of the continuous function $u = u(x, t)$ can be approximated in the following way:

$$\frac{\partial u}{\partial t} = \begin{cases} \delta_{t+} u_l^n = \frac{1}{k} (u_l^{n+1} - u_l^n) & \text{Forward time difference} \\ \delta_{t-} u_l^n = \frac{1}{k} (u_l^n - u_l^{n-1}) & \text{Backward time difference} \\ \delta_t u_l^n = \frac{1}{2k} (u_l^{n+1} - u_l^{n-1}) & \text{Centered time difference} \end{cases} \quad (2.4)$$

$$\frac{\partial u}{\partial x} = \begin{cases} \delta_{x+} u_l^n = \frac{1}{h} (u_{l+1}^n - u_l^n) & \text{Forward difference in space} \\ \delta_{x-} u_l^n = \frac{1}{h} (u_l^n - u_{l-1}^n) & \text{Backward difference in space} \\ \delta_x u_l^n = \frac{1}{2h} (u_{l+1}^n - u_{l-1}^n) & \text{Centered difference in space} \end{cases} \quad (2.5)$$

Furthermore, the difference operators can be combined to approximate higher order partial derivatives.

$$\delta_{tt} = \delta_{t+} \delta_{t-} \approx \frac{d^2}{dt^2}, \quad (2.6)$$

therefore:

$$\frac{\partial^2 u}{\partial t^2} \approx \delta_{tt} u_l^n = \delta_{t+} \delta_{t-} u_l^n = \frac{1}{k^2} (u_l^{n+1} - 2u_l^n + u_l^{n-1}) \quad (2.7a)$$

$$\frac{\partial^2 u}{\partial x^2} \approx \delta_{xx} u_l^n = \delta_{x+} \delta_{x-} u_l^n = \frac{1}{h^2} (u_{l+1}^n - 2u_l^n + u_{l-1}^n) \quad (2.7b)$$

Using the building blocks presented above, a list of all the difference operators which are used in this project are given in Equation (2.8). As the largest spatial dimension used is 2, a grid function $u(x, y, t) \approx u_{l,m}^n$ is considered. For simplicity the spatial step in both x and y dimensions is considered equal, i.e. $x = lh$ and $y = mh$. Throughout the project, a variety of state variables and their corresponding grid functions will be considered, but the action of the operators on the grid functions $u_{l,m}^n$ will be the same for all of them:

$$\partial_t u \approx \delta_t u_{l,m}^n = \frac{1}{2k} (u_{l,m}^{n+1} - u_{l,m}^{n-1}), \quad (2.8a)$$

$$\partial_{t-} u \approx \delta_{t-} u_{l,m}^n = \frac{1}{k} (u_{l,m}^n - u_{l,m}^{n-1}), \quad (2.8b)$$

$$\partial_{tt}^2 u \approx \delta_{tt} u_{l,m}^n = \frac{1}{k^2} (u_{l,m}^{n+1} - 2u_{l,m}^n + u_{l,m}^{n-1}), \quad (2.8c)$$

$$\partial_x^2 u \approx \delta_{xx} u_{l,m}^n = \frac{1}{h_x^2} (u_{l+1,m}^n - 2u_{l,m}^n + u_{l-1,m}^n), \quad (2.8d)$$

$$\partial_y^2 u \approx \delta_{yy} u_{l,m}^n = \frac{1}{h_y^2} (u_{l,m+1}^n - 2u_{l,m}^n + u_{l,m-1}^n), \quad (2.8e)$$

$$\Delta u = \partial_x^2 u + \partial_y^2 u \approx \delta_{\Delta} u_{l,m}^n = \delta_{xx} u_{l,m}^n + \delta_{yy} u_{l,m}^n, \quad (2.8f)$$

$$\Delta \Delta u = \partial_x^4 u + 2\partial_x^2 \partial_y^2 u + \partial_y^4 u \approx \delta_{\Delta \Delta} u_{l,m}^n = \delta_{xxxx} u_{l,m}^n + 2\delta_{xxyy} u_{l,m}^n + \delta_{yyyy} u_{l,m}^n, \quad (2.8g)$$

$$\partial_x^4 u \approx \delta_{xxxx} u = \frac{1}{h_x^4} (u_{l+2,m}^n - 4u_{l+1,m}^n + 6u_{l,m}^n - 4u_{l-1,m}^n + u_{l-2,m}^n) \quad (2.8h)$$

$$\partial_y^4 u \approx \delta_{yyyy} u = \frac{1}{h_y^4} (u_{l,m+2}^n - 4u_{l,m+1}^n + 6u_{l,m}^n - 4u_{l,m-1}^n + u_{l,m-2}^n) \quad (2.8i)$$

$$u \approx \mu_t u_{l,m}^n = \frac{1}{2} (u_{l,m}^{n+1} + u_{l,m}^{n-1}), \quad (2.8j)$$

$$u \approx \mu_{t-} u_{l,m}^n = \frac{1}{2} (u_{l,m}^n + u_{l,m}^{n-1}). \quad (2.8k)$$

2.2 Interpolation and Spreading Operators

As the continuous space is discretized and data will only be available at certain locations, i.e. integer multiples of the spatial step h , it becomes an issue when wanting to extract outputs or add inputs at locations which lie between spatial grid points. In order to tackle this, interpolation and spreading operators are introduced.

Interpolation is used to find values of the grid function u_l^n at values between the points of the spatial grid, spread over a domain \mathcal{D} . This is done by multiplying with scaling factors values which bound the desired location, $x_o \in \mathcal{D}$. An example of possible interpolation functions I of increasing order, highlighted by the subscript, is given below:

$$I_0(x_o)u_l^n = u_{l_o}^n, \quad (2.9a)$$

$$I_1(x_o)u_l^n = (1 - \alpha_o)u_{l_o}^n + \alpha_o u_{l_o+1}^n, \quad (2.9b)$$

$$I_3(x_o)u_l^n = \frac{\alpha_o(\alpha_o - 1)(\alpha_o - 2)}{-6}u_{l_o-1}^n + \frac{(\alpha_o - 1)(\alpha_o + 1)(\alpha_o - 2)}{2}u_{l_o}^n \\ + \frac{\alpha_o(\alpha_o + 1)(\alpha_o - 2)}{-2}u_{l_o+1}^n + \frac{\alpha_o(\alpha_o + 1)(\alpha_o - 1)}{6}u_{l_o+2}^n, \quad (2.9c)$$

with, $l_o = \text{floor}(x_o/h)$ and $\alpha_o = x_o/h - l_o$.

When one wants to excite the grid function u_l^n , defined over a 1-D domain \mathcal{D} , at a desired location, x_i which lies in-between the points of the grid space, this excitation needs to be somehow distributed over available points. This is where spreading functions come into play, which are in fact duals of the interpolation functions. Below spreading functions J of different orders are given:

$$J_0(x_i) = \frac{1}{h} \begin{cases} 1, & l = l_i \\ 0, & \text{otherwise} \end{cases} \quad (2.10a)$$

$$J_1(x_i) = \frac{1}{h} \begin{cases} (1 - \alpha_i), & l = l_i \\ \alpha_i, & l = l_i + 1 \\ 0, & \text{otherwise} \end{cases} \quad (2.10b)$$

$$J_3(x_i) = \frac{1}{h} \begin{cases} \frac{\alpha_o(\alpha_o-1)(\alpha_o-2)}{-6}, & l = l_i - 1 \\ \frac{(\alpha_o-1)(\alpha_o+1)(\alpha_o-2)}{2}, & l = l_i \\ \frac{\alpha_o(\alpha_o+1)(\alpha_o-2)}{-2}, & l = l_i + 1 \\ \frac{\alpha_o(\alpha_o+1)(\alpha_o-1)}{6}, & l = l_i + 2 \\ 0, & \text{otherwise} \end{cases} \quad (2.10c)$$

with $l_i = \text{floor}(x_i/h)$ and $\alpha_i = x_i/h - l_i$.

It can be noticed that interpolation functions and spreading functions of the same order are analogous, except for the distribution factor $1/h$.

For a grid function spread over a 2-D domain, an additional spatial layer is needed, where $m_{i/o} = \text{floor}(y_{i/o}/h_y)$ and $\alpha_{y,i/o} = y_{i/o}/h_y - m_{i/o}$, where h_y is the spatial step in y direction. The distribution factor necessary for spreading will need to take into account both spacial dimensions so it will be $1/h^2$, with the subscripts indicating the dimension of the spatial step.

It must be noted that these are not the only possible interpolation/spreading functions. One can consider for example the case of an output being extracted as an averaged value over a multitude of grid points. In that case the interpolation function can take any mathematical distribution form (like a normal distribution

for example) as long as it is normalized, i.e. its integral is equal to 1. Such an interpolation function will have a dual spreading function with its corresponding distribution factor. Its physical interpretation can be spreading a load over multiple points, as would be the case of a connection between two elements that is not just punctual.

2.3 Identities

Two identities which are of use in the current project are:

$$\delta_{tt}u_{l,m}^n = \frac{2}{k} (\delta_t \cdot u_{l,m}^n - \delta_{t-} u_{l,m}^n), \quad (2.11)$$

and:

$$\langle f, J \rangle_{\mathcal{D}} = If, \quad (2.12)$$

with $\langle \cdot, \cdot \rangle$ being an L^2 inner product over the appropriate domain of the functions, \mathcal{D} . f is a grid function and I and J must be dual interpolation and spreading functions, defined over \mathcal{D} .

Chapter 3

Friction Models

Friction has a very important place in sound synthesis as it is the excitation mechanism for a variety of instruments. A bow is typically used to excite string instruments like the ones in the violin family, the Arabic rebab family or the traditional Chinese Erhu [32], [5]. Such a bow can be used by more experimental musicians to excite other instruments, like the cymbals of a drum set. Another plate-type bowed instrument is the musical saw.

The friction does not necessarily need to be triggered by a bow. In the case of the friction drum, a stick is usually fixed through the center of a drum membrane which is excited by either rubbing the stick with a cloth or moving the stick up and down against the membrane [1].

Models of such friction excitation need to be able to capture the non-linear relationship between the interacting elements. These can be categorized in either static or dynamic models. In the former, the frictional force is only dependent on the relative sliding velocity between the two elements, while in the latter the force is described through a differential equation. Two such models, which will be used for simulating the instruments taken up in this project are presented in this chapter.

3.1 Friction Mechanisms

An overview of friction contributions is given in this section [17].

3.1.1 Coulomb Friction

If one considers two elements in contact with a normal force of f_N [N], then the Coulomb friction is proportional to this normal force by a factor μ_C [-]:

$$f_C = \mu_C f_N, \quad (3.1)$$

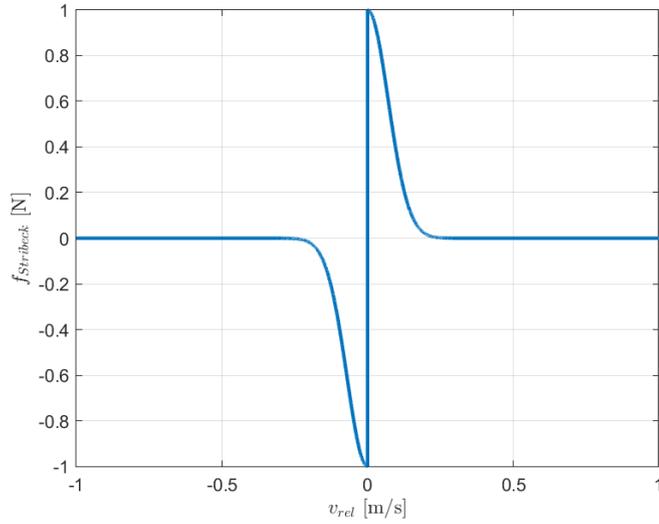


Figure 3.1: Stribeck force, $f_{Stribeck}$ as a function of the relative velocity v_{rel} with a normal force $f_N = 2$ [N], $\mu_C = 0.3$, $\mu_S = 0.8$ and a Stribeck velocity $v_S = 0.1$ [m/s].

In this case the resulting friction force is only dependent on the normal force between the two interacting elements.

3.1.2 Viscous Friction

The viscous friction is proportional to the relative velocity v_{rel} [m/s] between the two elements:

$$f_V = s_V v_{rel}, \quad (3.2)$$

where s_V [kg/s] is the viscous coefficient.

3.1.3 Stribeck Friction

Stribeck friction, $f_{Stribeck}$ [N], introduces the velocity dependence property, while capturing the Stribeck effect, i.e. the observed dip of the friction force at low relative velocities compared to the velocity at stick ($v_{rel} = 0$). One function that can capture this is:

$$f_{Stribeck} = \text{sgn}(v_{rel}) (f_S - f_C) e^{-(v_{rel}/v_S)^2}, \quad (3.3)$$

where $f_S = \mu_S f_N$ [N] is a static friction (stiction) proportional to the normal force f_N by a coefficient μ_S [-] and is the friction force at zero relative velocity. Furthermore, v_S [m/s] represents the Stribeck velocity and gives the curve of the exponential decay. Figure 3.1 shows a plot of the Stribeck friction for a given choice of parameters.

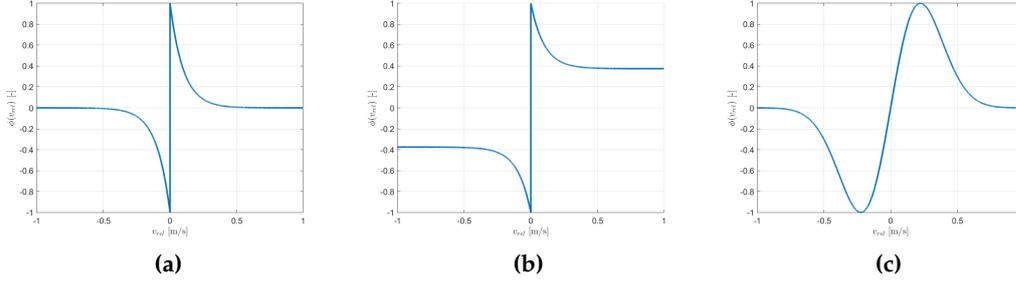


Figure 3.2: Various non-linear friction characteristics with parameters $a = 10$, $\epsilon = 0.375$ (a) $\phi(v_{rel}) = \text{sgn}(v_{rel})e^{-a|v_{rel}|}$ (b) $\phi(v_{rel}) = \text{sgn}(v_{rel}) \left(\epsilon - (1 - \epsilon)e^{-a|v_{rel}|} \right)$ (c) $\phi(v_{rel}) = \sqrt{2a}v_{rel}e^{-av_{rel}^2+1/2}$.

3.2 Static Friction Model

Many features relevant to the friction mechanism used as excitation in music instruments can be captured by *static* models where the frictional force acting at the contact point between two elements is a function of their relative velocity. This is in a way a type of non-linear viscous friction which may be written as:

$$f_{friction} = f_N \phi(v_{rel}), \quad (3.4)$$

with, $f_{friction}$ being the friction force, f_N being the friction normal force, v_{rel} being the relative velocity between the two elements and $\phi(v_{rel})$ is a given non-linear characteristic which aims to model the stick/slip regime of friction.

In [2] a number of possible friction characteristics are proposed which are illustrated in Figure 3.2. While the first 2 exhibit a discontinuity at $v_{rel} = 0$, the third one is continuous throughout the entire domain and is therefore easier to work with in numerical methods.

Looking particularly at the friction characteristic used in Figure 3.2b:

$$\phi(v_{rel}) = \text{sgn}(v_{rel}) \left(\epsilon - (1 - \epsilon)e^{-a|v_{rel}|} \right), \quad (3.5)$$

one can see some resemblance to the friction contributions described in Section 3.1. Starting from this, the following friction model is proposed:

$$f_{friction} = f_N \text{sgn}(v_{rel}) (\mu_C + (\mu_S - \mu_C)e^{-(v_{rel}/v_S)^2}) + s_V v_{rel}, \quad (3.6)$$

Expanding Equation (3.6), one can see the different friction components, while

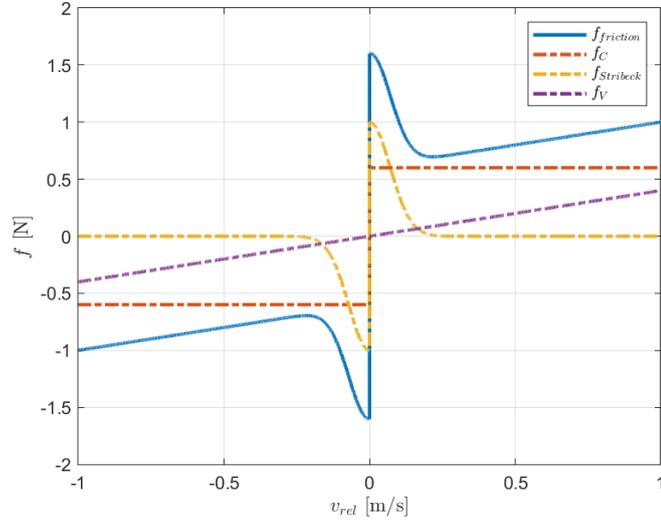


Figure 3.3: Friction force resulting from the static friction model proposed in Equation 3.7. Parameters used: $f_N = 2$ [N], $\mu_C = 0.3$ [-], $\mu_S = 0.8$ [-], $v_S = 0.1$ [m/s] and $s_V = 0.4$ [kg/s].

the entire model is illustrated in Figure 3.3:

$$f_{friction} = f_C + f_{Stribeck} + f_V, \quad (3.7a)$$

$$f_C = \mu_C f_N \text{sgn}(v_{rel}), \quad (3.7b)$$

$$f_{Stribeck} = f_N \text{sgn}(v_{rel}) (\mu_C + (\mu_S - \mu_C) e^{-(v_{rel}/v_S)^2}), \quad (3.7c)$$

$$f_V = s_V v_{rel}, \quad (3.7d)$$

The reason why a model that includes all these different friction mechanisms can capture the stick/slip regime of friction is particularly due to the Stribeck friction model. Imagining the case of a bowed string, the bow will catch on to the string and "pick it up" when increasing normal force f_N is applied. The resulting frictional force transferred to the string will induce transverse displacement at the bowing location. When the bow is then moved, the relative velocity between the two elements increases and the resulting friction force goes down. The more it does this, the bow releases the string and it lets it vibrate naturally. When this vibration syncs with the velocity of the bow, the frictional force increases again and the bow sticks to the string once more, and the cycle resets.

3.3 Dynamic Friction Model

Experimental results presented by Woodhouse and Smith [30] with respect to bowed strings show that the measured frictional force vs. the relative velocity between the bow and the string (sliding velocity) exhibits hysteresis loops, meaning that the resulting force is not a function of this parameter alone (relative velocity). A dependence on some additional variables is needed.

In order to capture dynamic friction behaviors such as presliding displacement, frictional memory (also known as friction lag or hysteresis) and stick-slide-slip motion, dynamic friction models have been developed by several authors, [18], [4], [8] and [6]. These models describe the dependence of the frictional force on the relative velocity between the interacting elements via a differential equation.

In this project, one such dynamic model is considered, particularly the elasto-plastic model proposed in [6]. This model has been successfully implemented by Serafin et al. in [28] and Willemsen et al. in [36] in the context of a bowed string using a digital waveguide model and a FDTD method respectively (implemented in real-time).

3.4 Elasto-Plastic Model

The elasto-plastic friction model considers the contact between the two interacting elements as being highly irregular at the microscopic level. It follows that the actual contact surface is not the same as the overlapping surface. In order to model this, the contact is considered to be in the form of a large ensemble of bristles, each acting as a damped stiff spring, which can break after a certain displacement. The behavior resulting from the entire ensemble of bristles can be investigated with regards to the average displacement of all the bristles z [m]. Initially when the bristles start to displace, each produces an elastic reaction force, i.e. sticking regime or pre-sliding. After some time however, the break-away displacement, z_{ba} [m] is reached by some bristles, which start to break, while others remain in the elastic regime. This represents the elasto-plastic regime or sliding. When all the bristles have reached the breaking point, a completely plastic regime is entered, known as slipping, where the average bristle displacement is given by its steady-state value z_{ss} [m], which is a function of the relative velocity v_{rel} . Steady state implies that the average bristle displacement does not change in time, i.e. $\dot{z} = 0$. This mechanism is illustrated in Figure 3.4 for the case of a bowed string. Taken from [36].

The frictional force can be expressed as a combination of friction mechanisms as given in:

$$f_{friction}(v_{rel}, z) = s_0 z + s_1 \dot{z} + s_2 v_{rel} + s_3 w, \quad (3.8)$$

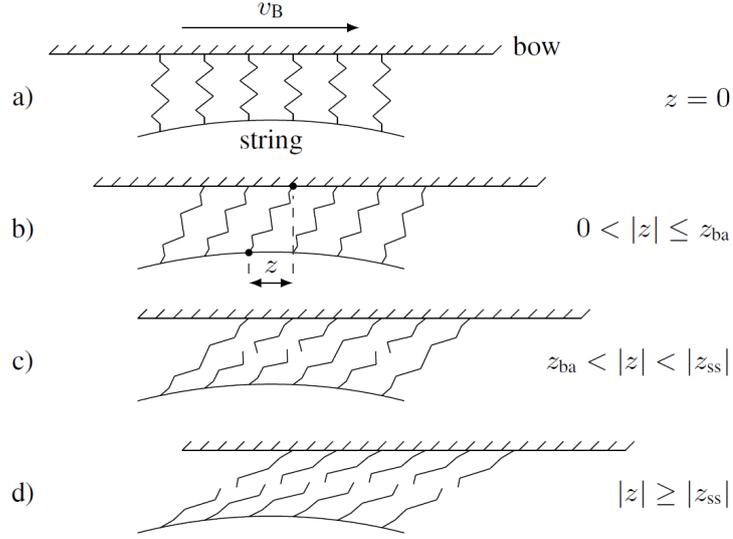


Figure 3.4: Illustration of the microscopic bristle displacements between the two interacting elements during friction. Taken from [36]. a) Initial state. b) Elastic sticking regime. c) Elasto-plastic sliding regime. d) Plastic slipping regime.

where s_0 is the average bristle stiffness [N/m], s_1 is the damping coefficient of the bristles [kg/s] and s_2 is the viscous friction [kg/s]. s_3 [N] is a force coefficient proportional to the normal bowing force f_N (which is an external input and can vary over time) scaled with a pseudorandom function $w(t) \in [-1, 1]$ and is used to add noise to the total frictional force, as per [27].

The time derivative of the average bristle displacement to, \dot{z} [m/s] contributes to the total friction and its presence turns Equation (3.8) into a differential equation. Its value is given by:

$$\dot{z} = r(v_{rel}, z) = v \left[1 - \alpha(z, v_{rel}) \frac{z}{z_{ss}(v_{rel})} \right]. \quad (3.9)$$

Equation (3.9) introduces an important parameter of the elasto-friction model which controls the transition between the various regimes of friction. It is the adhesion map $\alpha(v_{rel}, z)$ given by:

$$\alpha(v_{rel}, z) = \begin{cases} 0 & |z| \leq z_{ba}, \text{sgn}(v_{rel}) = \text{sgn}(z) \\ \alpha_m(v_{rel}, z) & z_{ba} < |z| < |z_{ss}(v_{rel})|, \text{sgn}(v_{rel}) = \text{sgn}(z) \\ 1 & |z| \geq |z_{ss}(v_{rel})|, \text{sgn}(v_{rel}) = \text{sgn}(z) \\ 0 & \text{sgn}(v_{rel}) \neq \text{sgn}(z) \end{cases} \quad (3.10)$$

Looking again at Equation 3.9 and evaluating the effect of $\alpha(v_{rel}, z)$, one can see that in the case where the average bristle displacement, z is less than the break-away displacement z_{ba} and $\alpha(v_{rel}, z) = 0$, then $\dot{z} = v_{rel}$, which implies a sticking behavior.

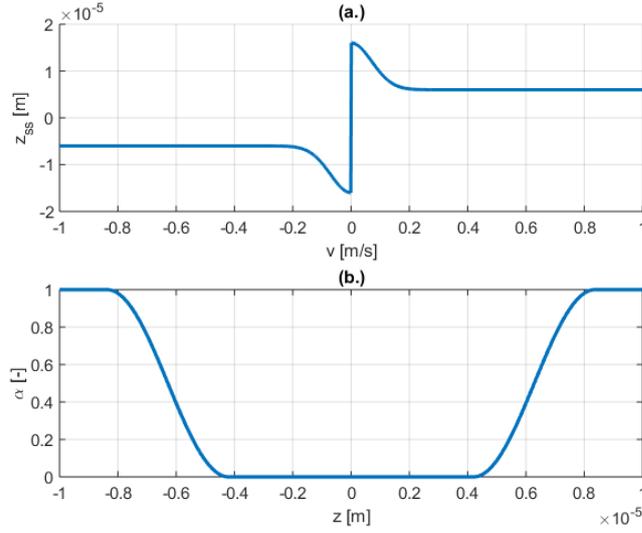


Figure 3.5: (a.) Steady-state bristle displacement $z_{ss}(v)$ for a constant normal force f_N . (b.) A plot of the adhesion map $\alpha(v, z)$ plotted against z when the signs of v and z are the same.

As z_{ba} is surpassed, and some bristles have reached the breaking point, the elasto-plastic regime of sliding is entered. Here, z will be a proportion, governed by $\alpha_m(v, z)$, of the steady-state bristle displacement $z_{ss}(v)$, with:

$$\alpha_m = \frac{1}{2} \left[1 + \operatorname{sgn}(z) \sin \left(\pi \frac{z - \operatorname{sgn}(z) \frac{1}{2} (|z_{ss}(v_{rel})| + z_{ba})}{|z_{ss}(v_{rel})| - z_{ba}} \right) \right]. \quad (3.11)$$

When the average bristle displacement surpasses the break-away displacement and the plastic or slipping regime is entered, $\alpha(v, z) = 1$ and being a steady-state regime, i.e. $\dot{z} = 0$, it follows that $z = z_{ss}(v_{rel})$, with:

$$z_{ss}(v_{rel}) = \frac{\operatorname{sgn}(v_{rel})}{s_0} \left[f_C + (f_S - f_C) e^{-(v_{rel}/v_S)^2} \right]. \quad (3.12)$$

Figure 3.5 illustrates the steady-state bristle displacement z_{ss} and the adhesion map $\alpha(v_{rel}, z)$ when the signs of v_{rel} and z are the same.

Considering the case of the steady-state solution alone, i.e. introducing Equation (3.12) and the condition $\dot{z} = 0$ in Equation (3.8), the frictional force is then given by:

$$f_{friction}(v_{rel}) = \operatorname{sgn}(v_{rel}) \left[f_C + (f_S - f_C) e^{-(v_{rel}/v_S)^2} \right] + s_2 v_{rel} + s_3 w, \quad (3.13)$$

which is only a function of the relative velocity v_{rel} and is in fact the same as the static model proposed in Equation (3.6), except for the random force term $s_3 w$ (as

$f_C = f_N \mu_C$, $f_S = f_N \mu_S$ and s_2 and s_V can be the same viscous friction coefficient). Therefore, when steady-state is reached, the elasto-plastic dynamic model is the same as the proposed static model, with the difference between the two occurring at the initial sticking regime and the sliding regime at small bristle displacements.

Chapter 4

Violino Arpa

The sound of musical instruments like a violin is more complex than that of a bowed string alone as the body of the instrument interacts with the string and also vibrates along with the air inside its hollow area. In this Chapter an attempt at such a more complex simulation is carried out by means of FDTD methods, with the aim of reproducing the possible sound of the Violino Arpa.

The work presented here has been included in a collaborative project at Aalborg University, where the experience of playing the Violino has been recreated using augmented reality. This resulted in a paper which has been submitted to the Sound and Music Computing Conference 2021 (SMC2021) and is currently under peer review. The paper can be found in Appendix A.

4.1 Continuous Model

The physical system used to model the Violino Arpa consists of a bowed stiff string rigidly connected to a resonant plate, that captures the unusual shape of the real instrument. Throughout this section the PDEs describing each element will be presented.

4.1.1 Bowed Stiff String

The transverse displacement of a bowed stiff string of length L_s [m] is noted as $u(\chi, t)$ [m], with $\chi \in [0, L_s]$ [m] being a coordinate along the length of the string. It can be described by the following PDE:

$$\begin{aligned} \rho_s A_s \partial_t^2 u = & T_s \partial_\chi^2 u - E_s I_s \partial_\chi^4 u - 2\rho_s A_s \sigma_{0,s} u_t + \\ & 2\rho_s A_s \sigma_{1,s} \partial_t \partial_\chi^2 u - \delta(\chi - \chi_B) f_B + \delta(\chi - \chi_C) f_C, \end{aligned} \quad (4.1)$$

where ρ_s [kg/m³] is the material density of the string with a cross-sectional area $A_s = \pi r^2$ [m²] given by the radius r [m]. The tension of the string is T_s [N]. It has

a Young's Modulus E_s [Pa] and moment of inertia $I_s = \pi r^4/4$ [m⁴].

The coefficients $\sigma_{0,s}$ [s⁻¹] and $\sigma_{1,s}$ [m²/s] introduce frequency-dependent and frequency-independent damping respectively in the system, while f_B [N] is the bowing force resulting from the considered friction interaction model, with options described in Chapter 3. This force is applied at a single point χ_B [m] along the string, as specified by the Dirac function $\delta(\chi - \chi_B)$ [1/m].

Furthermore, f_C is the connection force resulting from the rigid connection to the resonant plate, applied at a location χ_C [m], given by $\delta(\chi - \chi_C)$ [1/m].

Simply supported boundary conditions are considered at the edges of the stiff string, i.e. the following condition needs to be satisfied at $x = 0$ and $x = L_s$:

$$u(0) = \partial_\chi^2 u(0) = 0, \quad (4.2a)$$

$$u(L_s) = \partial_\chi^2 u(L_s) = 0. \quad (4.2b)$$

4.1.2 Plate

A 2-D plate of length $L_{p,x}$ [m] in x direction and $L_{p,y}$ [m] in y direction can be modelled as a Kirchhoff thin plate with losses, where the transverse displacement, $w(x, y, t)$ [m] is described by the following PDE:

$$\begin{aligned} \rho_p H_p \partial_t^2 w = & -D_p \Delta \Delta w - 2\rho_p H_p \sigma_{0,p} \partial_t w + \\ & 2\rho_p H_p \sigma_{1,p} \Delta \partial_t w - \delta(x - x_C, y - y_C) f_C. \end{aligned} \quad (4.3)$$

Here the density of the plate material is given by ρ_p [kg/m³], while $D_p = E_p H_p^3 / 12(1 - \nu_p^2)$ [kg·m²·s⁻²] is the bending stiffness of the plate and results from the Young's Modulus of the material E_p [Pa], the plate thickness H_p [m] and the dimensionless Poisson's Ratio ν_p [-]. Loss coefficients $\sigma_{0,p}$ [s⁻¹] and $\sigma_{1,p}$ [m²/s] are introduced similar to the string. Finally, the connection force at is applied at a single connection point (x_C, y_C) by means of a 2-D Dirac delta function $\delta(x - x_C, y - y_C)$. Notice also the use of the 2-D Laplacian operator defined as:

$$\Delta \triangleq \partial_x^2 + \partial_y^2. \quad (4.4)$$

Clamped boundary conditions are considered at the edges of the plate so that:

$$w = \mathbf{n} \cdot \nabla w = 0, \quad (4.5)$$

where ∇w is the gradient of w and where \mathbf{n} indicates a normal at the boundary.

4.1.3 Connection

As briefly mentioned, the two elements are assumed to be rigidly connected. This means that the connection force f_C is the same for both elements, but with opposite sign (action/reaction). Furthermore, it means that the relative displacement

between the two elements at the connection points is zero, i.e.:

$$\eta(t) = u(\chi_C, t) - w(x_C, y_C, t) = 0. \quad (4.6)$$

4.1.4 Bowing Force

For the resulting bowing force, the two cases presented in Chapter 3 will be considered: the proposed static friction model given in Equation (3.13) and the elastoplastic dynamic model given in Equation (3.8).

Additionally, the relative velocity between the bow and the string $v_{rel}(t)$ [m/s] can be found using the following equation:

$$v_{rel} = \partial_t u(\chi_B) - v_B, \quad (4.7)$$

with $u(\chi_B)$ being the string displacement at the bowing position and v_B being the velocity of the bow, which is an external input.

4.1.5 Complete System

The complete system for the Violino Arpa can be written as:

$$\left\{ \begin{array}{l} \rho_s A_s \partial_t^2 u = T_s \partial_\chi^2 u - E_s I_s \partial_\chi^4 u - 2\rho_s A_s \sigma_{0,s} u_t + \\ \quad 2\rho_s A_s \sigma_{1,s} \partial_t \partial_\chi^2 u - \delta(\chi - \chi_B) f_B + \delta(\chi - \chi_C) f_C \\ \rho_p H_p \partial_t^2 w = -D_p \Delta \Delta w - 2\rho_p H_p \sigma_{0,p} \partial_t w + \\ \quad 2\rho_p H_p \sigma_{1,p} \Delta \partial_t w - \delta(x - x_C, y - y_C) f_C \\ \eta = u(\chi_C) - w(x_C, y_C) = 0. \end{array} \right. \quad (4.8a)$$

$$\left\{ \begin{array}{l} \rho_p H_p \partial_t^2 w = -D_p \Delta \Delta w - 2\rho_p H_p \sigma_{0,p} \partial_t w + \\ \quad 2\rho_p H_p \sigma_{1,p} \Delta \partial_t w - \delta(x - x_C, y - y_C) f_C \\ \eta = u(\chi_C) - w(x_C, y_C) = 0. \end{array} \right. \quad (4.8b)$$

$$\left\{ \begin{array}{l} \rho_p H_p \partial_t^2 w = -D_p \Delta \Delta w - 2\rho_p H_p \sigma_{0,p} \partial_t w + \\ \quad 2\rho_p H_p \sigma_{1,p} \Delta \partial_t w - \delta(x - x_C, y - y_C) f_C \\ \eta = u(\chi_C) - w(x_C, y_C) = 0. \end{array} \right. \quad (4.8c)$$

4.2 Discretization

The system given in Equation (4.8) can be discretized using FDTD methods described in Chapter 2.

The time is discretized in samples as $t = nk$, with $k = 1/f_S$ [s] being the time step, which follows from a desired sampling frequency f_S [Hz] and $n \in \mathbb{N}$ being an integer counting the number of time steps. Next, the three spatial dimensions used in the system are discretized. First, $\chi = ph_s$, where h_s is the spatial grid step and $p \in [0, N_\chi]$ is an integer counting the grid points in space. With the string being of length L_s it follows that $N_\chi = \text{floor}(L_s/h_s)$ is the total number of intervals the string is divided in. Similarly for the dimensions of the plate, $x = lh_p$ and $y = mh_p$, where h_p is the spatial step of discretization for the plate, considered equal in both directions. For a rectangular grid of lengths $L_{p,x}$ and $L_{p,y}$, it follows that $l \in [0, N_x]$ and $m \in [0, N_y]$ are both integers counting the spatial steps in the two directions. Then, $N_x = \text{floor}(L_{p,x}/h_p)$ and $N_y = \text{floor}(L_{p,y}/h_p)$.

One could assume that a more accurate solution can be obtained with the numerical analysis if the spatial step is as small as possible, as one can converge to a more accurate representation of the models themselves. However, this is not the case as the numerical schemes can become unstable if the spatial grid size is too small. Stability conditions can be found however and although their derivation is not a part of the current project, they will be given for each element in the system. Simulations with discretizations that are closer to the stability conditions provide more accurate results.

Having these discretizations in place, the grid functions u_p^n and $w_{l,m}^n$ are approximations to the continuous time transverse displacements of the string and the plate respectively. Furthermore, the derivatives found in the continuous time PDEs from the previous section can then be approximated using the operators presented in Chapter 2. The resulting FDSs are presented in the following.

4.2.1 Bowed Stiff String

The equation describing the transverse displacement of the bowed stiff string, given in Equation (4.1) can be discretized in the following way:

$$\begin{aligned} \delta_{tt}u_p^n = & c_s^2 \delta_{\chi\chi}u_p^n - \kappa_s^2 \delta_{\chi\chi\chi\chi}u_p^n - 2\sigma_{0,s} \delta_t \cdot u_p^n + \\ & 2\sigma_{1,s} \delta_{t-} \delta_{\chi\chi}u_p^n - \frac{J_B f_B^n}{\rho_s A_s} + \frac{J_{C,s} f_C^n}{\rho_s A_s}, \end{aligned} \quad (4.9)$$

where the parameters $c_s = \sqrt{T_s/\rho_s A_s}$ [m/s] and $\kappa_s = \sqrt{E_s I_s/\rho_s A_s}$ [m²/s] are introduced.

Additionally, J_B and $J_{C,s}$ are third order spreading operators (cubic) used to add the discrete bowing force f_B^n and the discrete connection force f_C^n at the corresponding locations, even if they lie in-between spatial grid points. They are of the form given in Equation (2.10c).

The simply supported boundary conditions are discretized as such:

$$u_0^n = \delta_{\chi\chi}u_0^n = 0, \quad (4.10a)$$

$$u_{N_\chi}^n = \delta_{\chi\chi}u_{N_\chi}^n = 0. \quad (4.10b)$$

These operators can be expanded and the equations given in Equation (4.10) are evaluated at these boundary points in order to find values values of the grid function, u^n , at locations outside the grid, i.e. $p = -1$ and $p = N_{\chi+1}$ in terms of available values. These are needed in order to be able to apply update equations at the edge points, $p = 1$ and $p = N_\chi - 1$.

A stability condition for the stiff string is given by, as per [36] and [2]:

$$h_s \geq h_{s,min} = \sqrt{\frac{c_s^2 k^2 + 4\sigma_{1,s} k + \sqrt{(c_s^2 k^2 + 4\sigma_{1,s} k)^2 + 16\kappa_s^2 k^2}}{2}}, \quad (4.11)$$

4.2.2 Plate

The discretized version of the PDE governing the transverse displacement of the plate, given in Equation (4.3) is:

$$\begin{aligned} \delta_{tt}w_{l,m}^n = & -\kappa_p^2\delta_{\Delta\Delta}w_{l,m}^n - 2\sigma_{0,p}\delta_t.w_{l,m}^n + \\ & 2\sigma_{1,p}\delta_{t-\delta_\Delta}w_{l,m}^n - \frac{J_{C,p}f_C^n}{\rho_p H_p}. \end{aligned} \quad (4.12)$$

where $\kappa_p = \sqrt{D_p/\rho_p H_p}$ [m^4/s^2] and $J_{C,p}$ is a first order 2-D spreading operator for the discretized connection force f_C^n , given by:

$$J_{C,p} = \frac{1}{h_p^2} \begin{cases} (1 - \alpha_{x_C})(1 - \alpha_{y_C}) & l = l_C, m = m_C \\ (1 - \alpha_{x_C})\alpha_{y_C} & l = l_C, m = m_C + 1 \\ \alpha_{x_C}(1 - \alpha_{y_C}) & l = l_C + 1, m = m_C \\ \alpha_{x_C}\alpha_{y_C} & l = l_C + 1, m = m_C + 1 \\ 0 & \text{else,} \end{cases} \quad (4.13)$$

with $l_C = \text{floor}(x_C/h_p)$, $m_C = \text{floor}(y_C/h_p)$, $\alpha_{x_C} = x_C/h_p - l_C$ and $\alpha_{y_C} = y_C/h_p - m_C$.

So far the plate has been considered as rectangular, with lengths L_x and L_y , but it is the desire to model the plate as close as possible to the shape of the Violino Arpa. This is possible by "sculpting" the shape of the violin from this original rectangular grid using a staircase approximation, [7], as long as boundary conditions are satisfied at the edges of the plate. Luckily, for the case of clamped conditions, the only thing needed is to make sure to have two neighboring points in both x and y directions that can be set to zero, i.e. the shape of the violin's resonant plate needs to be bounded by two rows and two columns of additional grid points fixed to zero (clamped). This follows from the expansion of the $\delta_{\Delta\Delta}$ operator.

The image of the real instrument shown in Figure 1.1 is converted to gray scale and overlapped with a rectangular grid scaled accordingly to fit the real dimensions of the violin, and the overlapping area is used to select the grid points of interest for the plate. All the other grid points are set to zero. Figure 4.1 shows a picture of the resulting grid together with the string while indicating the connection point (the bridge location on the violin).

A stability condition for the plate is given by [2]:

$$h_p \geq h_{p,min} = 2\sqrt{k(\sigma_{1,p} + \sqrt{\kappa_p^2 + \sigma_{1,p}^2})}, \quad (4.14)$$

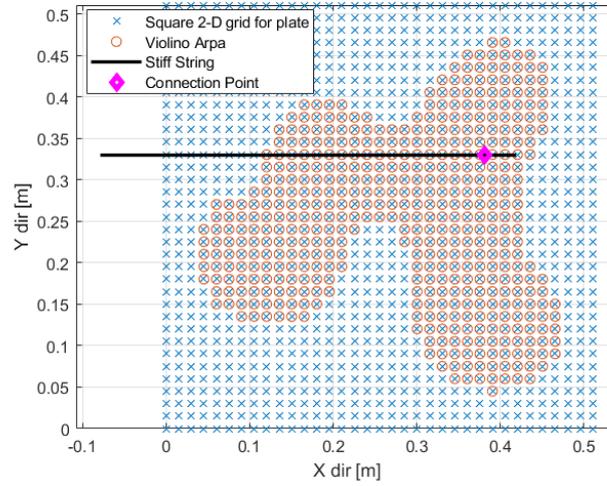


Figure 4.1: An example of the selected points for the Violino Arpa starting from a grid of 34 intervals in both directions, i.e. $N_x = N_y = 34$. The position stiff string and its connection point to the plate is also illustrated.

4.2.3 Connection

The connection condition in Equation (4.6) can be discretized as:

$$\eta^n = \langle u^n, J_{C,s} \rangle_{D_s} - \langle w^n, J_{C,p} \rangle_{D_p} = 0. \quad (4.15)$$

with $\langle \cdot, \cdot \rangle$ being an L^2 inner product over the appropriate domain (1D for the string and 2D for the plate). Basically, $\langle u^n, J_{C,s} \rangle$ gives the value of the grid point u^n at the connection location χ_C and likewise for the second term.

Using the identity from Equation (2.11), then the condition in Equation (4.15) can be rewritten as:

$$\eta^n = I_{C,s}(\chi_C)u_p^n - I_{C,p}(x_C, y_C)w_p^n = 0, \quad (4.16)$$

with $I_{C,s}$ and $I_{C,p}$ being the dual interpolation operators to the spreading operators $J_{C,s}$ and $J_{C,p}$.

Furthermore, if the condition in Equation (4.16) is true at all samples n , therefore it will also be true for the "next" sample $n + 1$, i.e.:

$$\eta^{n+1} = I_{C,s}(\chi_C)u_p^{n+1} - I_{C,p}(x_C, y_C)w_p^{n+1} = 0. \quad (4.17)$$

The values of u_p^{n+1} and w_p^{n+1} can be expressed only in terms of "previous" samples by expanding the finite difference operators in Equation (4.9) and Equation (4.12). Inserting those expressions in Equation (4.17), an equation for calculating the connection force f_C^n can be found:

$$f_c^n = -\frac{b^n - \frac{k^2 f_B^n I_{C,s} I_B}{(1+\sigma_{0,s})\rho_s A_s}}{k^2 \left[\frac{I_{C,s} I_{C,s}}{(1+\sigma_{0,s})\rho_s A_s} + \frac{I_{C,p} I_{C,p}}{(1+\sigma_{0,p})\rho_p H_p} \right]}, \quad (4.18)$$

with:

$$\begin{aligned} b^n = & \frac{1}{1 + \sigma_{0,s}k} [c_s^2 k^2 I_{C,s} \delta_{\lambda\lambda} u_p^n - \kappa_s^2 k^2 I_{C,s} \delta_{\lambda\lambda\lambda\lambda} u_p^n + 2\sigma_{1,s}k(I_{C,s} \delta_{\lambda\lambda} u_p^n - \\ & I_{C,s} \delta_{\lambda\lambda} u_p^{n-1}) + 2I_{C,s} u_p^n - (1 - \sigma_{0,s}k)I_{C,s} u_p^{n-1}] - \\ & \frac{1}{1 + \sigma_{0,p}k} [-\kappa_p^2 k^2 I_{C,p} \delta_{\Delta\Delta} w_{l,m}^n + 2\sigma_{1,p}k(I_{C,p} \delta_{\Delta\Delta} w_{l,m}^n - \\ & I_{C,p} \delta_{\Delta\Delta} w_{l,m}^{n-1}) + 2I_{C,p} w_{l,m}^n - (1 - \sigma_{0,p}k)I_{C,p} w_{l,m}^{n-1}] \end{aligned} \quad (4.19)$$

Assuming that the the stiff string cannot be bowed at the connection point between the two elements, the term $I_{C,s} I_B$, i.e. the interpolation operator for the string connection applied to the spreading operator for the bowing force, will be zero and the term subtracted from b^n in the denominator of Equation (4.18) can be removed.

4.2.4 Bowing Force

The discrete counterpart of static friction bowing force will be:

$$f_{B,st}^n(v_{rel}^n) = \text{sgn}(v_{rel}^n) f_N^n \left[\mu_C + (\mu_S - \mu_C) e^{-(v_{rel}^n/v_S)^2} \right] + s_2 v_{rel}^n + s_3 w^n, \quad (4.20)$$

For the dynamic elasto-plastic friction model, the discretization is:

$$f_{B,dyn}^n(v_{rel}^n, z^n) = s_0 z^n + s_1 r^n + s_2 v_{rel}^n + s_3 w^n, \quad \text{with:} \quad (4.21)$$

$$r^n(v_{rel}^n, z^n) = v_{rel}^n \left[1 - \alpha(z^n, v_{rel}^n) \frac{z^n}{z_{ss}(v_{rel}^n)} \right]. \quad (4.22)$$

Additionally, the relative velocity between the bow and the membrane described in Equation (4.7) will be:

$$v_{rel}^n = I_B(x_B^n) \delta_t u_p^n - v_B^n. \quad (4.23)$$

4.2.5 Solving the Complete System

In order to calculate the update values for the grid functions: u_p^{n+1} and $w_{l,m}^{n+1}$, a number of unknown variables must first be determined, depending on which friction model is chosen. For the case of the elasto-plastic dynamic model, values for both the relative velocity between the bow and the string, v_{rel}^n as well as for the average bristle displacement z^n are needed in order to calculate the resulting

bowing force $f_{B,dyn}^n$ at the "current" time step. For the static model, only v_{rel}^n is needed.

The solution to this problem is presented first for the dynamic friction model as it is the more complicated of the two and the solution for the static friction model is a reduced version of the former.

As there are two unknowns, it follows that a system of two equations dependent on v_{rel}^n and z^n is needed for a possible solution. Even if the equations are non-linear, solutions can be found using numerical analysis methods like a multivariate Newton-Raphson.

The first equation $g_1(v_{rel}^n, z^n) = 0$ can be obtained by making use of the identity given in Equation (2.11) and introducing it together with Equation (4.23) in Equation (4.9). This results in:

$$g_1(v^n, z^n) = I_B J_B \frac{f_{B,dyn}^n(v^n, z^n)}{\rho_s A_s} + \left(\frac{2}{k} + 2\sigma_{0,s} \right) v_{rel}^n + q^n = 0, \quad (4.24)$$

with

$$q^n = -\frac{2}{k} \delta_{t-} I_B u_p^n + 2\sigma_{0,s} v_B^n + \frac{2}{k} v_B^n - c_s^2 I_B \delta_{\chi\chi} u_p^n + \kappa_s^2 I_B \delta_{\chi\chi\chi\chi} u_p^n - 2\sigma_{1,s} \delta_{t-} I_B \delta_{\chi\chi} u_p^n.$$

The second equation $g_2(v_{rel}^n, z^n)$ can be found by applying the trapezoidal rule to z^n , [2], resulting in:

$$\begin{aligned} g_2(v^n, z^n) &= r^n - a^n = 0, \quad \text{with} \\ a^n &= (\mu_{t-})^{-1} \delta_{t-} z^n, \quad \text{i.e.:} \\ \frac{1}{2} (a^n + a^{n-1}) &= \frac{1}{k} (z^n - z^{n-1}) \end{aligned} \quad (4.25)$$

With these equations in place, the following iteration is then used to calculate the unknown values v^n, z^n :

$$\begin{bmatrix} v_{rel}^{n,(i+1)} \\ z^{n,(i+1)} \end{bmatrix} = \begin{bmatrix} v_{rel}^{n,(i)} \\ z^{n,(i)} \end{bmatrix} - \begin{bmatrix} \frac{\partial g_1}{\partial v_{rel}} & \frac{\partial g_1}{\partial z} \\ \frac{\partial g_2}{\partial v_{rel}} & \frac{\partial g_2}{\partial z} \end{bmatrix}^{-1} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \quad (4.26)$$

where i is the iteration number. The threshold for convergence is set at 10^{-7} , with a maximum number of iterations of 99.

Once the values of v_{rel}^n and z^n are known, they are inserted in Equation (4.21) and the resulting bowing force can be found $f_{B,dyn}^n(v_{rel}^n, z^n)$. Finally, all the information necessary for calculating the update values of the grid functions u_p^{n+1} and $w_{l,m}^{n+1}$ is available and can be done so by expanding the finite difference operators in Equation (4.9) and Equation (4.12).

For the case of the static friction model, only the first equation $g_1(v_{rel})$ is needed, where $f_{B,dyn}^n(v_{rel}^n, z^n)$ is replaced by $f_{B,st}^n(v_{rel}^n)$. The Newton-Raphson scheme is then adjusted to be an iteration over a single unknown variable, v_{rel}^n :

$$v_{rel,(i+1)}^n = v_{rel,(i)}^n - \frac{g_1}{\frac{dg_1}{dv_{rel}}} \quad (4.27)$$

4.3 Implementation

As a main goal, the finite difference scheme presented in Section 4.2 was implemented as a real-time audio application in C++ using the JUCE framework [26]. A demonstrative video is available at [24]. In order to reach this point, the implementation was first carried out in Matlab [12] where it could be investigated and quality checked in more detail.

The parameters used can be found in Table 4.1 and have been chosen based on the work of Serafin [27], Willemsen et al. [36] and again Willemsen et al. [37]. Some of the parameters are fixed, while others can be changed in the real-time implementation. For these, bounds of their possible values are given. For offline simulations using the Matlab implementation where these values are fixed, they will be specified for each example. Additionally, if results of simulations with other parameters than the ones in Table 4.1 are presented, they will again be specified.

4.3.1 Prototype

A Matlab implementation of the Violino Arpa was developed as a first step in the project. This platform provides easier means to check the model and investigate if the results are in line with expectations, as the simulation is carried offline and access to all the variables in the model is available at all samples.

Before considering the complete Violino Arpa model, the elasto-plastic bowing model was checked against the open-source implementation provided by Willemsen et al. [36] at Github [33]. The Violino Arpa string alone, without the connection to the resonant plate, was bowed at a location of $0.25L_s$ and the transverse displacement was extracted at this bowing location and at $2/3L_s$. The string was tuned to a fundamental frequency of $f_0 = 440.0$ corresponding to the note A_4 . From this the wave speed can be found as: $c_s = 2f_0L_s$. As it can be seen, the results match perfectly. The spatial grid step $h_s = 0.011313$ was taken as close to the stability condition as possible resulting in $N_\chi = 44$ spatial intervals for discretization.

Having this check in place, the string is connected to the resonant plate and the results with the elasto-plastic dynamic friction model are investigated. Figure 4.3 shows the displacements of the system in the middle of the simulation. A total of $N_\chi = 30$ grid intervals are used for the discretization of the string and $N_x =$

Name	Symbol [unit]	Value
String		
Length	L_s [m]	0.498
Material Density	ρ_s [kg/m ³]	7850
Radius	r_s [m]	0.0005
Young's Modulus	E_s [Pa]	210 ¹¹
Fundamental Freq.	f_0 [Hz]	$100 \leq f_0 \leq 660$
Wave Speed	c_s [m/s]	$2f_0/L_s$
Freq. dep. loss	$\sigma_{0,s}$ [s ⁻¹]	$0 \leq \sigma_{0,s} \leq 2$
Freq. indep. loss	$\sigma_{1,s}$ [m ² /s]	$0 \leq \sigma_{1,s} \leq 0.5$
Grid spacing	h_s [m]	0.0166
Plate		
Length	L_x [m]	0.526
Width	L_y [m]	0.526
Material Density	ρ_p [kg/m ³]	50
Thickness	H_p [m]	0.01
Young's Modulus	E_p [Pa]	200000
Poisson's Ratio	ν_p [-]	0.3
Freq. dep. loss	$\sigma_{0,p}$ [s ⁻¹]	0
Freq. indep. loss	$\sigma_{1,p}$ [m ² /s]	$0 \leq \sigma_{1,p} \leq 0.1$
Grid spacing	h_p [m]	0.015
Bowing Model		
Coulomb Friction	μ_C [-]	0.3
Static Friction	μ_S [-]	0.8
Normal Force	f_N [N]	$0 \leq f_N \leq 20$
Bow Velocity	v_B [m/s]	$0 \leq v_B \leq 0.2$
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 [N]	0
Pseudorandom Fct.	w [-]	$-1 \leq w \leq 1$
Breakaway Disp.	z_{ba} [m]	$0.7f_C/s_0$
Other		
Sample Rate	f_S [Hz]	44100
Time Step	k [s]	$1/f_S$

Table 4.1: Parameter values used for the Violino Arpa simulations.

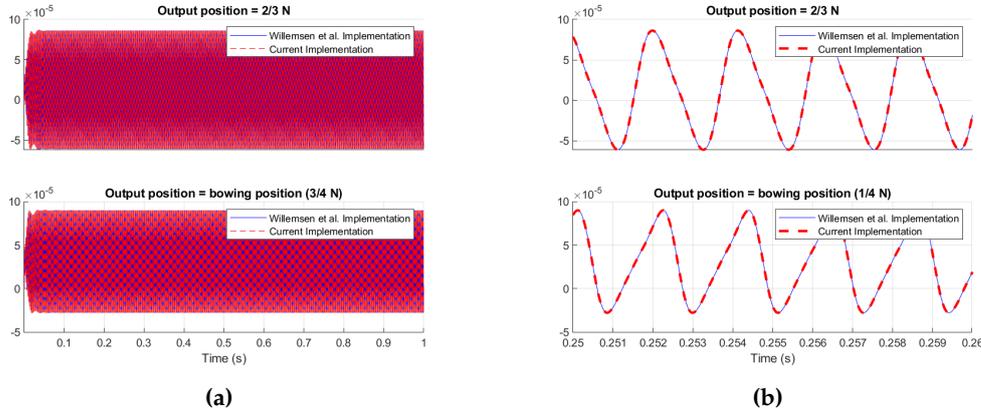


Figure 4.2: A comparison of a bowed string simulation with an elasto-plastic friction model using the open source implementation by Willemssen et al. [36] available at Github [33] and the implementation of the current project. String is tuned to $f_0 = 440.0$ [Hz] and is bowed at $\chi_B = 0.25L_s$ [m] with a normal force $f_N = 1$ [N] and bowing velocity $v_B = 0.1$ [m/s]. Furthermore, $\sigma_{0,s} = 0$ (a) Full 1 second simulation interval (b) Zoomed in interval.

$N_y = 35$ intervals discretize the square grid of the plate. This was chosen in order to be consistent with the real-time audio implementation. The string is tuned to $f_0 = 440.0$ and is bowed at $\chi_B = 0.25L_s$ with a normal force $f_N = 1$ and bowing velocity $v_B = 0.1$. Furthermore, $\sigma_{0,s} = 1.0$, $\sigma_{1,s} = 0.005$, $\sigma_{0,p} = 0.0$, $\sigma_{1,p} = 0.005$.

A plot of the displacements of the Violino Arpa throughout this 1 second simulation is shown in Figure 4.4. The figure shows the Helmholtz motion in the vibration of the string, u , which is typical of bowed instruments and tends to produce triangular wave shapes. Also observable here are the stick-slip behavior of the bowing interaction, visible when looking at the resulting relative velocity v_{rel} with values fluttering around zero (stick) followed by an abrupt drop (slip).

Finally it is checked whether the implementation exhibits a hysteresis loop in the resulting bowing force vs. relative velocity, as it is something the elasto-plastic model is expected to produce and it is in line with experimental measurements shown by Woodhouse and Smith in [30]. This is illustrated in Figure 4.5.

Simulation with Static Friction Model

Running the simulation again with the same parameters as before but with the static friction model as given in Equation (4.20) the implementation does not produce satisfying results. The Newton-Raphson scheme fails to converge for many of the analyzed samples. Figure 4.6 shows a time series of the displacements of the system during a simulation and a very noisy signal is immediately noticeable. Looking next at Figure 4.7 it can be seen that the resulting bowing force as a func-

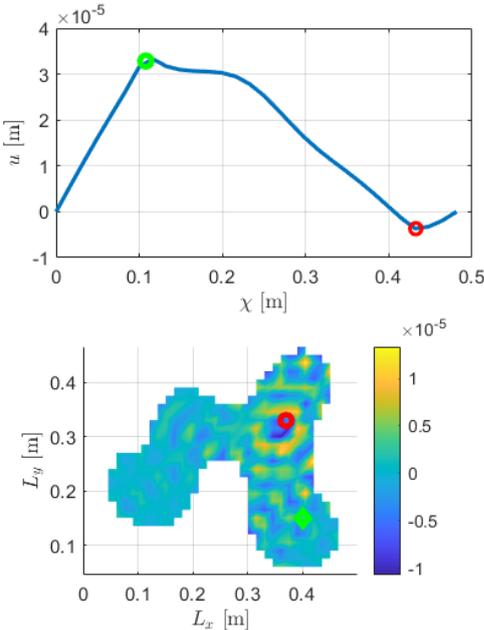


Figure 4.3: Snapshot showing the displacements of the friction drum’s components at a time step in the middle of a bowing simulation, top being the transverse displacements of the string u and bottom being the transverse displacements of the plate w . The red circle marks the connection point and the green circle marks the bowing position.

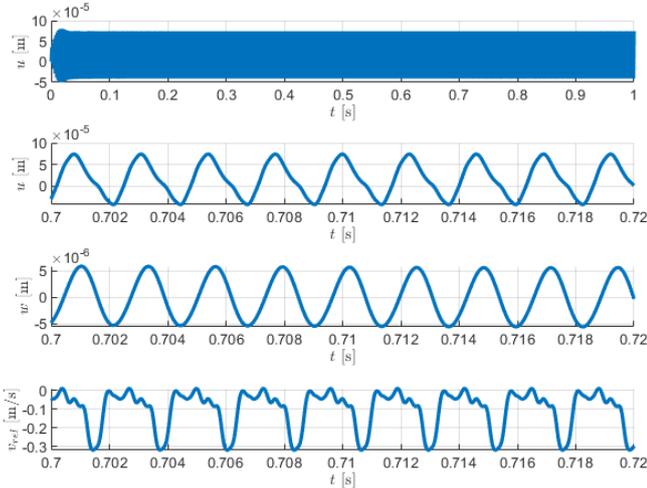


Figure 4.4: A plot of the resulting displacements in time during a bowing simulation of the Violino Arpa using an elasto-plastic friction model.

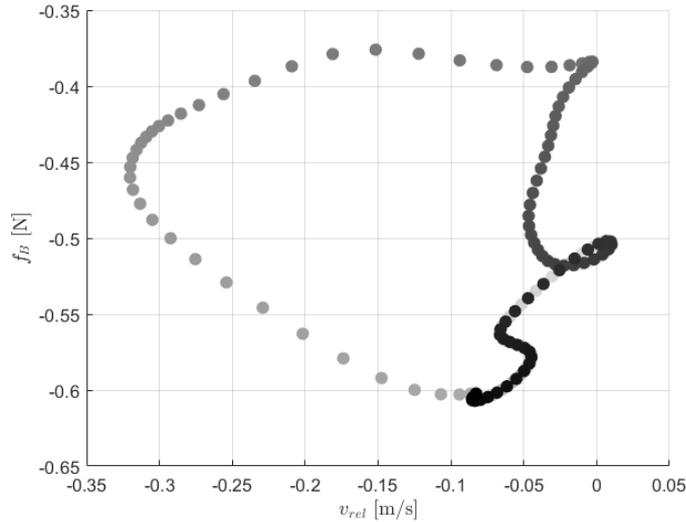


Figure 4.5: Hysteresis loop showing 150 points of relative velocity v_{rel} [m/s] and bowing force f_B [N], going from light to dark with increasing samples.

tion of the relative velocity falls in line with the expected behavior of the proposed static model shown in Figure 3.3. However, it seems that there are large jumps in the relative velocity between succeeding samples which causes the model to be unstable. Perhaps this is expected due to the large discontinuity present at zero relative velocity point. The static friction model misses the interesting behavior of the friction at low relative velocities. It was investigated whether this behavior may be due to the connection to the resonant plate but similar results were observed when simulating the bowed string alone.

An Alternative Static Model

As a reminder, the static model proposed initially in Equation (3.6) and discretized as per Equation (4.20) is in fact the same as the elasto-plastic dynamic model at steady-state. This was the main reason for choosing it as it would provide a more direct comparison between the two. However, as per the results presented above, it is suspected that the Newton-Raphson solution is not stable due to the discontinuity at zero relative velocity. Therefore an alternative static friction model is investigated which does not have such a discontinuity. It is in fact the model proposed in Figure 3.2c and taken from [2] with an additional viscous force:

$$f_B = f_N \sqrt{2a v_{rel}} e^{-a v_{rel}^2 + 1/2} + s_2 v_{rel}, \quad (4.28)$$

where the parameter $a = 1/v_S^2 = 100$ [s²/m²].

Figure 4.8 shows a comparison of the two proposed static models.

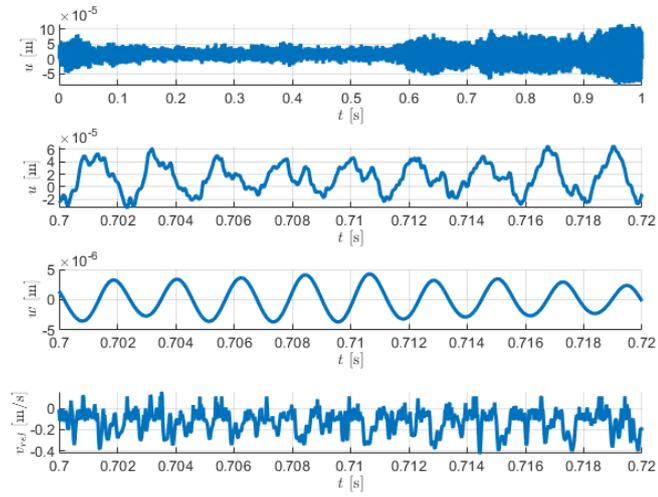


Figure 4.6: A plot of the resulting displacements in time during a bowing simulation of the Violino Arpa using the proposed static friction model given in Equation (3.6).

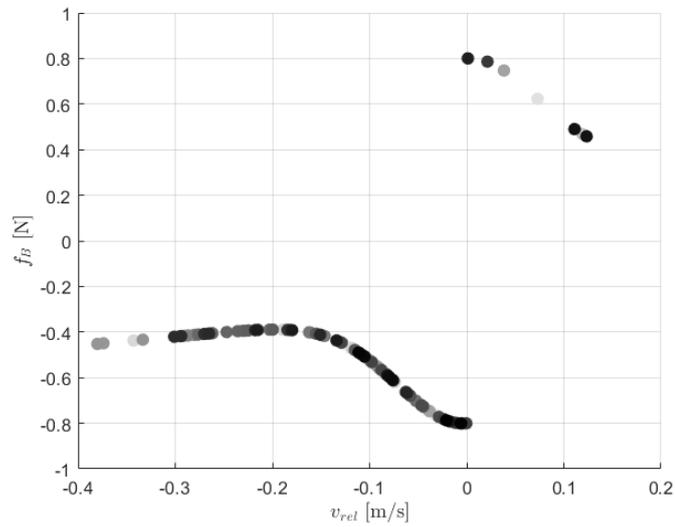


Figure 4.7: A plot showing 150 consecutive points of relative velocity v_{rel} [m/s] and bowing force f_B [N], going from light to dark with increasing samples, using the proposed static friction model given in Equation (3.6).

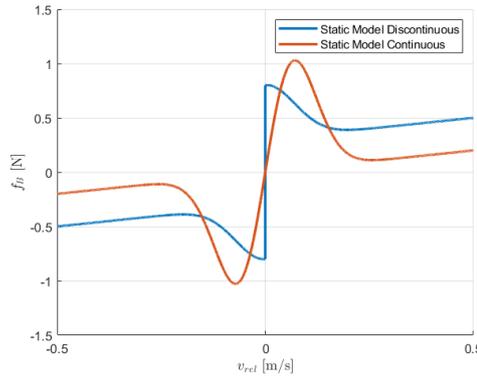


Figure 4.8: A comparison of the two proposed static models.

Next, a bowing simulation of the Violino Arpa is carried out with this continuous static friction model with the same parameters that produced Figure 4.4 and Figure 4.6 (for the elasto-plastic and the static discontinuous model respectively). The Newton-Raphson iterative algorithm now converges for all samples and the displacements of the system are plotted in Figure 4.9a. One can see the results are much more in line with the expectations and exhibits similar features to the elasto-plastic model, i.e. Helmholtz motion and stick-slip behavior. The hysteresis loop behavior in the friction force versus relative velocity plane is however not captured as this friction model does not have frictional memory, like the elasto-plastic does, but the results fall in the expected v_{rel} vs f_B curve. This is illustrated in Figure 4.9b.

4.3.2 Real-Time Application

An audio application was developed in C++ using the JUCE platform [26]. The application was integrated with the Sensel Morph, which is a tablet-size pressure sensitive controller, [10], ideal for modulating the bowing excitation in a natural way, particularly the normal bowing force, $f_N \in [0, 2]$ which is mapped to the pressure. Other parameters modulated via the Sensel are the bowing position along the string $\chi_B \in [0.1L_s, 0.7L_s]$ and is mapped to the x touch position on the Sensel (horizontal). The bowing velocity $v_B \in [0, 1]$ is mapped to the y touch position (vertical). The integration of the Sensel with the JUCE C++ code is based on the work of Willemsen et al. [35]. Furthermore, a number of sliders are used to control another set of parameters: the tuning of the string $f_0 \in [100, 660]$ (Frequency slider), the string's loss coefficients $\sigma_{0,s} \in [0, 2]$ and $\sigma_{1,s} \in [0, 0.5]$ (sig0 and sig1 sliders) and the plate's frequency dependent loss coefficient $\sigma_{1,s} \in [0, 0.1]$ (sig1p slider).

A fixed number of grid intervals is chosen for the discretization: for the string $N_\chi = 30$ grid intervals are used and $N_x = N_y = 35$ intervals for the square plate

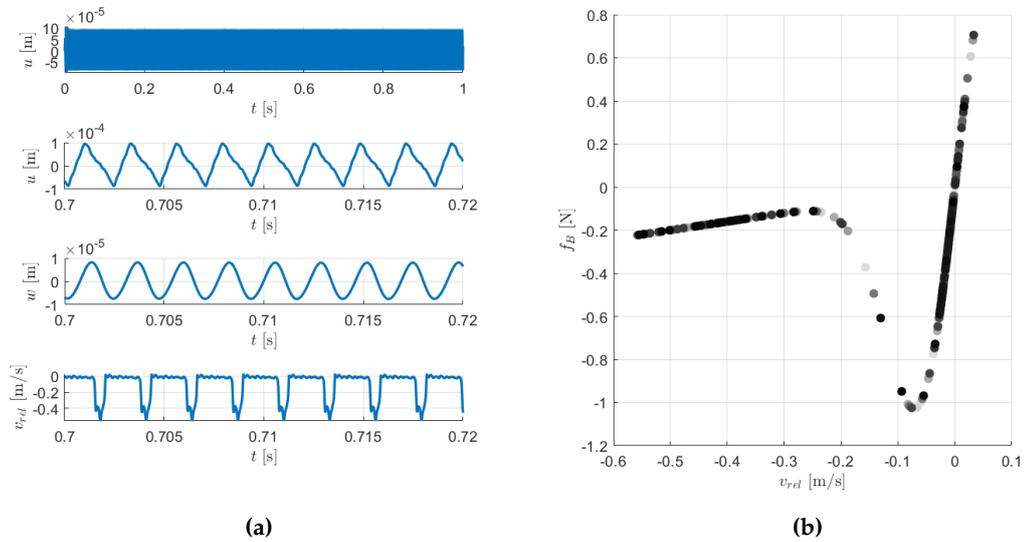


Figure 4.9: (a) Resulting displacements in time during a bowing simulation of the Violino Arpa using a continuous static friction model (b) Resulting bowing force as a function of the relative velocity between the bow and the string for the same simulation.

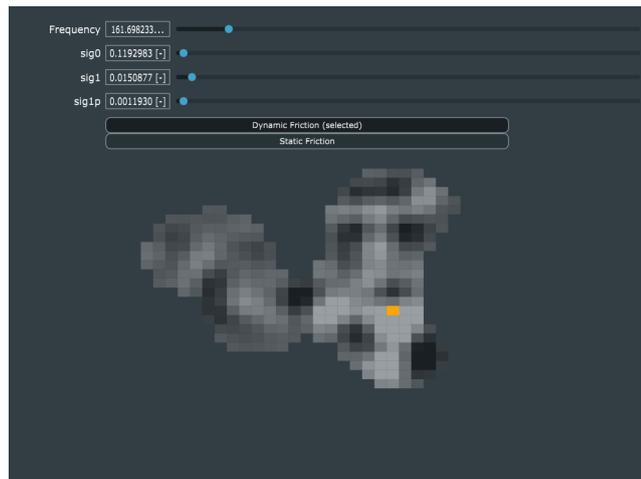


Figure 4.10: A comparison of the two proposed static models.

from which the amobea resonant plate is shaped. The shape is pre-computed using Matlab as described in the Section 4.2.2 and the locations of interest on the grid are saved in a text file which is read in the C++ implementation.

Both the dynamic elasto-plastic friction model as well as the continuous static friction model, shown in Equation (4.28), are implemented in the application and can be switched in real-time using a button group.

Sound is mono and taken as the transverse displacements u of the string at the connection point to the resonant plate. An additional gain factor is used to bring the displacements in the range of $[-1,1]$. This gain is varied as a function of the tuning frequency as for lower frequencies, the resulting displacements are larger than for high frequencies of tuning.

The vibrations of the violin's discretized resonant plate are plotted in gray scale in real-time for visual feedback. The location of the connection point to the string is highlighted as an orange grid square.

Figure 4.10 shows a screenshot of the application during a simulation and a demonstrative video can be found on Youtube at [24].

4.3.3 Real-Time Comparison of Friction Models

In order to evaluate the sounds produced by the real-time audio application under more natural playing conditions, modulated input parameters are saved in text file logs, during a "performance". These parameters can then be read-in as inputs to the app using either the dynamic friction model or the static friction model and the results can be compared. This could provide an interesting evaluation base for assessing whether the more complicated elasto-plastic implementation has benefits over the simpler static model in terms of real-time interaction. They are: bowing normal force f_N , bowing velocity v_B , bowing position χ_B , tuning frequency of the string f_0 .

Figure 4.11 shows the results for the two cases.

It must be noted that the input parameters were not logged in a continuous manner, i.e. at every single sample as that would cause audio dropouts and the natural "performance" aspect would be lost. Instead they were saved as part of the Sensel's hi-resolution callback function which was set at 150 Hz. Still, the Sensel only produces changes in the parameters when a touch is detected and thus this 150 Hz sample rate is not necessarily mirrored in the input log files. Lags due to the actual writing to file would also cause issues. In order to tackle this, for each input parameter saved, the sample number at print is also saved. When these logs are read as inputs, the associated samples are compared with the "current" sample in the model and parameters are changed if a new sample is reached. In-between these samples the parameters are kept constant. This explains the discontinuities in the input parameter values in Figure 4.11.

A closer inspection can be done by zooming in on a specific time interval and comparing the two results. This can be seen in Figure 4.12 where a 2.5 second interval is focused on where input parameters are somewhat constant except for a triangular envelope in the normal bowing force f_N . Here it can be seen that the elasto-plastic model seems to behave more smoothly with more gradual changes in the wave shape. This is perhaps due to accurate modelling of the various friction

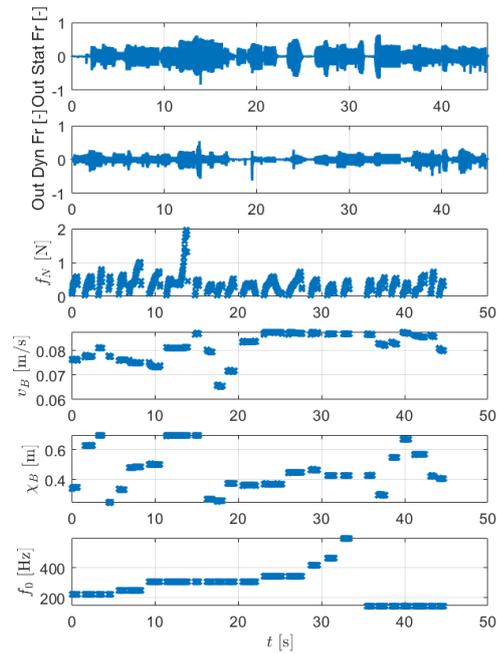


Figure 4.11: A comparison of the outputs of the real-time Violino Arpa app using a static and a dynamic friction model under modulated input parameters.

regimes described in Section 3.4.

Videos with the two simulation of the real-time app where the sound can be heard can be found on Youtube at [20] and [19] for the dynamic friction model and the continuous static friction model respectively.

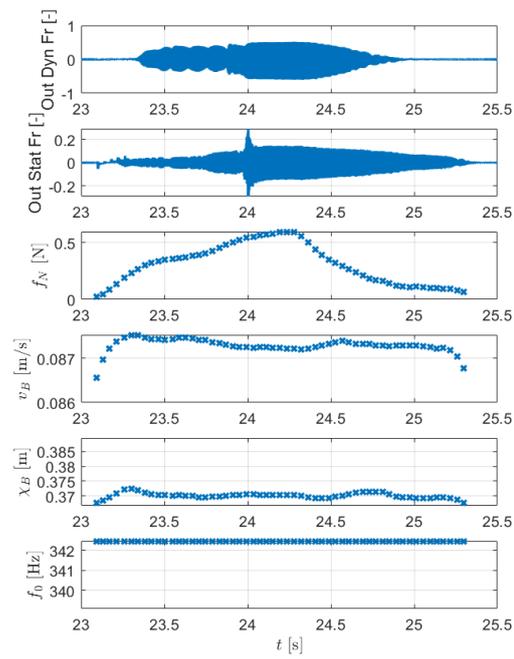


Figure 4.12: A comparison of the outputs of the rea-time Violino Arpa app using a static and a dynamic friction model under modulated input parameters zoomed in on a 2.5 second interval.

Chapter 5

Friction Drum

This chapter presents the steps towards the development of a real-time audio application that models a friction drum using both a static and a dynamic friction model. A paper was written as a result of this work in collaboration with Silvin Willemsen and Stefania Serafin and is currently under peer review for the DAFx 2021 conference in Vienna. It can be found in Appendix B.

5.1 Continuous Model

The friction drum can be modelled as a bowed membrane connected to an acoustic tube, which mimics the sound box. For the bowing model, either the dynamic elasto-plastic friction model given in Equation (3.8) is used or the continuous static friction model presented in the last chapter in Equation (4.28). The PDEs for each of the components in isolation are presented in this section.

5.1.1 Membrane

A square membrane is considered, defined over a domain $\mathcal{D}_m = [0, L_x] \times [0, L_y]$, with L_x and L_y being the lengths of the membrane [m] in the Cartesian coordinates (x, y) . Its transverse displacement at a time t [s], $u(x, y, t)$ [m] is characterized by the following PDE:

$$\partial_t^2 u = c_m^2 \Delta u - 2\sigma_{0,m} \partial_t u + 2\sigma_{1,m} \Delta \partial_t u + \frac{f_{c,m}}{\rho_m H_m} E_m - \frac{f_b}{\rho_m H_m} \delta(x - x_B, y - y_B), \quad (5.1)$$

where the 2-D Laplacian operator Δ is defined in Equation (4.4).

The parameter $c_m = \sqrt{T_m / \rho_m H_m}$ is a measure of wave speed resulting from the membrane's tension per meter T_m [N/m], its density ρ_m [kg/m³] and its thickness H_m [m]. Furthermore, $\sigma_{0,m}$ [s⁻¹] and $\sigma_{1,m}$ [m²/s] are parameters controlling the frequency-dependent and frequency-independent loss respectively.

The force resulting from the connection to the acoustic tube is f_C [N] and is distributed over some area of the membrane using the distribution function E_m [$1/m^2$]. Details regarding this distribution are given in the next section, dealing with the discretization of the continuous model. Finally, the resulting bowing force f_B [N] is applied at a bowing location (x_B, y_B) by means of the 2-D Dirac delta function $\delta(x - x_B, y - y_B)$ [$1/m^2$].

Dirichlet boundary conditions are assumed for the membrane so that:

$$u = 0, \quad (5.2)$$

at the edges of the membrane.

5.1.2 Acoustic Tube

The behavior of acoustic tubes is typically modelled in terms of deviation in pressure about a mean [2]. However in order to have a coupled system, the vibration of air layers of equal pressure is considered, i.e. longitudinal vibration of the air column $\zeta(\chi, t)$ [m] in a tube of uniform cross-section and length L_χ . This is described by the following PDE:

$$\partial_t^2 \zeta = c_t^2 \partial_\chi^2 \zeta - \frac{f_C}{\rho_t A_t} E_t, \quad (5.3)$$

with the wave speed $c_t = \sqrt{B_t/\rho_t}$ [m/s] resulting from the bulk modulus B_t [Pa] and density ρ_t [kg/m^3] of the air inside the tube. $\chi \in [0, L_\chi]$ is a spatial coordinate along the length of the tube. f_C [m] is the connection force distributed via the distribution E_t [1/m], equal and opposite to the membrane.

For the sizes of typical friction drum tubes, this 1-D wave approximation does not hold. It is only valid for tubes where the length scale in the longitudinal direction is significantly larger than the others. Ideally, one would use a 3-D wave equation to model this sound box but this would greatly increase the complexity of the implementation. Using the wave speed c_t based on the bulk modulus and density of air would result in a tube tuned to a fundamental frequency that is above what is expected for a typical "drum" sound. However, since this is a virtual model, we don't have to assume the drum is played in air! Therefore this wave speed can be tuned to produce the desired type of sound in the friction drum system.

The boundary conditions for the tube are considered as such: at the membrane-side of the tube a Neumann boundary condition is used, while at the open end of the tube a radiating boundary condition is chosen:

$$\partial_\chi \zeta = 0, \quad (5.4a)$$

$$\partial_\chi \zeta = -\alpha_1 \partial_t \zeta - \alpha_2 \zeta, \quad (5.4b)$$

with the constants α_1 and α_2 modelling the loss and inertia at the open end of the tube.

5.1.3 Connection

A rigid connection between the membrane and the tube is assumed, meaning that:

$$\eta = \langle u, E_m \rangle_{\mathcal{D}_m} - \langle \zeta, E_t \rangle_{\mathcal{D}_t} = 0, \quad (5.5)$$

with $\langle \cdot, \cdot \rangle$ being an L^2 inner product over the appropriate domain (1D for the tube and 2D for the membrane). Therefore η is the relative displacement of the two components over the connection. More details regarding the choice of connection distributions is given in Section 5.2.

5.1.4 Bowing Force

As mentioned, the resulting bowing force is modelled using either the elasto-plastic dynamic friction model given in Equation (3.8) is used or the continuous static friction model presented in the last section in Equation (4.28).

The relative velocity between the bow and the string $v_{rel}(t)$ can be found using the following equation:

$$v_{rel} = \partial_t u(x_B, y_B) - v_B, \quad (5.6)$$

with $u(x_B, y_B)$ being the transverse displacement of the membrane at the bowing position and v_B being the velocity of the bow.

5.1.5 Complete System

The complete system for the friction drum can be written as:

$$\left\{ \begin{array}{l} \partial_t^2 u = c_m^2 \Delta u - 2\sigma_{0,m} \partial_t u + 2\sigma_{1,m} \Delta \partial_t u + \\ \quad + \frac{f_c}{\rho_m H_m} E_m - \frac{f_b}{\rho_m H_m} E_b \\ \partial_t^2 \zeta = c_t^2 \partial_\chi^2 \zeta - \frac{f_c}{\rho_t A_t} E_t, \\ \eta = \langle u, E_m \rangle_{\mathcal{D}_m} - \langle \zeta, E_t \rangle_{\mathcal{D}_t} = 0, \end{array} \right. \quad (5.7a)$$

$$(5.7b)$$

$$(5.7c)$$

5.2 Discretization

The same approach which was used to discretize the Violino Arpa model in Section 4.2, using the FDTD methods introduced in Chapter 2 are also used to discretize the friction drum continuous time model. The (x, y) -plane of the membrane is discretized as $x = lh_m$ and $y = mh_m$, with $l \in [0, \dots, N_x]$ and $m \in [0, \dots, N_y]$. Here, $N_x = \text{floor}(L_x/h_m)$ and $N_y = \text{floor}(L_y/h_m)$ are the horizontal and vertical number of grid intervals the membrane is divided in with grid spacing h_m [m]. For simplicity the same spacing is used in both directions. Similarly for the tube, $\chi = ph_t$,

where $p \in [0, \dots, N_\chi]$ and $N_\chi = \text{floor}(L_\chi/h_t)$ is the total number of grid intervals along the tube's length with a grid spacing h_t [m]. With these discretization in place, the grid functions $u_{l,m}^n$ and ζ_χ^n are approximations of the continuous time transverse membrane displacement $u(x, y, t)$ and longitudinal displacement of the air column inside the acoustic tube, $\zeta(\chi, t)$.

Again, stability conditions with respect to the size of the grid spacing will be given for each individual component.

Using the finite difference operators from Chapter 2, the continuous time PDEs describing the elements of the friction drum can be approximated and computed numerically.

5.2.1 Membrane

The PDE describing the transverse displacement of the membrane given in Equation (5.1) is discretized as:

$$\begin{aligned} \delta_{tt}u_{l,m}^n = & c_m^2 \delta_\Delta u_{l,m}^n - 2\sigma_{0,m} \delta_t \cdot u_{l,m}^n + 2\sigma_{1,m} \delta_t - \delta_\Delta u_{l,m}^n + \\ & + \frac{f_c^n}{\rho_m H_m} J_m - \frac{f_b^n}{\rho_m H_m} J_b(x_B, y_B) \end{aligned} \quad (5.8)$$

Here the parameters J_m and J_b are introduced as discretized versions of the membrane connection distribution E_m and the 2-D Dirac delta function $\delta(x - x_B, y - y_B)$ which specifies the bowing position. For the case of J_b , it can be taken as a first order 2-D spreading operator, as was considered for the plate connection of the Violino Arpa in Equation (4.13). It is defined as:

$$J_b = \frac{1}{h_m^2} \begin{cases} (1 - \alpha_{x_B})(1 - \alpha_{y_B}) & l = l_B, m = m_B \\ (1 - \alpha_{x_B})\alpha_{y_B} & l = l_B, m = m_B + 1 \\ \alpha_{x_B}(1 - \alpha_{y_B}) & l = l_B + 1, m = m_B \\ \alpha_{x_B}\alpha_{y_B} & l = l_B + 1, m = m_B + 1 \\ 0 & \text{otherwise,} \end{cases} \quad (5.9)$$

with $l_B = \text{floor}(x_B/h_m)$, $m_B = \text{floor}(y_B/h_m)$, $\alpha_{x_B} = x_B/h_m - l_B$ and $\alpha_{y_B} = y_B/h_m - m_B$.

So far the membrane is considered as a square grid but friction drums are however typically circular. As was done for "sculpting" the resonant plate of the Violino Arpa, the membrane of the friction drum can also be modelled as a circle using a staircase approximation [7]. A new issue however is that of the membrane connection to the acoustic tube. It is clear that the entire membrane contributes to the displacement of the air inside the tube and therefore a spreading function as the ones previously described, which model distributions to a point location, cannot be used here. A uniform distribution over the entire membrane could be

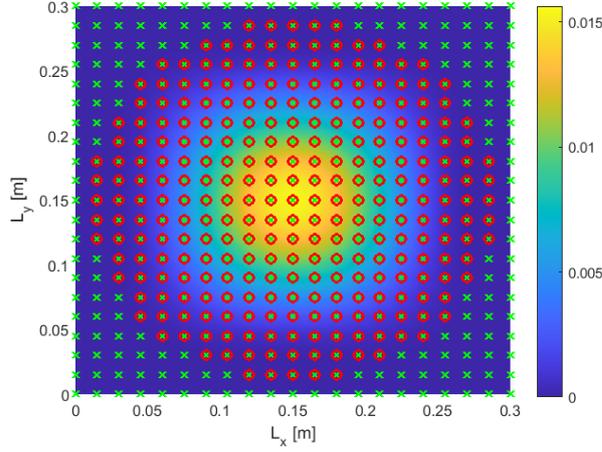


Figure 5.1: Circular grid approximation from a rectangular grid and the normalized Hann distribution used for the connection to the tube, I_m . The green crosses are the original grid points from the square $L_x \times L_y$ grid, while the red circles are the points used in the calculation.

the way to go, but since the boundary layer effect causes the air at the boundaries of the tube to be semi-stationary, a distribution more skewed towards the center would be a better solution. It is decided to use a 2-D Hann distribution for the connection distribution spread over 72.25% of the area of the grid, centered at the middle of the membrane. This is illustrated in Figure 5.1 together with the grid points of the circular membrane approximated from the initial rectangular grid.

This connection distribution, named I_m , is normalized such that its integral is equal to 1 and can be seen to act as an interpolation function acting on $u_{l,m}^n$. Therefore its dual spreading function J_m will be:

$$J_m = \frac{1}{h_m^2} I_m. \quad (5.10)$$

Boundary conditions are satisfied as long as the values of $u_{l,m}^n$ at the edge rows and columns of the grid space are set to 0.

Using von Neumann analysis [2], a stability condition can be derived and is given by the following inequality:

$$h_m \geq h_{m,min} = \sqrt{2c_m^2 k^2 + 8\sigma_{1,m} k}. \quad (5.11)$$

5.2.2 Acoustic Tube

Equation (5.3) describing $\zeta(\chi, t)$ can be discretized as:

$$\delta_{tt} \zeta_p^n = c_t^2 \delta_{\chi\chi} \zeta_p^n - \frac{F_c^n}{\rho_t A_t} J_t. \quad (5.12)$$

J_t is the spreading operator for the connection force acting on the tube, i.e. the discretized version of E_t . This is related to its dual interpolant function I_t in the following way:

$$J_t = \frac{1}{h_t} I_t. \quad (5.13)$$

I_t is taken as a half-Hann window spread over 4% of the length of the tube, with its peak at the connection point (the top of the acoustic tube), normalized such that its integral is 1. Spreading the connection across some length of the tube dampens out some of the high frequencies and produces a more realistic friction drum sound.

The boundary conditions of the tube presented in Equation 5.4 are discretized in the following way:

$$\delta_{\chi} \zeta_p^n = 0, \quad \text{at } p = 0 \quad (5.14a)$$

$$\delta_{\chi} \zeta_p^n = -\alpha_1 \delta_t \zeta_p^n - \alpha_2 \mu_t \zeta_p^n, \quad \text{at } p = N_{\chi}, \quad (5.14b)$$

and a stability condition on the grid size h_t is given by [2]:

$$h_t \geq h_{t,min} = c_t k. \quad (5.15)$$

5.2.3 Connection

The rigid connection given in Equation (5.5) can be discretized as:

$$\eta^n = \langle u_{l,m}^n, J_m \rangle_{\mathcal{D}_m} - \langle \zeta_p^n, J_t \rangle_{\mathcal{D}_t} = 0, \quad (5.16)$$

with \mathcal{D}_m and \mathcal{D}_t being the domains of the membrane and of the tube respectively.

Employing the identity given in Equation (2.12) to Equation (5.16) evaluated at sample $n + 1$, results in the following equality which will be of use when solving the finite difference system:

$$I_m u_{l,m}^{n+1} = I_t \zeta_p^{n+1}. \quad (5.17)$$

5.2.4 Bowing Force

The discretization of Equation (5.6) will be:

$$v_{rel}^n = I_b \delta_t u_{l,m}^n - v_B^n, \quad (5.18)$$

which will again be of use for solving the entire system.

The dynamic elasto-plastic friction force is discretized as was the case for the Violino Arpa, but the equation is given here as well:

$$f_{B,dyn}^n(v_{rel}^n, z^n) = s_0 z^n + s_1 r^n + s_2 v_{rel}^n + s_3 w^n, \quad \text{with:} \quad (5.19)$$

$$r^n(v_{rel}^n, z^n) = v_{rel}^n \left[1 - \alpha(z^n, v_{rel}^n) \frac{z^n}{z_{ss}(v_{rel}^n)} \right]. \quad (5.20)$$

Finally, the continuous static model is discretized as:

$$f_{B,stat}^n = f_N^n \sqrt{2a} v_{rel}^n e^{-av_{rel}^n{}^2+1/2} + s_2 v_{rel}^n + s_3 w^n, \quad (5.21)$$

with $a = 100 \text{ [s}^2/\text{m}^2]$.

5.2.5 Solving the Complete System

An additional caveat to the friction drum problem as opposed to the Violino Arpa is the fact that the bowing will always occur at the connection point. This is because the entire membrane is connected to the tube (72.25% of it to be more precise but the point holds). Therefore the bowing force f_B and the connection force f_C will be interdependent. Consequently, in order to calculate the update values for the grid functions: $u_{l,m}^{n+1}$ and ζ_p^{n+1} , values for three parameters need to be computed simultaneously in the case of the elasto-plastic friction model: v_{rel}^n , z^n and f_c^n . For the case of the static friction model, z^n is not needed. Once more, this is done by means of a multivariate Newton-Raphson numerical method. Three equations dependent on the unknown variables are needed at each sample n .

The first function $g_1(v^n, z^n, F_c^n)$ can be found by making use of the following identity:

$$\delta_{tt} u_{l,m}^n = \frac{2}{k} (\delta_t u_{l,m}^n - \delta_{t-} u_{l,m}^n), \quad (5.22)$$

and introducing it together with Equation (5.18) in Equation (5.8), which results in:

$$g_1(v^n, z^n, F_c^n) = I_b J_b \frac{F_b^n(v^n, z^n)}{\rho_m H_m} - I_b J_m F_c^n + \left(\frac{2}{k} + 2\sigma_{0,m} \right) v^n + q^n = 0, \quad (5.23)$$

with

$$q^n = -\frac{2}{k} \delta_{t-} I_b u_{l,m}^n + 2\sigma_{0,m} v_B^n + \frac{2}{k} v_B^n - c_m^2 I_b \delta_{\Delta} u_{l,m}^n - 2\sigma_{1,m} \delta_{t-} I_b \delta_{\Delta} u_{l,m}^n.$$

The second equation needed is, as per [36] and [2]:

$$g_2(v^n, z^n) = r^n - a^n = 0, \quad \text{with} \quad (5.24)$$

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

where the operators applied to z^n describe the trapezoid rule.

Finally, the third equation comes from the rigid connection condition in Equation (5.17). The displacements of the membrane $u_{l,m}^{n+1}$ and for the tube ζ_p^{n+1} can be

extracted and expressed only in terms of values at current or previous samples by expanding the operators in Equations (5.8) and (5.12). This results in

$$\begin{aligned} g_3(v^n, z^n, F_c^n) &= F_c^n k^2 \left[\frac{I_m J_m}{(1 + \sigma_{0,m}) \rho_m H_m} + \frac{I_t J_t}{\rho_t A_t} \right] - \\ F_b^n (v^n, z^n) k^2 &\frac{I_m J_b}{(1 + \sigma_{0,m}) \rho_m H_m} + b^n = 0, \end{aligned} \quad (5.25)$$

with

$$\begin{aligned} b^n &= \frac{1}{1 + \sigma_{0,m} k} [c_m^2 k^2 I_m \delta_\Delta u_{l,m}^n + 2\sigma_{1,m} k (I_m \delta_\Delta u_{l,m}^n - \\ &I_m \delta_\Delta u_{l,m}^{n-1}) + 2I_m u_{l,m}^n - (1 - \sigma_{0,s} k) I_m u_{l,m}^{n-1}] - \\ &[c_t^2 k^2 I_t \delta_{\chi\chi} \zeta_p^n + 2I_t \zeta_p^n - I_t \zeta_p^{n-1}] \end{aligned} \quad (5.26)$$

The following iteration is then used to calculate the unknown values v^n , z^n and F_c^n :

$$\begin{bmatrix} v_{(i+1)}^n \\ z_{(i+1)}^n \\ F_{c,(i+1)}^n \end{bmatrix} = \begin{bmatrix} v_{(i)}^n \\ z_{(i)}^n \\ F_{c,(i)}^n \end{bmatrix} - \begin{bmatrix} \frac{\partial g_1}{\partial v} & \frac{\partial g_1}{\partial z} & \frac{\partial g_1}{\partial F_c} \\ \frac{\partial g_2}{\partial v} & \frac{\partial g_2}{\partial z} & \frac{\partial g_2}{\partial F_c} \\ \frac{\partial g_3}{\partial v} & \frac{\partial g_3}{\partial z} & \frac{\partial g_3}{\partial F_c} \end{bmatrix}^{-1} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \quad (5.27)$$

where i is the iteration number. The threshold for convergence is set at 10^{-7} , with a maximum number of iterations of 99.

For the static friction model, the second equation $g_2(v_{rel}^n, z^n) = 0$ is not needed and the iteration becomes:

$$\begin{bmatrix} v_{(i+1)}^n \\ F_{c,(i+1)}^n \end{bmatrix} = \begin{bmatrix} v_{(i)}^n \\ F_{c,(i)}^n \end{bmatrix} - \begin{bmatrix} \frac{\partial g_1}{\partial v} & \frac{\partial g_1}{\partial F_c} \\ \frac{\partial g_3}{\partial v} & \frac{\partial g_3}{\partial F_c} \end{bmatrix}^{-1} \begin{bmatrix} g_1 \\ g_3 \end{bmatrix} \quad (5.28)$$

Once the unknown values at at the sample n are known, update values for the grid points $u_{l,m}^{n+1}$ and ζ_p^{n+1} can be found by expanding the operators in Equation (5.8) and Equation (5.12).

5.3 Implementation

The main goal, as was the case for the Violino Arpa, is to implement the finite difference scheme presented in Section 5.2 in a real-time audio application with dynamically adjustable parameters. A focus is placed on having a natural interaction with the app and on giving the possibility to somewhat mimic the sound of an acoustic friction drum but also leaving room for expanding its possibilities.

Table 5.1 shows a list of the parameters used in the model. For parameters which can be modulated in the real-time app, bounding intervals for their possible values are given. The choice of values is again inspired form the work of Serafin

Name	Symbol [unit]	Value
Membrane		
Length	L_x [m]	0.3
Width	L_y [m]	0.3
Material Density	ρ_m [kg/m ³]	1400
Thickness	H_m [m]	0.007
Wave Speed	c_m [m/s]	$15 \leq c_m \leq 150$
Freq. dep. loss	$\sigma_{0,m}$ [s ⁻¹]	$0 \leq \sigma_{0,m} \leq 6$
Freq. indep. loss	$\sigma_{1,m}$ [m ² /s]	$0 \leq \sigma_{1,m} \leq 0.0026$
Grid spacing	h_m [m]	0.0167
Acoustic Tube		
Length	L_χ [m]	0.4
Area	A_t [m ²]	0.0707
Material Density	ρ_t [kg/m ³]	1.225
Wave Speed	c_t [m/s]	30
Radiation Damp. Ct.	α_1 [s·m ⁻¹]	0.008
Radiation Damp. Ct.	α_2 [m ⁻¹]	4.348
Grid spacing	h_t [m]	0.0011
Bowing Model		
Coulomb Friction	μ_C [-]	0.3
Static Friction	μ_S [-]	0.8
Normal Force	f_N [N]	$0 \leq f_N \leq 20$
Bow Velocity	v_B [m/s]	$0 \leq v_B \leq 0.2$
Stribeck Velocity	v_S [m/s]	0.1
Bristle Stiffness	s_0 [N/m]	10^5
Bristle Damping	s_1 [kg/s]	$0.001\sqrt{s_0}$
Viscous Friction	s_2 [kg/s]	4
Noise Coefficient	s_3 [N]	$0F_N \leq s_3 \leq 0.04F_N$
Pseudorandom Fct.	w [-]	$-1 \leq w \leq 1$
Breakaway Disp.	z_{ba} [m]	$0.7f_C/s_0$
Other		
Sample Rate	f_S [Hz]	44100
Time Step	k [s]	$1/f_S$

Table 5.1: Parameter values used for the friction drum simulations.

[27] and Willemsen et al. [36], but more changes were required here compared to the Violino Arpa as the bowing problem was extended to 2-D. Tweaks to the parameters were carried out in order to achieve the desired friction drum sound.

A demonstrative video of the friction drum audio app can be found at [23].

5.3.1 Prototype

Before embarking on the real-time implementation, an offline implementation was carried out in Matlab [12], where all the data in the simulation is available at every sample and thus the results of the model could be checked in greater detail.

Figure 5.2 shows a snapshot of the circular membrane being bowed with $f_N = 12$ and $v_B = 0.1$ coupled with the acoustic tube at some time step in the middle of a simulation. The discretization used is: $N_x = N_y = 30$ grid intervals for the square grid of the membrane, from which the circular shape is approximated and $N_\chi = 350$ grid intervals are used for the discretization of the acoustic tube. This large number of intervals is possible due to the low wave speed the tube is tuned to in order to achieve the drum-like sound, $c_t = 30$. Looking at the longitudinal displacement along the tube in Figure 5.2, one can see the free and the radiating boundaries at its endpoints.

Looking at the time series results of the same simulation, shown in Figure 5.3a, the Helmholtz motion can be seen in the transverse displacements u at the bowing point on the membrane, which shows a triangular shape. When propagating the displacements of the membrane along the length of tube to its open end, (ζ in the figure), this triangular shape is distorted, as the tube gets input from an averaged out membrane displacement (due to the Hann connection distribution) as well as due to the effect of radiation at the open end. Looking at the relative velocity v_{rel} , the by now familiar stick-slip motion can be observed. A hysteresis loop is observed when looking at the evolution of resulting friction force, f_B with changing relative velocity v_{rel} , illustrated in Figure 5.3b, which is an expected behavior as per experimental observations of bowed strings by Woodhouse and Smith [30].

Results with Continuous Static Friction Model

The exact same simulation is then carried out with the continuous static friction model, discretized in Equation (5.21). The same plots showing the time series of the displacements of the system and the $f_B v_S v_{rel}$ plane are generated and can be seen in Figure 5.4. The results are a more noisy sound as again, there are more abrupt changes in the relative velocity and consequently resulting bowing force. Perhaps tuning the a parameter in Equation 5.21 could reduce the stick time and improve the quality of the results, but it was chosen as $a = 1/v_S = 100$ in order to be consistent with the dynamic friction model.

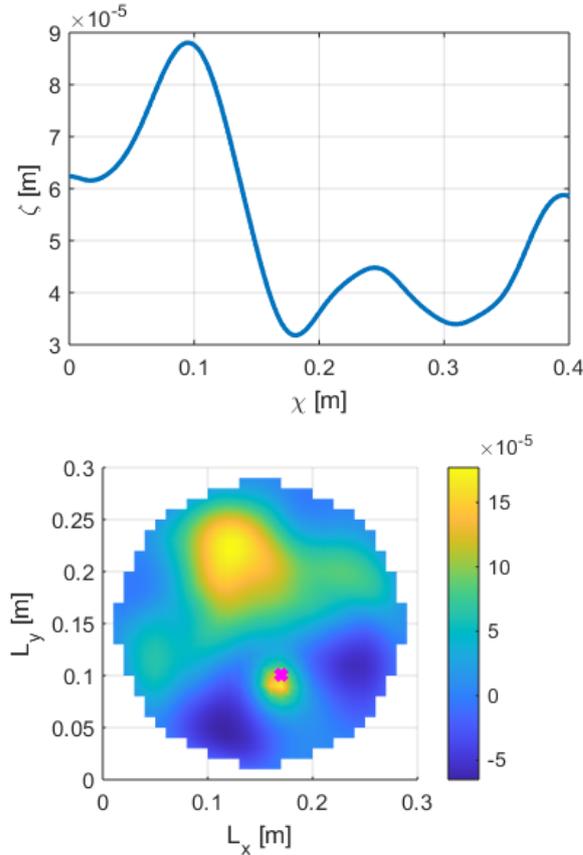


Figure 5.2: Snapshot showing the displacements of the friction drum's components at a time step in the middle of a bowing simulation, top being the longitudinal displacements of the air column in the acoustic tube ζ and bottom being the transverse displacements of the membrane u . The magenta cross highlights the bowing position.

5.3.2 Real-Time Application

A real-time application was implemented in C++ using the JUCE platform [26]. Figure 5.5 shows snapshots of the audio application while being used, where due to variation of the bowing position, normal bowing force and bowing velocity different modes of vibration are in resonance. For interaction with the app, again, the Sensel Morph was used as it proved to be an ideal solution due to its instant 3-D control: horizontal and vertical touch position and pressure sensors [10]. This was necessary as navigating across the 2-D surface of the membrane in order to vary the bowing position with ease provides a more natural feel. Hence the touch position was mapped to the bowing location (x_B, y_B) . Furthermore, the pressure was mapped to the bowing normal force and velocity, linearly coupled. The normal

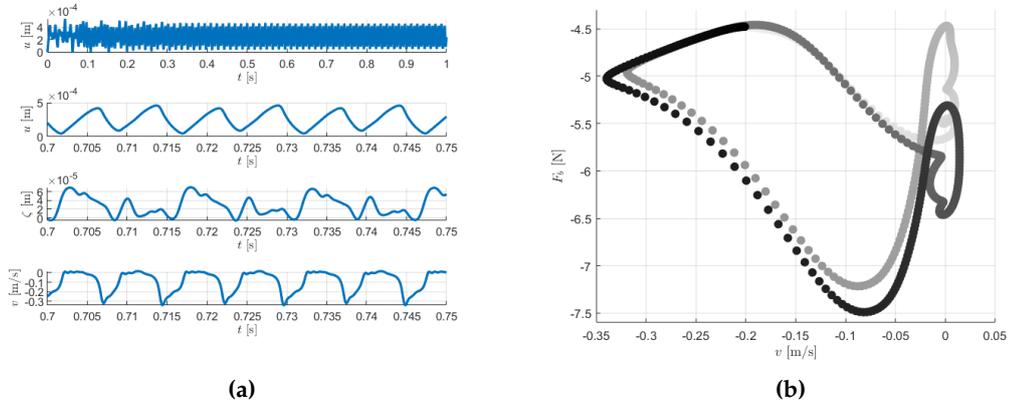


Figure 5.3: (a) A plot of the resulting displacements in time during a bowing simulation of the friction drum using an elasto-plastic friction model. (b) Hysteresis loop showing 700 points of relative velocity v_{rel} and bowing force f_B , going from light to dark with increasing samples in the same simulation.

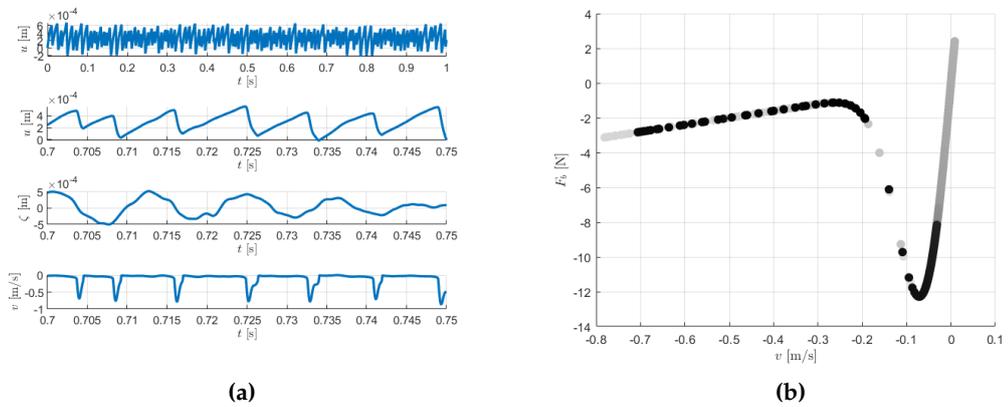


Figure 5.4: (a) A plot of the resulting displacements in time during a bowing simulation of the friction drum using a continuous static friction model. (b) Hysteresis loop showing 700 points of relative velocity v_{rel} and bowing force f_B , going from light to dark with increasing samples in the same simulation.

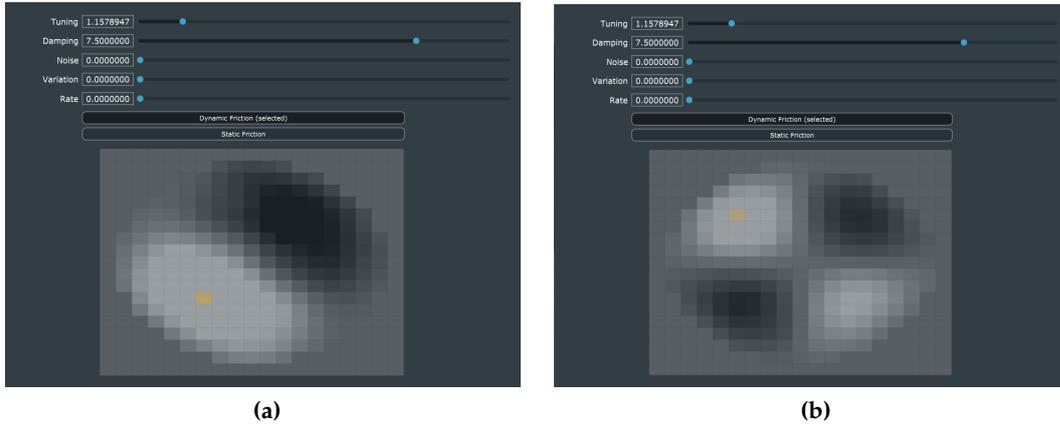


Figure 5.5: A screenshot of the real-time audio application where a resonance occurs with (a) mode 2 (b) mode 3.

force is limited in the range of $f_N \in [0, 20]$ while the bowing velocity is mapped in the range $v_B \in [0, 0.2]$. Therefore, the user can with only one finger control a series of important parameters.

Additionally a number of sliders are added to control more physical parameters of the model. The choice for their names was aimed at a non-technical user. Going from top to bottom, the first slider is called "Tuning" and controls the tuning of the membrane by changing the wave speed $c_m \in [15, 150]$. The second, named "Damping" groups the damping parameters $\sigma_{0,m} \in [0, 6]$ and $\sigma_{1,m} \in [0, 0.00266]$ of the membrane. As a side-note, checks were made in the choice of the ranges of c_m and $\sigma_{1,m}$ as to make sure that the stability condition given in Equation (5.11) is satisfied for the chosen grid size of the model. Also note, that even when the damping parameters are set to 0 the radiation damping and loss parameters for the tube boundary conditions: α_1 and α_2 are fixed. Hence, even with zero damping, there will still be decay present.

Moving back to the sliders, the third slider controls the $s_3 \in [0f_N, 0.04f_N]$ term in the bowing force, and is called "Noise" as it adds some white noise to the friction force proportional to the normal force f_N . The last two sliders named "Variation" and "Rate" control an additional effect included in the audio app: a vibrato effect which modulates via a sine wave the tuning of the membrane by a chosen frequency and with a chosen amount. "Variation" which adds an oscillation between $[0, 3]$ to the wave speed c_m and "Rate", which controls oscillation frequency of the sine wave in the range $[0, 10]$.

Again, as to not confuse a user with varying ranges for all the different parameters, it is chosen to map all the slider values linearly to a $[0, 10]$ non-dimensional scale.

As was the case for the Violino Arpa, a button group below the sliders can be

used to switch between the dynamic elasto-plastic friction model and the continuous static friction model in real-time.

Inspired by the work of Willemsen et al. [35], the vibration of the membrane is plotted in real-time in a gray scale, together with the bowing position, plotted with an orange color and an opacity given by the amount of pressure one applies to the Sensel. This provides a useful visual feedback when interacting with the application. Particularly one can clearly see when the excitation mechanism causes resonance with different modes of vibration, as illustrated in Figure 5.5.

The output is monophonic and is taken as the state of the model at the open end of the tube, i.e. $\zeta_{N_x}^n$ and amplified in the usual range $[-1,1]$. An adjustable gain is used in order to take into account the decreasing amplitudes of the model when increasing the wave speed c_m .

The number of grid intervals used to discretize the acoustic tube is $N_x = 350$, while the membrane is discretized in $N_x = N_y = 18$ intervals in the (x,y) direction.

5.3.3 Real-Time Comparison of Friction Models

The same type of comparison presented in Section 4.3.3 for investigating the sound of the Violino Arpa app with respect to the dynamic and static friction model, was also carried out for the friction drum app.

Control input parameters to the app were saved during a "performance" and could then be used as inputs to the app using either the dynamic or the static friction model. These parameters are: the bowing normal force f_N , bowing velocity v_B , bowing position (x_B, y_B) and membrane wave speed c_m . Figure 5.6 shows the sound output from two comparison runs using the different friction models but the same input control.

The issue of the discontinuous sample rate of the input parameters was discussed in Section 4.3.3. A closer look at the resulting wave forms from the two simulations can be seen in Figure 5.7.

Video recordings of the two simulations can be found on Youtube at [22] and [21].

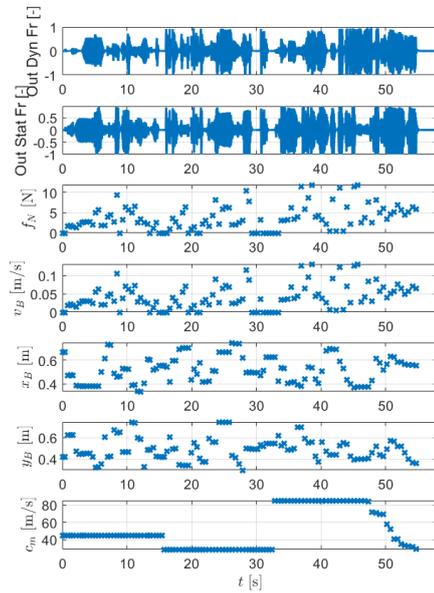


Figure 5.6: A comparison of the outputs of the real-time friction drum app using a static and a dynamic friction model (top two figures) under fixed modulated input parameters (the rest of the figures).

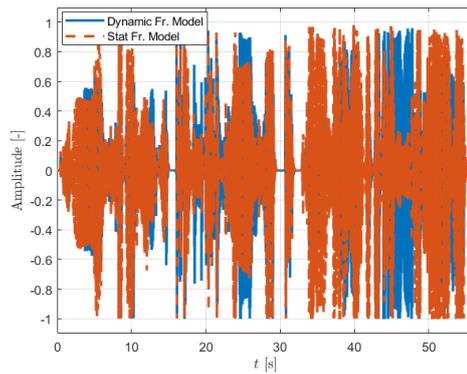


Figure 5.7: The overlaid outputs of the real-time friction drum app using a static and a dynamic friction model under the same input conditions.

Chapter 6

Evaluation

Due to COVID-19 restrictions the option to gather test subjects who could use the developed audio applications in person was not available. Luckily though, a possibility came about to present a demo of the friction drum app with the dynamic friction model during a Zoom session with 17 students enrolled in a physical modelling for sound synthesis class, as part of the Master education in Sound and Music Computing at Aalborg University Copenhagen (AAU Cph).

6.1 Violino Arpa Sound Samples

The Violino Arpa audio app was unfortunately not evaluated. However, the physical model which was developed in Matlab as the prototype to the real-time implementation was included in a study on the experience of recreating the Amoeba Violin via augmented reality (AR). Single note sample sounds were simulated with the FDTD model proposed in Section 4.1, discretized as close to stability condition as possible. They were then included in the AR application developed by students of the Medialogy master programme at AAU Cph, which was evaluated by a number of persons with regards to the possibility of increasing the visitor engagement in museums like the Danish Music Museum. This work resulted in a paper which is currently under peer review for the Sound and Music Computing Conference 2021 (SMC2021), and it can be found in Appendix A. In the app, the music scene, where the sound samples could be played with, was the scene where the majority of participants in the study spent most of their time, mostly playing with the sounds.

6.2 Friction Drum App Demo

The friction drum application was "performed" for a duration of approximately 5 minutes in which the various parameters of the interface was explored. After the demo, a qualitative interview was carried out and an open discussion regarding the application took place. The students did not know anything about the nature of the application prior to the discussion.

The following questions guided the discussion:

1. What type of instrument does the sound remind you of?
2. What type of excitation do you think it is?
3. Does the sound feel "natural" or "synthetic"?
4. What do you think each parameter controls?
5. What do you think about the visual feedback?
6. Do you find the interface intuitive?
7. What would you change with respect to the graphical user interface?

To the question of which instrument it was, there were many varied answers. One student mentioned a theremin, a hint towards the distinct sustained quality of the sound during constant bowing. Another mentioned a metallic hand drum due to the more percussive parts of the performance. A gong was also mentioned, due to the long decay of the sound (perhaps related to the bits in the performance where the damping of the model was minimized). Finally, two students hinted at the possible friction excitation, one saying that the app could be modelling a bowed bar as the sound reminds him of a low frequency saw. And the most encouraging response was "a cymbal drum contact miced and bowed", which is not far from a friction drum. The same student highlighted that the sound feels like it is coming from "inside" the instrument, hence the contact mic statement. As the sound in the friction drum model is retrieved by following the state at the open end of the tube, one could imagine that in a way the listener is indeed inside the drum itself.

Most of the answers involved some reference to a "metallic" sound, which perhaps hints at inharmonicity or perhaps was a reaction to playing the instrument with very low damping.

Another observation to note here is that due to the increased grid spacing in the model as compared to what could be achieved at the limit of the stability condition, results are not as accurate as they could be. Particularly there is a bandwidth limitation in terms of what frequencies can be represented with the numerical solution when working at grid intervals far away from the stability condition. Considering a

constant time step, as resulting from a desired sampling frequency, it turns out that simulations of elements with higher wave speeds, i.e. tuned to higher frequencies, are closer to the stability condition than for the case of lower wave speeds. This means that higher notes are more accurately modelled over a larger frequency bandwidth. Perhaps this could be a reason why "brighter" sounds caught more attention.

Still, the demo was carried out with the intent to achieve a broad range of sounds, i.e. make the app sound as a "new" instrument and not aiming to reproduce the sound of a friction drum. In the author's opinion, if one focuses on small percussive friction excitations without allowing the membrane to resonate, the app can sound like some of the friction drums in online videos.

Answers to some of the other questions which guided the general discussion were that the sound feels natural to some extent, until certain parameters are modulated, like the inclusion of the vibrato effect. One student compared the sound to "a chair being dragged across the floor" which is in the author's view, another encouraging observation, as friction was the governing phenomenon. This also reminds that friction is the sound excitation mechanism not only of music instruments but also of every day objects such as squeaking doors or rubbed wineglasses [27].

The version of the friction drum app which was demoed was a previous one compared to what was shown in Chapter 5. Some of the comments the students made were used to redesign the graphical user interface (GUI) of the app. Particularly, the idea to rename the sliders into words non-technical users could easily understand, or reducing the area of the membrane's real-time plot of the vibrations to not overlap with sliders. Hence a qualitative evaluation like this can prove to be invaluable when designing an application for general use.

Chapter 7

Conclusion

7.1 Summary

This thesis presents the development of two real-time audio applications which simulate a pair of exotic instruments driven by friction: the Violino Arpa, and a friction drum. The excitation mechanism in both applications was modelled using two different friction models, specifically an elasto-plastic dynamic friction model and a continuously differentiable static friction model.

As part of the development process the friction mechanism and the behavior of the models were checked in detail in offline simulation environments where all the parameters of the simulation were available at all samples. The elasto-plastic friction model produced the expected behavior, observed in experiments on bowed strings, with respect to stick/slip motion as well as hysteresis in the friction force versus relative velocity plane. Comparisons with the static model were carried out both in the prototype environment, with a static control input, as well as using the real-time applications, where the control input was varied in a natural manner, as recorded via a "performance" with the apps. References to videos showing these comparisons were given, as well as to videos demo-ing the two applications.

A qualitative evaluation of the friction drum application was carried out as a group interview during a Zoom lecture part of the physical modelling for sound synthesis class at AAU Cph. The real-time Violino Arpa app was unfortunately not evaluated, but the synthesis model it is based on was included in an AR application which was evaluated with respect to the possibility of increasing the visitor engagement in museums like the Danish Music Museum.

7.2 Future Work

Possible future work can go in a number of different directions. First and foremost, a qualitative evaluation of the Violino Arpa app should be carried out. It should

ideally be done in person in a setting where the users can make use of the Sensel Morph as a controller. Of course not everybody has access to such a controller, so different, more easily accessible, interaction models could be investigated for both audio applications.

Regarding possible evaluations, a more quantitative approach could be comparing the sounds of the app with the actual sounds of the Violino Arpa using spectral analysis methods. This would however only be possible with the permission of the Danish Music Museum. The same could be done for the friction drum, but as a reminder, the goal of this project was - more so in the case of the friction drum - not the perfect reproduction of the acoustic instruments, but creating new instruments based on the physics of their acoustic counterparts.

Another interesting direction regarding evaluation, which would be more easily implemented with the groundwork built in this project is to have a test subject based assessment regarding their preference of either the dynamic or the static friction model when used with the two applications. Does it matter more for one instrument than the other? Is it worth it to use the more complicated elasto-plastic implementation?

Moving on, additional future work should focus on optimizing the C++ implementation, which would result in reducing the grid spacing the systems are discretized in and therefore provide more accurate numerical solutions. Different mappings of the parameters which can be controlled can also be investigated. The velocity of the bow in the violin app could be taken as the actual motion velocity of the touch position on the Sensel. There is also room for work on the GUI of both applications. The state of the 1-D elements in the systems: the tube for the friction drum and string for the Violino Arpa should also be plotted.

All things considered there is a lot of work that can be done to improve the apps which can go on top of the great deal of work which was put into them so far. This is both fortunately, and unfortunately, the case in most projects in the field of PMSS: they can always be extended and are never fully done.

Bibliography

- [1] H. Balfour. “The friction-drum”. In: *The Journal of the Royal Anthropological Institute of Great Britain and Ireland* 37 (1907), pp. 67–92.
- [2] S. Bilbao. *Numerical Sound Synthesis*. John Wiley and Sons, Ltd, 2009. ISBN: 978-0-470-51046-9.
- [3] 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”. In: *Computer Music Journal* 43.2-3 (2020), pp. 31–47. DOI: 10.1162/comj_a_00517.
- [4] C. Canudas de Wit, H. Olsson, K. J. Astrom, and P. Lischinsky. “A new model for control of systems with friction”. In: *IEEE Transactions on Automatic Control* 40.3 (1995), pp. 419–425. DOI: 10.1109/9.376053.
- [5] B. Diamond. In: *Asian Music* 24.1 (1992), pp. 141–145. ISSN: 00449202, 15535630. URL: <http://www.jstor.org/stable/834456>.
- [6] P. Dupont, V. Hayward, B. Armstrong, and F. Altpeter. “Single state elastoplastic friction models”. In: *Automatic Control, IEEE Transactions on* 47 (June 2002), pp. 787–792. DOI: 10.1109/TAC.2002.1000274.
- [7] B. Hamilton. “Finite Difference and Finite Volume Methods for Wave-based Modelling of Room Acoustics”. PhD thesis. The Univeristy of Edinburgh, 2016.
- [8] V. Hayward, B. Armstrong, P. Corke, and J. Trevelyan. “A new computational model of friction applied to haptic rendering”. In: vol. 250. May 2008, pp. 403–412. ISBN: 978-1-85233-210-5. DOI: 10.1007/BFb0119418.
- [9] E. Y. Huynh. “Bowed Plates and Blown Strings: Odd Combinations of Excitation Methods and Resonance Structures Impact Perception”. *MA Thesis*. Department of Music Research, McGill University, Canada, 2019.
- [10] Sensel Inc. *Sensel Morph*. Available at <https://sensel.com/>. Accessed: 2021-04-07.

- [11] A. Luciani J.-L. Florens A. Razafindrakoto and C. Cadoz. “Optimized real time simulation of objects for musical synthesis and animated image synthesis”. In: *ICMC 86* (1986), 65–70.
- [12] MATLAB. *version (R2020b)*. Natick, Massachusetts: The MathWorks Inc., 2019.
- [13] M. McIntyre, R. T. Schumacher, and J. Woodhouse. “On The Oscillations of Musical Instruments”. In: *Journal of the Acoustical Society of America* 74 (1983), pp. 1325–1345.
- [14] G. E. Moore. “Cramming more components onto integrated circuits”. In: *Electronics* 38.8 (1965).
- [15] J. D. Morrison and J. M. Adrien. “MOSAIC: A Framework for Modal Synthesis”. In: *Computer Music Journal* 17.1 (1993), pp. 45–56. ISSN: 01489267, 15315169. URL: <http://www.jstor.org/stable/3680569>.
- [16] M. Müller. “Folk-folkelig-folkelige musikinstrumenter i Danmark”. In: *Folk og kultur, årbog for dansk etnologi og folkemindevidenskab* 23.1 (1994), pp. 5–20.
- [17] J. Na, Q. Chen, and X. Ren. “Chapter 1 - Friction Dynamics and Modeling”. In: *Adaptive Identification and Control of Uncertain Systems with Non-smooth Dynamics*. Ed. by J. Na, Q. Chen, and X. Ren. Emerging Methodologies and Applications in Modelling. Academic Press, 2018, pp. 11–18. ISBN: 978-0-12-813683-6. DOI: <https://doi.org/10.1016/B978-0-12-813683-6.00003-9>. URL: <https://www.sciencedirect.com/science/article/pii/B9780128136836000039>.
- [18] H. Olsson, K.J. Åström, C. Canudas de Wit, M. Gäfvert, and P. Lischinsky. “Friction Models and Friction Compensation”. In: *European Journal of Control* 4.3 (1998), pp. 176–195. ISSN: 0947-3580. DOI: [https://doi.org/10.1016/S0947-3580\(98\)70113-X](https://doi.org/10.1016/S0947-3580(98)70113-X). URL: <https://www.sciencedirect.com/science/article/pii/S094735809870113X>.
- [19] M. G. Onofrei. *Amoeba violin audio app simulation with a continuous static friction model*. <https://www.youtube.com/watch?v=f9EinyPgm0M>. Accessed: 2021-04-07. Youtube, 2021.
- [20] M. G. Onofrei. *Amoeba violin audio app simulation with a dynamic elasto-plastic friction model*. <https://www.youtube.com/watch?v=cYRQiwfPqno>. Accessed: 2021-04-07. Youtube, 2021.
- [21] M. G. Onofrei. *Friction drum audio app simulation with a continuous static friction model*. https://www.youtube.com/watch?v=8_-DXaGaR2o. Accessed: 2021-04-07. Youtube, 2021.
- [22] M. G. Onofrei. *Friction drum audio app simulation with a dynamic elasto-plastic friction model*. <https://www.youtube.com/watch?v=QSiW2I1xwlg>. Accessed: 2021-04-07. Youtube, 2021.

- [23] M. G. Onofrei. *Friction drum physical model simulation demo*. https://www.youtube.com/watch?v=_1IHdh6yg2g. Accessed: 2021-04-07. Youtube, 2021.
- [24] M. G. Onofrei. *Violino Arpa (Amoeba Violin) physical model simulation demo*. <https://www.youtube.com/watch?v=59SE3y7VQg4>. Accessed: 2021-04-07. Youtube, 2021.
- [25] IRCAM R. Dudas. *Modalys*. Available at <https://forum.ircam.fr/projects/detail/modalys/>. Accessed: 2021-04-07.
- [26] JUCE ROLI. *JUCE*. Available at <https://juce.com>. Accessed: 2021-04-07.
- [27] S. Serafin. "The sound of friction: Real time models, playability and musical applications". PhD thesis. Department of Music, Stanford University, 2004.
- [28] S. Serafin, F. Avanzini, and D. Rocchesso. "Bowed String Simulation Using An Elasto-Plastic Friction Model". In: 2003.
- [29] J. O. Smith. "Physical Modeling Using Digital Waveguides". In: *Computer Music Journal* 16.4 (1992), pp. 74–91. ISSN: 01489267, 15315169. URL: <http://www.jstor.org/stable/3680470>.
- [30] J.H. Smith and J. Woodhouse. "Tribology of rosin". In: *Journal of The Mechanics and Physics of Solids - J MECH PHYS SOLIDS* 48 (Aug. 2000), pp. 1633–1681. DOI: 10.1016/S0022-5096(99)00067-8.
- [31] J.O. Smith. "Efficient simulation of the reed bore and bow string mechanics". In: *ICMC* 86 (1986), 275–280.
- [32] J. Stock. "A Historical Account of the Chinese Two-Stringed Fiddle Erhu". In: *The Galpin Society Journal* 46 (1993), pp. 83–113. ISSN: 00720127. URL: <http://www.jstor.org/stable/842349>.
- [33] S. Willemsen. *ElastoPlastic*. Available at <https://github.com/SilvinWillemsen/ElastoPlastic>. Accessed: 2021-04-07.
- [34] S. Willemsen. *Lecture notes in Physical Models for Sound Synthesis*. 2020.
- [35] S. Willemsen, N. Andersson, S. Serafin, and S. Bilbao. "Real-time control of large-scaled modular physical models using the Sensel Morph". In: *Proceedings of the 16th Sound and Music Computing Conference*. 2019.
- [36] 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: *22nd International Conference on Digital Audio Effects*. 2019.
- [37] S. Willemsen, S. Serafin, S. Bilbao, and M. Ducceschi. "Real-Time Implementation of a Physical Model Of The Tromba Marina". In: June 2020.

Appendix A

Violino Arpa Paper

RECREATING THE AMOEBA VIOLIN USING PHYSICAL MODELLING AND AUGMENTED REALITY

Chrysoula KALOUMENOU¹, Sofia LAMDA¹, Panagiota POULIOU¹, Marius G. ONOFREI¹, Silvin WILLEMSSEN², Camilla JALLER², and Stefania SERAFIN²

¹Dept. of Architecture, Design & Media Technology, Aalborg University Copenhagen, Denmark

²Multisensory Experience Lab, CREATE, Aalborg University Copenhagen, Denmark

ABSTRACT

The Amoeba violin is a bowed string instrument from the Danish Music Museum in Copenhagen. The instrument is not played anymore due to its unpleasant sonorities and uncomfortable shape. In this paper we recreate the Amoeba violin using extended reality technologies and sound synthesis by physical models. We design and evaluate two applications that can be used either at the museum (augmented reality version) or at home (desktop version) to learn about the history of the instrument and its sonorities. The app was created for the Danish Music Museum, Musikmuseet, located in Copenhagen, in response to the demand for reduced contact of shared surfaces and official calls to stay indoors that followed the COVID-19 outbreak in 2020. User testing on both versions shows that they both are considered easy-to-access and educative, however the AR version was more favoured overall. Real-life implementation of the AR version would help interact with the instrument without touching it, whereas an implementation of the desktop version could be useful for people who cannot physically visit the museum.

1. INTRODUCTION

It is relatively recently that the experience of visiting a museum started diverting from its traditional meaning. Since the establishment of the museums as institutions up until the beginning of the 21st century, the visitors were generally expected to wander the exhibition grounds, while passively taking in the information that the curators had planned to provide them with [19]. This offered minimal chances for interaction and, ultimately, led to low visitor engagement [16]. In the last couple of decades, however, a growing number of museums around the world have enriched their exhibitions through the use of digital media, aiming to make the visit more engaging and memorable. As Brown argues in [8], implementing technologies like Virtual Reality (VR) and Augmented Reality (AR) is largely beneficial to cultural heritage sites, especially in communicating their product to younger audiences.

AR applications are an affordable solution that can work

very well in limited spaces, as modern mobile phones are powerful enough to support such technologies, and no extra spaces are required for its implementation.



Figure 1. A photograph of the Violino Arpa owned by the Danish Music Museum and a 3D visualization of it.

A few years earlier, in 1997, Feiner et al. [13] and Rekimoto [26], had already seen the potential uses of Augmented Reality (AR) for cultural applications. It wasn't long until the technology took off. One of the earliest example of a complete AR application for a cultural location was implemented at the archaeological sites of Pompeii, Italy [27], and Olympia, Greece [32] in 2001. That system, though primitive in its design and equipment, was rather powerful for its day and produced fairly believable results. This is confirmed by the generally good reviews those systems received from their users [23]. Since then, AR has been constantly gaining ground as an educational tool in the heritage sector. In fact it seems that there is a stronger tendency for the implementation of AR applications than for VR applications [5]. In this paper, we aim at using AR to reconstruct an instrument from the Danish Music Museum in Copenhagen, the Amoeba violin (see Figure 1). This instrument's sonorities were not considered as pleasant, so the instrument was not adopted (maybe also for its ergonomic). In order to recreate the sound of the instrument, we created a physically based simulation together with an interactive application that describes its history. We tested the application in both a desktop version and an AR version.

2. DESIGN

We designed an interactive application that aims to entertain the Danish Music Museum potential guests, providing educational information about the Amoeba violin also known as Violino Arpa (Fig. 1), a distinguished instrument that is an indivisible part of the museum's exhibition. It consists of two different versions: the AR and the desktop version, which we are going to compare and examine. The set up for running these two applications is comprised of an Android smartphone and a laptop.

2.1 Design considerations

The application elaborates on multiple characteristics and aspects of one of the most significant exhibits of the museum. The instrument's funky shape and history seem to be an attraction for the Danish Music Museum, which acted as an inspiration point for the concept development of the application. The application aims to familiarise the user with the instrument through the use of extended reality technologies, enhanced visual representations, and digital sound synthesis. In an attempt to provide a brief but concise set of information to the user, much like a typical museum visit would, we decided to organise the app in four stages (scenes) into which the user can navigate in a linear fashion. Three of those scenes correlate to the key areas that would be of interest regarding an historical musical instrument: its look; where the user has a chance to explore the shape of the Amoeba Violin in detail, its history; where facts regarding the background of the instrument are presented, and its sound; where an interactive musical player offers the chance to listen to the sounds of this unusual instrument. With regard to the amount of historical context offered, that is purposefully kept short and built around a story. This choice was made in an effort to limit less relevant information and make sure that what is included is memorable after using the application. In addition the aforementioned scenes another one was created, where the user is called to put together pieces of an Amoeba Violin in order to create a complete instrument. In fact, this is the first scene the user encounters on opening up the application; a design decision made in order to increase engagement. Each one of the four scenes fulfills a unique role that solidifies the above while attempting a fun approach. X-PLOER works as a prototype that could be applied to all the instruments of the museum.

3. IMPLEMENTATION

The app was built using Unity3D [31]. Multiple toolkits were also implemented, primarily EasyAR [12] and PlayMaker [18]. Autodesk Maya [4] was used to create the 3D components of the Violino Arpa, and Adobe Illustrator [2] and Adobe Photoshop [3] for the design of the history of the instrument.

The sound of the Violino Arpa is extremely hard to find, as the instrument was an unsuccessful experiment. Therefore, all of the sounds that are added in the application are simulations of what the instrument sounds like. Those simulations were designed in MATLAB and implemented in

real time using C++ and the JUCE framework [21].



Figure 2. The view when one enters the desktop application, which is an imitation of the staircase of the Danish Music Museum.

3.1 Physical modelling synthesis

The Violino Arpa is unique instrument that is fortunately preserved at Danish Music Museum in a good condition. However, due to its age combined with this desire for preservation the instrument cannot be played by the general public and recordings of its sound do not exist. This only leaves the possibility of a simulation of its sound, based on the physical properties and behavior of the instrument. In fact, the behavior of music instruments can be described by partial differential equations (PDEs), which describe the rates of change of certain variables of the considered process such as time or spatial position for instance. Solving these equations can take different routes with various degrees of complexity (and accuracy). There are certain techniques that can convert these PDEs to systems of equations which can be solved by linear algebra techniques, well suited for computer processing. These are called numerical analysis techniques, one of which is the Finite Difference Method (FDM) and was the choice for solving this physical system.

More detailed models can be employed, for example models where variation in material density and stiffness can be considered, like [10] who simulated the sound of an early viola da gamba by means Finite Element Methods. Such a route however can be very computationally expensive and the simulations can take significant time to run, leaving no possibility for extending this to a real-time implementation with the current computational infrastructure available.

The physical system used to model the Violino Arpa consists of a bowed stiff string, rigidly connected to a resonant plate. The PDEs describing this system in continuous time are given in Equations (1) to (3) and follow definitions from [7]. The displacements of the string and plate respectively are noted as $u(x, t)$ and $w(x, y, t)$. Furthermore, the subscripts t , x and y denote differentiation with respect to time, and the 2-D spatial coordinates.

The bowed stiff string is described by

$$\rho_s A_s u_{tt} = T_s u_{xx} - E_s I_s u_{xxxx} - 2\rho_s A_s \sigma_{0,s} u_t + 2\rho_s A_s \sigma_{1,s} u_{txx} - J_B F_B + J_{C,s} F_C, \quad (1)$$

with material density ρ_s [kg/m³], cross-sectional area $A_s = \pi r^2$ [m²], radius r [m] tension T_s [N], Young's Modulus E_s [Pa] and area moment of inertia $I_s = \pi r^4/4$ [m⁴]. Moving on, $\sigma_{0,s}$ [s⁻¹] and $\sigma_{1,s}$ [m²/s] are coefficients introducing frequency-dependent and frequency-independent damping respectively. Furthermore, F_B is the bowing force in [N] while J_B [m⁻¹] is a distribution function describing where the force is applied on the string. Finally, F_C is the connection force (via the rigid connection to the resonant plate) with $J_{C,s}$ [m⁻¹] describing the spread of the connection force.

The resonant plate is modelled using the Kirchhoff thin plate model with losses. Using the 2D Laplacian which when applied to w is equivalent to $\Delta w = w_{xx} + w_{yy}$ the PDE can be written as:

$$\rho_p H_p w_{tt} = -D_p \Delta \Delta w - 2\rho_p H_p \sigma_{0,p} w_t + 2\rho_p H_p \sigma_{1,p} \Delta w_t - J_{C,p} F_C. \quad (2)$$

The physical properties parameters that have been described for the string, are the same for the plate. The difference is indicated by the subscripts s or p . New parameters for the plate not previously described include plate thickness H_p [m] and $D_p = E_p H_p^3 / 12(1 - \nu_p^2)$ [kg·m²·s⁻²], with dimensionless Poisson's Ratio ν_p . Again, F_C [N] is the connection force over a distribution $J_{C,p}$ [m⁻²], (the magnitude of the force is the same for the plate and the stiff string, but the distributions can be different).

A perfectly rigid connection is assumed, meaning that at the location where the two elements connect, the displacement is equal. This can be written as

$$\langle u, J_S \rangle_{\mathcal{D}_s} = \langle w, J_P \rangle_{\mathcal{D}_p}, \quad (3)$$

where $\langle f, g \rangle_{\mathcal{D}}$ represents the inner product of f and g over domain \mathcal{D} . If the connection distributions, J_S and J_P are very localized (a Dirac function for instance), then it means that the connection is at a single location rather than distributed and Eq. (3) reduces to $u_P = w_P$ at some location P .

The boundary conditions, which describe the behavior of the elements at their edges are: simply supported for the string, i.e. $u = u_{xx} = 0$ and clamped for the plate, i.e. $u = u_n = 0$, where n is the coordinate normal to the edge.

Regarding the model used for the interaction between the bow and the string, a complex elasto-plastic friction model is used, capable of describing the non-linear relationship between the two components. This model was first introduced by Dupont [11] and investigated in detail with regards to its application in music by Serafin [28]. Furthermore, it was proved to be implementable in a real-time application by [36], whose implementation is the basis for the current model. This will not be further described here but the reader is invited to the above references for more details. Suffice to say is that the model takes in a number

of physical parameters related that describe the friction interaction and can provide a value for the bowing force F_B at each analysed sample.

These continuous time PDEs can be discretized and thus resolved numerically. This means the continuous system is divided in grid points in space: $x = lh$ and $y = mh$ and samples in time: $t = nk$, with h being the spatial step and k being the time step and $l \in [0, 1, \dots, N_x]$, $m \in [0, 1, \dots, N_y]$, where N_x and N_y are the number of horizontal and vertical grid points. For simplicity in the case of the plate, the same spatial step is assumed in both spatial directions. Discretizing the model in such a way comes with some downsides, as the accuracy and bandwidth of the solution is dependent on the number of points used to describe the model, with more points giving greater accuracy, but also more information to compute. A caveat, however, is that there is a limitation on the size of the spatial step, h (for a given time step, k). That is, there exists an h_{min} , such that $h \geq h_{min}$, which if not respected, will cause instability in the numerical solution. These stability conditions are not described further in detail here, but the reader is referred to [7]. All the sound simulations of the Violino Arpa have been carried out as close to the stability condition as possible, thus giving a full bandwidth solution, i.e. all frequencies up to the Nyquist frequency can be described.

The interesting shape of the Violino Arpa is "sculpted" from a square grid of discretized spatial points to be as close as possible to the real shape. A single string is then added to the system, which is tuned to different frequencies (or musical notes), by adjusting the tension of the string. Fig. 3 shows a snapshot of the displacements of the system in the middle of a simulation. A simulation is carried out for each tuning of the string at a sampling frequency f_S of 44100 Hz, with the bowing position being fixed at $0.29L_s$ from the left edge. The sound is extracted as the vibrations at the bridge location, where both the effects of the vibrating string and the resonant plate are combined.

An overview of all the parameters used in the physical model is given [31].

Finally, the musical notes of the Violino Arpa were extracted from the simulation. These sounds are being used in the 4th scene of the app, where the user can play the Violino Arpa.

3.2 Using the app

The app consists of four scenes that are largely similar in both versions. In the first one, the puzzle level (Fig. 7), the user sees a Violino Arpa broken down to smaller pieces and has to connect them in order to create a complete instrument. With each correct movement a harmonic sound of the Violino Arpa can be heard, and the pieces snap together. When the user tries to stick the piece to the wrong place, a false note is heard. Completion is not mandatory in order to proceed to the next scene; after 15 seconds, the "Next" button appears on the screen, regardless of the success level.

In the second scene, the users have the chance to explore the instrument in further detail (Fig. 1). The Violino Arpa

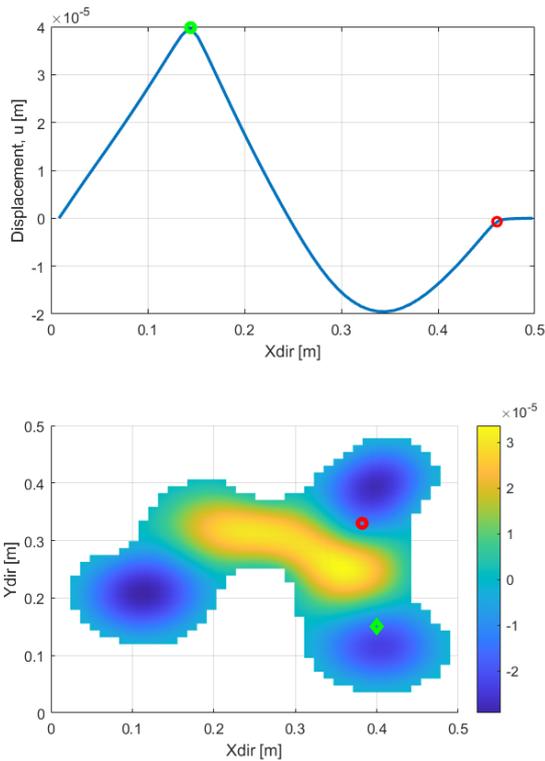


Figure 3. Displacements of the stiff string and the resonant plate in the middle of a simulation (note that the displacement is greatly scaled up). The string is bowed at the location specified by the green circle. The bridge location is highlighted by the red circle. The color gradient specified by the color-bar label gives the measure of displacement of the resonant plate.

1 is now presented in a single piece, with the option to rotate
 2 it in 360 degrees. In the AR version, the instrument gets
 3 larger as the users approach the poster, offering thus the
 4 option to focus in specific parts of it.

5 The third scene consists of a 2D story (Fig. 8) made of
 6 20th century engravings. There, the users can scroll and see
 7 colour being added to the the black-and-white engravings,
 8 while reading some key information about the history of
 9 the Violino Arpa.

10 Finally, in the last scene, the users see a music player,
 11 where each circle represents a note of the instrument. They
 12 can test different notes, and decide for themselves whether
 13 or not they like the sound of it.

14 Both versions are largely similar. The main difference
 15 lies in the first scene, where the lack of actual presence
 16 is compensated with the creation of a 3D space (Fig. 2),
 17 similar to the museum entrance where the AR version can
 18 be used (Fig. 5).

4. EVALUATION

The goal of this evaluation was:

- to test how AR technologies and non-immersive

desktop applications can enhance the overall experience, cultural appreciation, and educational reinforcement of a virtual museum visit, and

- to illustrate the differences in effectiveness and popularity between the two.

This was achieved by exploring the quality of both applications in terms of content, user interaction, and experience, as proposed by [15]. In order to have highly accurate and valid results, the subjects were encouraged to think aloud [35] and have been observed (recorded) while using the applications. Quantitative data were collected from the subjects after performing trials on both applications through questionnaires.

4.1 Participants

A total of 23 subjects, 12-39 years old (Mean:28.4, Std Deviation: 4.73) took part in the research experiment. All participants live in Copenhagen, where the Danish Music Museum is located. Only one of the participants knew and had visited the museum in the past. The selection of the participants was not based in any particular criteria since the study aims to reach people of different backgrounds and ages. All of them had experienced in the past non-immersive applications while less than 30% had experienced AR applications.

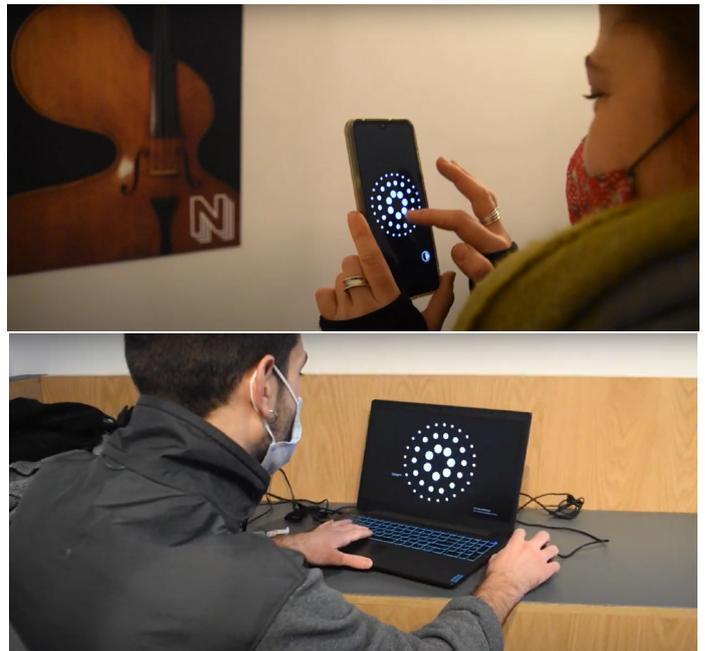


Figure 4. User interacting with the mobile application (top) and online version (bottom). In the AR version, the subject is located at the entrance of the Danish Music Museum where posters of the musical instruments are placed. The participant is playing with the notes of the Violino Arpa.

4.2 Procedure and Task

The participants were split into groups of 2, and were invited to the Danish Music Museum. Before the testing

1 began, they were briefly informed about the idea behind
2 the different versions of the app, and the setup (smart-
3 phone/laptop) was explained. They were encouraged to
4 explore the application without seeking instructions, ex-
5 cept from those given inside the app. It was pointed out
6 that the most essential part of the experiment is to under-
7 stand and decide which version of the application seems
8 more appealing to them.

9 Subsequently, one of them went to the museum entrance
10 to test the AR version using a smartphone (Fig. 5), while
11 the second one went to the reception to test the desktop ver-
12 sion of the app (Fig. 6). When they finished, they switched
13 places, in order to try the other version of the app. Notes
14 of their reactions and comments were taken throughout the
15 entire session.

16 After the testing, all of the participants had to complete a
17 questionnaire, answering about the experience using both
18 of the application versions, and the comparison between
19 them. The experiment duration was 30 minutes for each
20 group.



Figure 5. A screenshot from the application where the participants explore a puzzle with the components of the Violino Arpa.

21 4.3 Measurements

22 Self-reporting measurements were used in combination
23 with mini-interviews, recording, and think-aloud nota-
24 tions.

25 The items of the questionnaire were divided into three
26 categories: Online version, Mobile version, and a compar-
27 ison between the two. The specific terminology was used
28 (Online/Mobile) to describe the two versions of the app, in
29 order to be more understandable and less confusing for the
30 participants. These three sections have the same 9 ques-
31 tions concerning the functionality, efficiency, interactivity,
32 and usability of the application.

33 5. RESULTS

34 The results obtained from the users' experiences are de-
35 picted and analyzed in the following section. The chapter

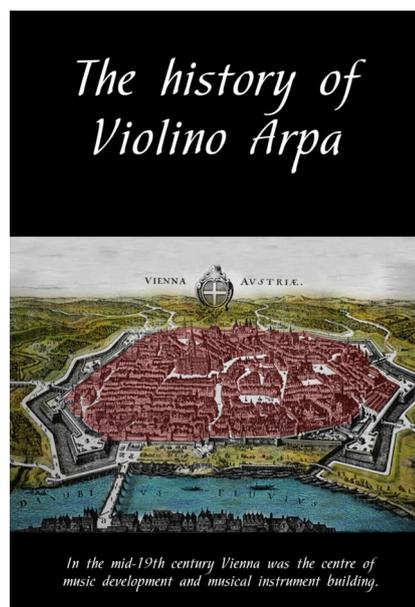


Figure 6. A screenshot from the scene where the participants explore a narrative of the history behind the Violino Arpa.

36 also refers to qualitative data gathered from the open-ended
37 questions in the questionnaire, mini-interviews and the ob-
38 servation of the subjects.

39 5.1 Quantitative Data

40 As an overall review of the data, most participants seem
41 to enjoy both experiences (Fig. 9 part A & B). Over 70%
42 of the subjects felt that both applications, AR and online,
43 were interesting, easy to use and they would like them in
44 the context of a museum. Around 65% of the subjects
45 seemed to be eager to explore the Danish Music Museum
46 after using the applications. The same amount of partici-
47 pants also stated that this experience had some educational
48 benefit. However, the application was lacking in embodi-
49 ment and continuity of narration. Over 60% of the partici-
50 pants stated that it would be easy to access both applica-
51 tions, with regard to the COVID-19 pandemic outbreak. In
52 the last section of the questionnaire the participants were
53 asked to choose between the two different versions of the
54 application. Fig. 9 (part C) visualizes the rest of the gath-
55 ered data. As an overview of the collected information,
56 the Augmented Reality (mobile) application seems to be
57 preferred over the desktop application. The participants'
58 results show that the mobile application was more inter-
59 esting and easier to use, and they would prefer it to be in
60 the context of a museum rather than the online application.
61 No difference was recorded in terms of narration, educa-
62 tional content, and accessibility between the two versions.
63 Around 30% of the participants felt that the mobile appli-
64 cation gives you a better activation of the body use.

65 5.2 Qualitative Data

66 In the following section, the qualitative data that were col-
67 lected are going to be presented. The data was gathered

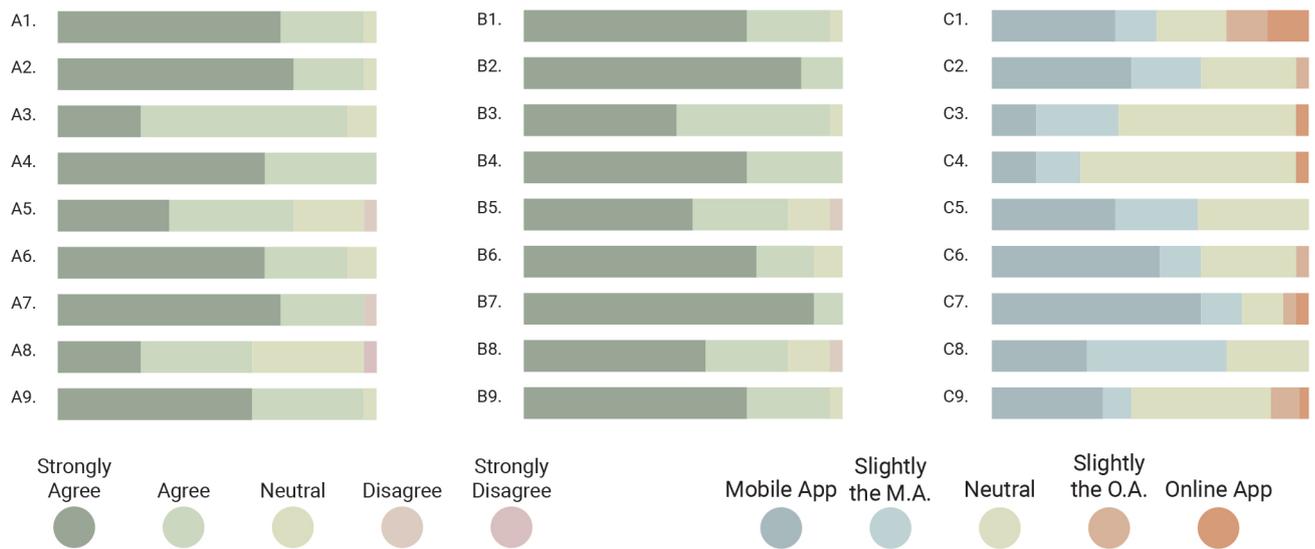


Figure 7. Visualization of the quantitative data that has been obtained from self-report measurements.

1 through the open-ended questions in the questionnaire, the
 2 think-aloud method and observation (following the partic-
 3 ipants' movements and reactions while using the applica-
 4 tions). In order to make it more understandable this section
 5 is divided into two categories: written/oral feedback and
 6 observed actions.

7 5.2.1 Written/Oral Feedback

8 The overall review on the participants' impressions was
 9 very good. The experience of both applications was de-
 10 scribed as "amazing", "interesting", "practical", and "gen-
 11 uine" (especially for the mobile application). A lot of peo-
 12 ple thought that it is a smart way to make people create
 13 new memories, containing an educational aspect that they
 14 will remember. As mentioned before, both applications
 15 were described as a positive experience however some ad-
 16 ditional comments on the technical aspects of the applica-
 17 tions were made. This information was extracted from the
 18 open-ended questions of the questionnaire, namely how
 19 the experiences using each of the apps were different, and
 20 if there were any additional comments.

21 During testing, most users performed similar actions. A
 22 large number of them was not sure what to do when com-
 23 ing across the puzzle pieces and asked for guidance, but
 24 after connecting the first piece thought it was a fun idea
 25 and excitement started building up. There were also many
 26 comments about the music player scene. Participants ex-
 27 pressed a strong will to have the names of the notes stated
 28 next to the circles, so that they would be able to play short
 29 melodies. Also, many of them pointed out that it would be
 30 better if the buttons were activated with touch (in AR) or
 31 hovering of the mouse (in the desktop version) to increase
 32 the continuity of the sound(s). One subject commented on
 33 the look of the player, saying it is too abstract and it looks
 34 more like an optical illusion, rather than something that
 35 could play music.

36 5.2.2 Observed Actions

37 Beside the aforementioned initial uncertainty in the first
 38 scene, it was also common for users of the AR version
 39 to rotate the smartphone instead of the model of the in-
 40 strument, due to the single "Rotate" instruction in the sec-
 41 ond scene, which many of them found too vague. Another
 42 common occurrence, but for users of the online version,
 43 was to try and scroll down in the third scene - the history
 44 of the Violino Arpa. Even though it works, it moves rather
 45 slowly. Therefore, they often had to be told that it's easier
 46 to go through this scene using the mouse buttons. Finally,
 47 the music player was the scene that majority of the partic-
 48 ipants spent most of their time, most of them just playing
 49 and making comments on the quality of the timbre, and
 50 some of them trying to figure out the notes and compose a
 51 melody.

52 6. DISCUSSION

53 The limited number of test subjects affects the accuracy
 54 of the results. To counterbalance that, we drew our initial
 55 results from the qualitative data, which were then substan-
 56 tiated by the quantitative data. The feedback from the qual-
 57 itative data was overall positive, and that is confirmed from
 58 the quantitative data results, which demonstrate a high
 59 level of interest in using both versions of the app. The pref-
 60 erence towards the AR version was markedly higher, with
 61 participants pointing out that the ability to move around
 62 and see the poster is very important. However, a few of
 63 them mentioned that the first scene of the app (creating a
 64 Violino Arpa from floating puzzle pieces) was somewhat
 65 tricky to complete on the small phone screen, and easier
 66 on the computer screen.

67 A recurring comment throughout the testing session was
 68 about the lack of information provided with regard to what
 69 is expected of the user in each scene. Although the inten-
 70 tion was to keep the amount of text minimal in order to

1 allocate more space for the visual elements, it turned out 54
2 that the subjects required more details, and reported that 55
3 both verbally and in writing.

4 Another common comment regarded the interaction dur- 56
5 ing the first scene of the online version. Several partici- 57
6 pants found the way they were moving inside the virtual 58
7 space too similar to popular game [1]. Though unforeseen, 59
8 we ascribe this impression to the first-person camera ap- 60
9 proach. 61

10 Even though we had an overall understanding of the partici- 62
11 pants' overview, some of the results gathered did not 63
12 meet our expectations. Most of the subjects stated that 64
13 the mobile version is more accessible than the online one. 65
14 We were expecting the opposite and we guess the partici- 66
15 pants were biased because the testing for both versions 67
16 took place at the museum grounds, where the poster of the 68
17 Violino Arpa is. The excitement of using the AR version 69
18 on location is likely to have obscured the added accessibil-
19 ity benefits of the online version.

20 7. CONCLUSIONS

21 The present study aimed at enhancing the visitor engage- 73
22 ment in museums, through the use of enticing interac- 74
23 tive applications. A novel application with AR capabili- 75
24 ties for mobile devices was developed alongside another 76
25 one, adapted for desktop computers and the results were 77
26 presented. Both versions of the app were generally well- 78
27 perceived and our findings indicate that a noteworthy per-
28 centage of museum visitors is likely to use such apps dur- 79
29 ing their visit in order to improve their experience and gain 80
30 a better understanding of the exhibits. However, the AR 81
31 version was overall more favoured by a clear margin. 82

32 The mobile application is being used to scan the Danish 83
33 Music Museum advertisement posters outside the actual 84
34 museum itself. As a real-life implementation, the posters 85
35 could be placed anywhere in the city and not only in-situ
36 where the exhibition is. Next to the poster there should be
37 a note for the people to scan the poster and take a small
38 taste of what the museum's actual experience holds.

39 8. REFERENCES

- 40 [1] Achterbosch, L., Pierce, R. & Simmons, G. 2008, 91
41 "Massively multiplayer online role-playing games: 92
42 The past, present, and future", *Computers in Entertain-* 93
43 *ment*, vol. 5, no. 4. 94
44 [2] Adobe Illustrator, "Vector graphics software | 95
45 Create web and mobile graphics to logos, icons, 96
46 book illustrations, product packaging, and bill- 97
47 boards", accessed December 9, 2020, available at 98
48 <https://www.adobe.com/>
49 [3] Adobe Photoshop, "Photo, image & design editing 100
50 software", accessed December 13, 2020, available at 101
51 <https://www.adobe.com/products/photoshop.html> 102
52 [4] Autodesk Maya, "3D computer animation, 103
53 modeling, simulation, and rendering soft-

ware", accessed December 9, 2020, available at
<https://www.autodesk.com/products/maya>

- [5] Bekele, M.K., Pierdicca, R., Frontoni, E., Malinverni, 56
E.S. & Gain, J. 2018, "A survey of augmented, virtual, and mixed reality for cultural heritage", *Journal on Computing and Cultural Heritage*, vol. 11, no. 2. 57
[6] Bell, J.T. and Fogler, H.S., 1995, "The investigation and application of virtual reality as an educational tool", *Proceedings of the American Society for Engineering Education Annual Conferenc*, pp. 1718-1728. 58
[7] Bilbao, S.D., 2009. *Numerical sound synthesis*. John Wiley & Sons. 59
[8] Brown, D. 2007. "Te Ahua Hiko: Digital Cultural Heritage and Indigenous Objects, People, and Environments" in *Theorizing Digital Cultural Heritage*, pp. 77-91. 60
[9] Charitos, D., Lepouras, G., Vassilakis, C., Katifori, V., Charissi, A. & Halatsi, L. 2001, "Designing a virtual museum within a museum", *Proceedings VAST 2001 Virtual Reality, Archeology, and Cultural Heritage*, pp. 284. 61
[10] Chatziaoannou, V. 2019, "Reconstruction of an early viola da gamba informed by physical modeling", *Journal of the Acoustical Society of America*, vol. 145, no. 6, pp. 3435-3442. 62
[11] Dupont, P., Hayward, V., Armstrong, B. & Altpeter, F. 2002, "Single state elastoplastic friction models", *IEEE Transactions on Automatic Control*, vol. 47, no. 5, pp. 787-792. 63
[12] EasyAR SDK, "Motion and surface Tracking using Augmented Reality", accessed December 9,2020, available at <https://www.easyar.com/> 64
[13] Feiner, S., MacIntyre, B., Höllerer, T. & Webster, A. 1997, "A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment", *Personal and Ubiquitous Computing*, vol. 1, no. 4, pp. 208-217. 65
[14] Finkelstein, S., Nickel, A., Lipps, Z., Barnes, T., Wartell, Z. & Suma, E.A. 2011, "Astrojumper: Motivating exercise with an immersive virtual reality exergame", *Presence: Teleoperators and Virtual Environments*, vol. 20, no. 1, pp. 78-92. 66
[15] Finstad, K. 2010, "The usability metric for user experience", *Interacting with Computers*, vol. 22, no. 5, pp. 323-327. 67
[16] Gerval, J.-. & Le Ru, Y. 2015, *Fusion of multimedia and mobile technology in audioguides for museums and exhibitions*. 68
[17] Han, D.I. and Jung, T., 2018, "Identifying tourist requirements for mobile AR tourism applications in urban heritage tourism", *Augmented Reality and Virtual Reality*, pp. 3-20, Springer, Cham. 69

- 1 [18] Hutong Games LLC, "Visual scripting tool for
2 Unity", accessed December 9, 2020, available at
3 <https://assetstore.unity.com/>
- 4 [19] Jørgensen, M.H., Knudsen, A.S., Wilmot, T.M., Lund,
5 K.D., Serafin, S. & Purwins, H. 2015, "A mobile music
6 museum experience for children". NIME, pp. 36-37.
- 7 [20] Mason, C.A. 1996, "Desktop virtual reality in the
8 RAF", IEE Colloquium (Digest), , no. 68.
- 9 [21] Mathworks, "Analyzing data, developing algorithms,
10 creating models", accessed December 9, 2020, avail-
11 able at <https://se.mathworks.com/products/matlab.html>
- 12 [22] Noar, M.D. 1991, "Endoscopy simulation: A brave
13 new world?", Endoscopy, vol. 23, no. 3, pp. 147-149.
- 14 [23] Owen, R., Buhalis, D. & Pletinckx, D. 2005, "Visitors'
15 Evaluations of ICTs Used in Cultural Heritage", Vast,
16 Vol. 2005, pp. 6-13.
- 17 [24] Patterson, R., Winterbottom, M.D. & Pierce, B.J. 2006,
18 "Perceptual issues in the use of head-mounted visual
19 displays", Human factors, vol. 48, no. 3, pp. 555-573.
- 20 [25] Pletinckx, D., Callebaut, D., Killebrew, A.E. & Silber-
21 man, N.A. 2000, "Virtual-reality heritage presentation
22 at Ename", IEEE Multimedia, vol. 7, no. 2, pp. 45-48.
- 23 [26] Rekimoto, J. 1997, "NaviCam: A magnifying glass ap-
24 proach to augmented reality", Presence: Teleoperators
25 and Virtual Environments, vol. 6, no. 4, pp. 399-412.
- 26 [27] Scagliarini, D., Coralini, A., Vecchietti, E., Cinotti,
27 T.S., Roffia, L., Galasso, S., Malavasi, M., Pigozzi,
28 M., Romagnoli, E. & Sforza, F. 2001, "Exciting un-
29 derstanding in Pompeii through on-site parallel interac-
30 tion with dual time virtual models", Proceedings VAST
31 2001 Virtual Reality, Archeology, and Cultural Her-
32 itage, pp. 83.
- 33 [28] Serafin, S., 2004. The sound of friction: Real time
34 models, playability and musical applications (Doctoral
35 dissertation, Department of Music, Stanford Univer-
36 sity).
- 37 [29] Skarlatos, D., Agrafiotis, P., Balogh, T., Bruno, F., Cas-
38 tro, F., Petriaggi, B.D., Demesticha, S., Doulamis, A.,
39 Drap, P., Georgopoulos, A., Kikillos, F., Kyriakidis, P.,
40 Liarokapis, F., Poullis, C. & Rizvic, S. 2016, Project
41 iMARECULTURE: Advanced VR, immersive serious
42 games and augmented reality as tools to raise aware-
43 ness and access to European underwater cultural her-
44 itage.
- 45 [30] Stowell, D., Robertson, A., Bryan-Kinns, N. &
46 Plumbley, M.D. 2009, "Evaluation of live human-
47 computer music-making: Quantitative and qualitative
48 approaches", International Journal of Human Com-
49 puter Studies, vol. 67, no. 11, pp. 960-975.
- 50 [31] Unity Technologies, "Unity Real-Time Development
51 Platform - 3D,2D, VR & AR Visualizations", accessed
52 December 9, 2020, available at <https://unity.com/>
- 53 [32] Vlahakis, V., Ioannidis, N., Karigiannis, J., Tso-
54 tros, M., Gounaris, M., Stricker, D., Gleue, T., Daehne, P. &
55 Almeida, L. 2002, "Archeoguide: An augmented real-
56 ity guide for archaeolog sites", IEEE Computer Graph-
57 ics and Applications, vol. 22, no. 5, pp. 52-60.
- 58 [33] Vongsingthong, S. and Smachet, S. 2014, "Internet
59 of things: a review of applications and technologies",
60 Suranaree Journal of Science and Technology, vol. 21,
61 no. 4, pp.359-374.
- 62 [34] Vukotic, I., Moyse, E., Bianchi, R.M. & ATLAS out-
63 reach group 2020, "ATLASrift - A virtual reality ap-
64 plication", Proceedings of the Division of Particles and
65 Fields 2015, DPF 2015.
- 66 [35] Whitney, P. & Budd, D. 1996, "Think-aloud protocols
67 and the study of comprehension", Discourse Processes,
68 vol. 21, no. 3, pp. 341-351.
- 69 [36] Willemsen, S., Bilbao, S. and Serafin, S. 2019, "Real-
70 time implementation of an elasto-plastic friction model
71 applied to stiff strings using finite-difference schemes",
72 22nd International Conference on Digital Audio Ef-
73 fects.

Appendix B

Friction Drum Paper

REAL-TIME IMPLEMENTATION OF AN ELASTO-PLASTIC FRICTION DRUM USING FINITE DIFFERENCE SCHEMES

Marius George Onofrei

Dept. of Architecture, Design & Media
Technology
Aalborg University
Copenhagen, Denmark
monofr11@student.aau.dk

Silvin Willemsen

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

Stefania Serafin

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

ABSTRACT

Physical modelling sound synthesis is a powerful method for constructing virtual instruments aiming to mimic the sound of real-world counterparts, while allowing for the possibility of engaging with these instruments in ways which may be impossible in person. Such a case is explored in this paper: particularly the simulation of a friction drum. It is an instrument played by causing the membrane of a drum head to vibrate via friction. This typically involves rubbing the membrane via a stick or a cord attached to its center, with the induced vibrations being transferred to the air inside a sound box.

This paper describes the development of a real-time audio application which models such an instrument as a bowed membrane connected to an acoustic tube. This is done by means of a numerical simulation using finite-difference time-domain (FDTD) methods in which the excitation, whose position is free to change in real-time, is modelled by a highly non-linear elasto-plastic friction model. Additionally, the virtual instrument allows for dynamically modifying physical parameters of the model, thereby allowing the user to generate new and interesting sounds that go beyond a real-world friction drum.

1. INTRODUCTION

The friction drum has been described as a peculiar musical instrument or even a noisy toy [1]. In Scandinavian tradition, the friction drum was used in the Middle Ages as a rhythmic instrument. Over time, children used it in the 19th century to play when they went door to door during the Christmas holidays and sang [2]. Figure 1 shows the friction drum present at the Danish Music Museum. As can be seen, the drum is a combination of a stick inserted in the middle and a cylindrical drum. The sound is produced by rubbing the stick up and down the middle of the drum, using a frictional excitation, hence the name friction drum. Friction has been extensively investigated in the sound synthesis literature, being the sound excitation mechanism of several musical instruments such as the violin and the musical saw, but also everyday sounds such as squeaking doors and rubbed wineglasses [3]. The literature has examined different ways of simulating friction, and elasto-plastic friction models have proven to be an accurate way to simulate dry interactions between rubbed surfaces [4]. Such models have also been recently used in combination with finite difference schemes

Copyright: © 2021 Marius George Onofrei et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



Figure 1: A picture of the friction drum present at the Danish music museum.

[5]. In this paper, we combine an elasto-plastic friction model together with a membrane simulation based on FDTD methods to model the interaction between the stick and the drum head in a friction drum. This is furthermore coupled with a 1D wave model with radiating boundary conditions at the open end used to describe the drum's sound box.

We mathematically describe the different elements of the drum, and we present a real-time implementation with using the SENSEL Morph as a controller [6] to play the virtual instrument.

2. FRICTION DRUM MODEL

The friction drum can be modelled by two main components and an excitation mechanism: a membrane connected to an acoustic tube, where the membrane can be bowed via a non-linear elasto-plastic friction model. This section describes the partial differential equations (PDEs) describing these components in isolation.

2.1. Membrane

First off, for a membrane defined over a domain $\mathcal{D}_m = [0, L_x] \times [0, L_y]$, with L_x and L_y being the lengths of the membrane [m] in the Cartesian coordinates (x, y) , the transverse displacement at a

time t [s]: $u(x, y, t)$ [m] can be described by the following PDE:

$$\partial_t^2 u = c_m^2 \Delta u - 2\sigma_{0,m} \partial_t u + 2\sigma_{1,m} \Delta \partial_t u, \quad (1)$$

where the 2D Laplacian operator is defined as:

$$\Delta \triangleq \partial_x^2 + \partial_y^2, \quad (2)$$

and the parameter $c_m = \sqrt{T_m / \rho_m H_m}$ is a measure of wave speed resulting from the membrane's tension per meter T_m [N/m], its density ρ_m [kg/m³] and its thickness H_m [m]. Furthermore, $\sigma_{0,m}$ [s⁻¹] and $\sigma_{1,m}$ [m²/s] are parameters controlling the frequency-dependent and frequency-independent loss respectively.

Dirichlet boundary conditions are assumed for the membrane so that:

$$u = 0, \quad (3)$$

at the edges of the membrane.

2.2. Acoustic Tube

The longitudinal vibration of an air column $\zeta(\chi, t)$ [m] in a tube of uniform cross-section and length L_χ can be described by the following equation:

$$\partial_t^2 \zeta = c_t^2 \partial_\chi^2 \zeta, \quad (4)$$

with the wave speed $c_t = \sqrt{B_t / \rho_t}$ [m/s] resulting from the bulk modulus B_t [Pa] and density ρ_t [kg/m³] of the air inside the tube. $\chi \in [0, L_\chi]$ is a spatial coordinate along the length of the tube.

This 1D wave approximation for a tube holds true if the length scale in the longitudinal direction is significantly greater than in the others. For the case of the friction drum this is not true and this approximation will actually not produce the desired "drum" type sound for the wave speed c_t resulting from the bulk modulus and density of air. However, this value can be tuned to produce the desired sound. The choice for this value is given in Section 4. A more accurate model would be the 3D wave equation but would be too computationally demanding for a real-time implementation.

A Neumann boundary condition is imposed at the side of the tube connected to the membrane, while at the open end of the tube a radiating boundary condition is chosen:

$$\partial_\chi \zeta = 0, \quad (5a)$$

$$\partial_\chi \zeta = -\alpha_1 \partial_t \zeta - \alpha_2 \zeta, \quad (5b)$$

with the constants α_1 and α_2 modelling the inertia and loss at the open end of the tube.

2.3. Connection

A connection between the two components is considered, which means that a connection force is added to the PDEs describing the two separate components in the following way:

$$\partial_t^2 u = c_m^2 \Delta u - 2\sigma_{0,m} \partial_t u + 2\sigma_{1,m} \Delta \partial_t u + \frac{f_{c,m}}{\rho_m H_m} E_m, \quad (6a)$$

$$\partial_t^2 \zeta = c_t^2 \partial_\chi^2 \zeta + \frac{f_{c,t}}{\rho_t A_t} E_t, \quad (6b)$$

where $F_{c,m}$ [N] is the connection force acting on the membrane, while $F_{c,t}$ [N] acts on the tube. A_t [m²] is the area of the tube and the terms E_m [m⁻²] and E_t [m⁻¹] represent some distributions over which the connection force is applied on. Notice that the distributions are given over the appropriate domain, i.e. over a

surface (m²) for the membrane and over a length (m) for the tube. This essentially means that the connection forces applied to the zero-input PDEs given in Equations (1) and (4) are scaled with the mass per unit length or area of each considered component.

For the current model, a rigid connection is assumed meaning that:

$$F_c = F_{c,t} = -F_{c,m}, \quad (7a)$$

$$\eta = \langle u, E_m \rangle - \langle \zeta, E_t \rangle = 0, \quad (7b)$$

with $\langle \cdot, \cdot \rangle$ being an L^2 inner product over the appropriate domain (1D for the tube and 2D for the membrane). Thus η represents the relative displacement of the two components over the connection. A more detailed discussion on the choice of connection distributions is given in the Section 3.

2.4. Excitation - Bowing Model

An elasto-plastic friction model, first proposed in [7], is used for the friction drum excitation mechanism. Such a model, applied to the bowing of a stiff string, was shown by both [8] and [5] to capture realistic details experimentally observed by Smith and Woodhouse, [9] in the bowing of strings: like a hysteresis loop in the bowing force versus relative velocity plane. This was arrived at by both a digital waveguide model, implemented by [8] as well as a FDTD method used by [5], in which the authors presented a working real-time implementation of a bowed stiff string.

The elasto-plastic friction model assumes the contact between two interacting elements is highly irregular at the microscopic level, i.e. not all the overlapping surface is actually in contact. Instead, the contact can be modelled via a large group of bristles each contributing to the total friction force. These bristles are modelled as damped stiff springs and therefore each generates increasing contact force with increasing displacement, describing an elastic regime (stick). However, each bristle can only displace so far before it "breaks" and not all bristles "break" at the same time. This represents the elasto-plastic regime of the friction model, where only some bristles have reached the breaking point (slide). Once all bristles "break", a completely plastic regime is entered (slip). In the case of the bowed membrane, after the slip a "new" portion of the bow gets in contact with the membrane and the stick-slide-slip cycle restarts.

Adding the bowing force to the membrane can be done by introducing an extra term to (6a), the PDE which governs the membrane connected to the tube:

$$\partial_t^2 u = \dots - \frac{f_b(v, z)}{\rho_m H_m} \delta(x - x_B, y - y_B). \quad (8)$$

Again the force is applied over some distribution which is in fact a single point on the Cartesian grid of the membrane given by the bowing position at some time t : $(x_B(t), y_B(t))$. This is achieved by the use of a 2D Dirac delta function $\delta(x - x_B, y - y_B)$ [m⁻²]. Furthermore, the force needs to be scaled with the mass per unit area of the membrane of the component. As for the bowing force itself, F_b [N], it is a function of the relative velocity between the bow and the membrane, v [m/s], and the average bristle displacement, z [m]:

$$F_b(v, z) = s_0 z + s_1 \dot{z} + s_2 v + s_3 w, \quad (9)$$

where v can be computed as the difference between the velocity of the membrane at the bowing location and the externally supplied

velocity of the bow, $v_B(t)$ [m/s]:

$$v = \partial_t u(x_B, y_B) - v_B. \quad (10)$$

Furthermore, s_0 is the bristle stiffness [N/m], s_1 is the damping coefficient of the bristles [kg/s] and s_2 is the viscous friction [kg/s]. s_3 [N] is a force coefficient proportional to the normal bowing force $F_N(t)$ (which is an external input and can vary over time) scaled with a pseudorandom function $w(t) \in [-1, 1]$ and is used to add noise to the total bowing force, as per [3]. The time derivative of the average bristle displacement, \dot{z} [m/s] is given by:

$$\dot{z} = r(v, z) = v \left[1 - \alpha(z, v) \frac{z}{z_{ss}(v)} \right]. \quad (11)$$

Perhaps the most important function in this elasto-plastic friction model is introduced above: the adhesion map $\alpha(z, v)$ which controls the transition between the various regimes of friction. The function is defined as follows and is illustrated in Figure 2 (a.):

$$\alpha(v, z) = \begin{cases} 0 & |z| \leq z_{ba}, \text{sgn}(v) = \text{sgn}(z) \\ \alpha_m(v, z) & z_{ba} < |z| < |z_{ss}(v)|, \text{sgn}(v) = \text{sgn}(z) \\ 1 & |z| \geq |z_{ss}(v)|, \text{sgn}(v) = \text{sgn}(z) \\ 0 & \text{sgn}(v) \neq \text{sgn}(z) \end{cases} \quad (12)$$

where z_{ba} [m] is the breakaway bristle displacement, below which the friction regime is purely elastic. Indeed, when $\alpha = 0$, it follows that $\dot{z} = v$. Then, when z_{ba} is surpassed, the elasto-plastic regime is entered where the value of z will be a proportion, governed by $\alpha_m(v, z)$, of the steady-state bristle displacement $z_{ss}(v)$, with:

$$\alpha_m = \frac{1}{2} \left[1 + \text{sgn}(z) \sin \left(\pi \frac{z - \text{sgn}(z) \frac{1}{2} (|z_{ss}(v)| + z_{ba})}{|z_{ss}(v)| - z_{ba}} \right) \right]. \quad (13)$$

At steady-state, when slipping occurs and therefore $\dot{z} = 0$, $\alpha = 1$ and together with Equation (11) it follows that $z = z_{ss}(v)$, with $z_{ss}(v)$ [m] being defined as:

$$z_{ss}(v) = \frac{\text{sgn}(v)}{s_0} \left[F_C + (F_S - F_C) e^{-(v/v_S)^2} \right] \quad (14)$$

where the Coulomb force $F_C = \mu_C F_N$ [N] and stiction force $F_S = \mu_S F_N$ [N] are given as a proportion of the normal force, $F_N(t)$ of the bow acting on the membrane. These proportions are controlled by the dimensionless dynamic and static friction coefficients respectively, μ_C and μ_S . Looking at Figure 2 (b.) which illustrates $z_{ss}(v)$ for a fixed F_N one can see that the Stribeck effect, i.e. the dip of force at low velocities is captured, as the magnitude of z_{ss} at values of v close to zero is larger than for higher relative velocities. This allows for a larger total friction force to be obtained in this region before the plastic regime is reached. After slipping occurs, the "grip" of the bow on the membrane is briefly lost and the membrane displaces in the opposite direction, hence $\text{sgn}(v) \neq \text{sgn}(z)$ and α becomes again zero, meaning that the bow again sticks to the membrane and a new stick-slip cycle begins.

2.5. Complete System

The complete system for the friction drum can be therefore written in continuous time as:

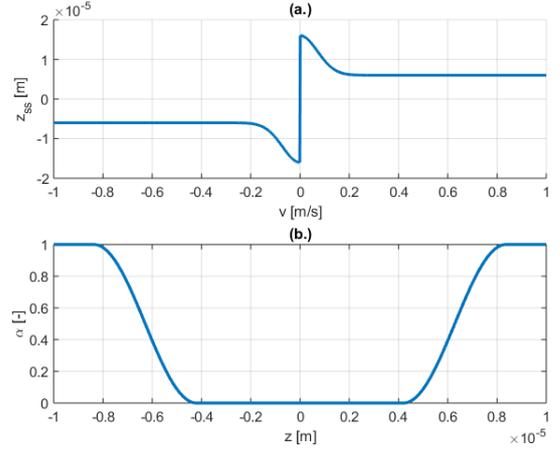


Figure 2: (a.) Steady-state bristle displacement $z_{ss}(v)$ for a constant normal force F_N . (b.) A plot of the adhesion map $\alpha(v, z)$ plotted against z when the signs of v and z are the same.

$$\begin{cases} \partial_t^2 u = c_m^2 \Delta u - 2\sigma_{0,m} \partial_t u + 2\sigma_{1,m} \Delta \partial_t u + \\ \quad + \frac{F_c}{\rho_m H_m} E_m - \frac{F_b(v, z)}{\rho_m H_m} E_b \\ \partial_t^2 \zeta = c_t^2 \partial_\chi^2 \zeta - \frac{F_c}{\rho_t A_t} E_t, \end{cases} \quad (15a)$$

$$(15b)$$

3. DISCRETIZATION

The system given in (15) is discretized using FDTD methods, which subdivides the continuous model into grid points in space and samples in time. The (x, y) -plane of the membrane is discretised as $x = lh_m$ and $y = mh_m$, with $l \in [0, \dots, N_x]$ and $m \in [0, \dots, N_y]$. Here, $N_x = L_x/h_m$ and $N_y = L_y/h_m$ are the horizontal and vertical number of grid intervals the membrane is divided in with grid spacing h_m [m]. For simplicity the same spacing is used in both directions. Similarly for the tube, $\chi = ph_t$, where $p \in [0, \dots, N_\chi]$ and $N_\chi = L_\chi/h_t$ is the total number of grid intervals along the tube's length with a grid spacing h_t [m]. Time t is discretized as $t = nk$ where $k = 1/f_s$, sampling frequency f_s [Hz] and temporal index $n \in \mathbb{N}$.

An important thing to take into account when it comes to numerical models is the issue of stability, from which limitations arise on the possible size of the grid spacing h_m and h_t . Stability conditions are available for each individual component and will be presented in the upcoming subsections. Working with grid spacings that satisfy the stability conditions as close to equality as possible ensures a more accurate numerical scheme.

Using these discrete definitions for space and time, the continuous state variables presented in the previous section can then be approximated by grid functions as $u(x, y, t) \approx u_{l,m}^n$ for the membrane and $\zeta(\chi, t) \approx \zeta_l^n$ for the tube. Furthermore, approximations

to the derivatives can be described in the following way:

$$\partial_t u \approx \delta_t u_{l,m}^n = \frac{1}{2k} (u_{l,m}^{n+1} - u_{l,m}^{n-1}), \quad (16a)$$

$$\partial_t u \approx \delta_{t-} u_{l,m}^n = \frac{1}{k} (u_{l,m}^n - u_{l,m}^{n-1}), \quad (16b)$$

$$\partial_t^2 u \approx \delta_{tt} u_{l,m}^n = \frac{1}{k^2} (u_{l,m}^{n+1} - 2u_{l,m}^n + u_{l,m}^{n-1}), \quad (16c)$$

$$\partial_x^2 u \approx \delta_{xx} u_{l,m}^n = \frac{1}{h_m^2} (u_{l+1,m}^n - 2u_{l,m}^n + u_{l-1,m}^n), \quad (16d)$$

$$\partial_y^2 u \approx \delta_{yy} u_{l,m}^n = \frac{1}{h_m^2} (u_{l,m+1}^n - 2u_{l,m}^n + u_{l,m-1}^n), \quad (16e)$$

$$\Delta u \approx \delta_{\Delta} u_{l,m}^n = \delta_{xx} u_{l,m}^n + \delta_{yy} u_{l,m}^n, \quad (16f)$$

$$\mu_t u \approx u_{l,m}^n = \frac{1}{2} (u_{l,m}^{n+1} + u_{l,m}^{n-1}), \quad (16g)$$

$$\mu_t - u \approx u_{l,m}^n = \frac{1}{2} (u_{l,m}^n + u_{l,m}^{n-1}), \quad (16h)$$

$$\partial_t \zeta \approx \delta_t \zeta_p^n = \frac{1}{2k} (\zeta_p^{n+1} - \zeta_p^{n-1}), \quad (16i)$$

$$\partial_t \zeta \approx \delta_{t-} \zeta_p^n = \frac{1}{k} (\zeta_p^n - \zeta_p^{n-1}), \quad (16j)$$

$$\partial_t^2 \zeta \approx \delta_{tt} \zeta_p^n = \frac{1}{k^2} (\zeta_p^{n+1} - 2\zeta_p^n + \zeta_p^{n-1}), \quad (16k)$$

$$\partial_x^2 \zeta \approx \delta_{\chi\chi} \zeta_p^n = \frac{1}{h_t^2} (\zeta_{p+1}^n - 2\zeta_p^n + \zeta_{p-1}^n). \quad (16l)$$

With these definitions in place we can move on to discretize the individual components of the friction drum model.

3.1. Membrane

The complete membrane including the bowing force and connection to the tube can be discretized as

$$\begin{aligned} \delta_{tt} u_{l,m}^n &= c_m^2 \delta_{\Delta} u_{l,m}^n - 2\sigma_{0,m} \delta_t u_{l,m}^n + 2\sigma_{1,m} \delta_{t-} u_{l,m}^n + \\ &+ \frac{F_c^n}{\rho_m H_m} J_m - \frac{F_b^n(v^n, z^n)}{\rho_m H_m} J_b(x_B, y_B) \end{aligned} \quad (17)$$

with J_m and $J_b(x_B, y_B)$ being spreading operators. The former is a discretized version of the connection distribution E_m and the latter, a discrete 2D Dirac delta function which defines the bowing position in the continuous model. Here we use a first order 2D spreading function defined as

$$J_b = \frac{1}{h_m^2} \begin{cases} (1 - \alpha_{x_B})(1 - \alpha_{y_B}) & l = l_B, m = m_B \\ (1 - \alpha_{x_B})\alpha_{y_B} & l = l_B, m = m_B + 1 \\ \alpha_{x_B}(1 - \alpha_{y_B}) & l = l_B + 1, m = m_B \\ \alpha_{x_B}\alpha_{y_B} & l = l_B + 1, m = m_B + 1 \\ 0 & \text{otherwise,} \end{cases} \quad (18)$$

with $l_B = \text{floor}(x_B/h_m)$, $m_B = \text{floor}(y_B/h_m)$, $\alpha_{x_B} = x_B/h_m - l_B$ and $\alpha_{y_B} = y_B/h_m - m_B$. This spreading function, necessary for exciting a discretized grid has a dual: the interpolation function I_b which is of interest when obtaining the state of a discrete grid between grid points and will be of use further along for the discretization of the complete system. See [10] for more details on this. These two functions are related as such:

$$J_b = \frac{1}{h_m^2} I_b. \quad (19)$$

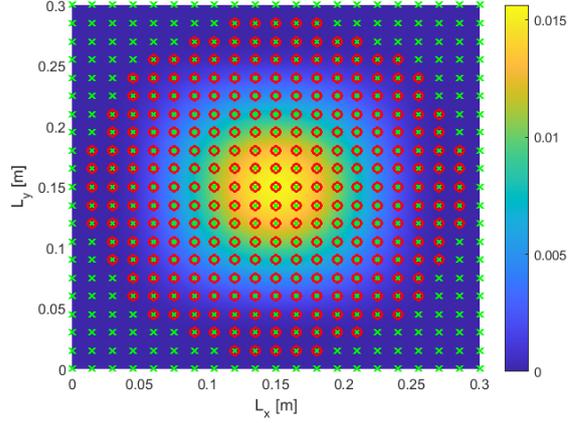


Figure 3: Circular grid approximation from a rectangular grid and the normalized Hann distribution used for the connection to the tube, I_m . The green crosses are the original grid points from the square $L_x \times L_y$ grid, while the red circles are the points used in the calculation.

Drum heads are typically of circular shape and although the membrane is defined in Cartesian coordinates over some rectangle of length $L_x \times L_y$, one can "sculpt" a circular grid using a staircase approximation, as done in [11], as long as boundary conditions are satisfied. Since Dirichlet conditions are assumed, the only thing needed is that points on the rows and columns at the edge of the square grid need to be fixed to zero. Regarding the connection with the tube, it is clear that the entire membrane contributes to movement of the air column inside the tube. However, there is a factor which point towards skewing the weight of the membrane displacements towards its center. Due to the boundary layer effect, the air at the edges of the tube will be semi-stationary. Therefore a 2D Hann distribution over 72.25% of the area of the grid is used, centered at the middle of the membrane. This is illustrated in Figure 3 together with the grid points of the circular membrane approximated from the initial rectangular grid.

This connection distribution, named I_m , is normalized such that its integral is equal to 1 and can be seen to act as an interpolation function acting on $u_{l,m}^n$. Therefore its dual spreading function J_m will be defined in the same way as Equation (19).

Notice in Equation (17) the use of the δ_{t-} operator in the mixed time/space derivative term, which is used in order to keep the numerical scheme explicit.

Using von Neumann analysis [10], a stability condition can be derived and is given by the following inequality:

$$h_m \geq h_{m,min} = \sqrt{2c_m^2 k^2 + 8\sigma_{1,m} k}. \quad (20)$$

3.2. Acoustic Tube

A discretized version of the acoustic tube and its connection is given by:

$$\delta_{tt} \zeta_p^n = c_t^2 \delta_{\chi\chi} \zeta_p^n - \frac{F_c^n}{\rho_t A_t} J_t, \quad (21)$$

with J_t being the spreading operator for the connection force acting on the tube, essentially the discretized version of E_t . This is

related to its dual interpolant function I_t in the following way:

$$J_t = \frac{1}{h_t} I_t. \quad (22)$$

with I_t taken as a normalized half-Hann window spread over 4% of the length of the tube, with its peak at the connection point (the top of the acoustic tube). This was preferred due to a Dirac type connection in order to dampen out some of the high frequencies which would result from an excitation of the tube at a single point and thus produce a more realistic friction drum sound. Notice in Equation (22) that h_t is not squared, as was the case for h_m in Equation (19). This is due to the different spatial dimensions of the components.

The boundary conditions of the tube presented in Equation 5 are discretized in the following way:

$$\delta_\chi \cdot \zeta_p^n = 0, \quad \text{at } p = 0 \quad (23a)$$

$$\delta_\chi \cdot \zeta_p^n = -\alpha_1 \delta_t \cdot \zeta_p^n - \alpha_2 \mu_t \cdot \zeta_p^n, \quad \text{at } p = N_\chi, \quad (23b)$$

and a stability condition on the grid size h_t is given by [10]:

$$h_t \geq h_{t,min} = c_t k. \quad (24)$$

3.3. Connection

The rigid connection given in Equation (7b) can be discretized as

$$\eta^n = \langle u_{l,m}^n, J_m \rangle_{\mathcal{D}_m} - \langle \zeta_p^n, J_t \rangle_{\mathcal{D}_t} = 0, \quad (25)$$

with \mathcal{D}_m and \mathcal{D}_t being the domains of the membrane and of the tube respectively. If the equality in Equation (25) is true at sample n then it follows that it will be true at sample $n + 1$ as well. This together with the following identity will provide valuable information for solving the complete discretized system:

$$\langle f, J \rangle_{\mathcal{D}} = I f, \quad (26)$$

where f is a grid function in some domain \mathcal{D} and I and J must be dual interpolation and spreading functions. This results in the following equality:

$$I_m u_{l,m}^{n+1} = I_t \zeta_p^{n+1}. \quad (27)$$

3.4. Excitation - Bowing Model

For the bowing force, the discrete counterpart of Equation (9) is taken as:

$$F_b^n(v^n, z^n) = s_0 z^n + s_1 r^n + s_2 v^n + s_3 w^n, \quad \text{with:} \quad (28)$$

$$r^n(v^n, z^n) = v^n \left[1 - \alpha(z^n, v^n) \frac{z^n}{z_{ss}(v^n)} \right]. \quad (29)$$

Additionally, the relative velocity between the bow and the membrane described in Equation (10) will be:

$$v^n = I_b \delta_t \cdot u_{l,m}^n - v_B^n. \quad (30)$$

3.5. Solving the System

In order to calculate the update values for the grid functions: $u_{l,m}^{n+1}$ and ζ_p^{n+1} , three unknown variables must first be determined: v^n , z^n and F_c^n and for this we need a system of three equations dependent on these variables at each sample n , which can then be solved using a multivariate Newton-Raphson method. An interesting observation is that the reaction of the air inside the acoustic tube, i.e. the connection force, will instantaneously affect the bowing force and vice-versa. This would not be the case in a simpler model where bowing would not occur at the connection point.

The first function $g_1(v^n, z^n, F_c^n)$ can be found by making use of the following identity:

$$\delta_{tt} u_{l,m}^n = \frac{2}{k} (\delta_t \cdot u_{l,m}^n - \delta_{t-} u_{l,m}^n), \quad (31)$$

and introducing it together with Equation (30) in Equation (17), which results in:

$$g_1(v^n, z^n, F_c^n) = I_b J_b \frac{F_b^n(v^n, z^n)}{\rho_m H_m} - I_b J_m F_c^n + \left(\frac{2}{k} + 2\sigma_{0,m} \right) v^n + q^n = 0, \quad (32)$$

with

$$q^n = -\frac{2}{k} \delta_{t-} I_b u_{l,m}^n + 2\sigma_{0,m} v_B^n + \frac{2}{k} v_B^n - c_m^2 I_b \delta_\Delta u_{l,m}^n - 2\sigma_{1,m} \delta_{t-} I_b \delta_\Delta u_{l,m}^n.$$

The second equation needed is, as per [5] and [10]:

$$g_2(v^n, z^n) = r^n - a^n = 0, \quad \text{with} \quad (33)$$

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

where the operators applied to z^n describe the trapezoid rule.

Finally, the third equation comes from the rigid connection condition in Equation (27). The displacements of the membrane $u_{l,m}^{n+1}$ and for the tube ζ_p^{n+1} can be extracted and expressed only in terms of values at current or previous samples by expanding the operators in Equations (17) and (21). This results in

$$g_3(v^n, z^n, F_c^n) = F_c^n k^2 \left[\frac{I_m J_m}{(1 + \sigma_{0,m}) \rho_m H_m} + \frac{I_t J_t}{\rho_t A_t} \right] - F_b^n(v^n, z^n) k^2 \frac{I_m J_b}{(1 + \sigma_{0,m}) \rho_m H_m} + b^n = 0, \quad (34)$$

with

$$b^n = \frac{1}{1 + \sigma_{0,m} k} [c_m^2 k^2 I_m \delta_\Delta u_{l,m}^n + 2\sigma_{1,m} k (I_m \delta_\Delta u_{l,m}^n - I_m \delta_\Delta u_{l,m}^{n-1}) + 2I_m u_{l,m}^n - (1 - \sigma_{0,s} k) I_m u_{l,m}^{n-1}] - [c_t^2 k^2 I_t \delta_\chi \zeta_p^n + 2I_t \zeta_p^n - I_t \zeta_p^{n-1}] \quad (35)$$

The following iteration is then used to calculate the unknown values v^n , z^n and F_c^n :

$$\begin{bmatrix} v_{c,(i+1)}^n \\ z_{c,(i+1)}^n \\ F_{c,(i+1)}^n \end{bmatrix} = \begin{bmatrix} v_{c,(i)}^n \\ z_{c,(i)}^n \\ F_{c,(i)}^n \end{bmatrix} - \begin{bmatrix} \frac{\partial g_1}{\partial v} & \frac{\partial g_1}{\partial z} & \frac{\partial g_1}{\partial F_c} \\ \frac{\partial g_2}{\partial v} & \frac{\partial g_2}{\partial z} & \frac{\partial g_2}{\partial F_c} \\ \frac{\partial g_3}{\partial v} & \frac{\partial g_3}{\partial z} & \frac{\partial g_3}{\partial F_c} \end{bmatrix}^{-1} \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} \quad (36)$$

where i is the iteration number. The threshold for convergence is set at 10^{-7} , with a maximum number of iterations of 99.

Once the three values at the sample n are known, update values for the grid points $u_{l,m}^{n+1}$ and ζ_p^{n+1} can be found by expanding the operators in Equation (17) and Equation (21).

Name	Symbol [unit]	Value
Membrane		
Length	L_x [m]	0.3
Width	L_y [m]	0.3
Material Density	ρ_m [kg/m ³]	1400
Thickness	H_m [m]	0.007
Wave Speed	c_m [m/s]	$15 \leq c_m \leq 150$
Freq. dep. loss	$\sigma_{0,m}$ [s ⁻¹]	$0 \leq \sigma_{0,m} \leq 6$
Freq. indep. loss	$\sigma_{1,m}$ [m ² /s]	$0 \leq \sigma_{1,m} \leq 0.0026$
Grid spacing	h_m [m]	0.0167
Acoustic Tube		
Length	L_χ [m]	0.4
Area	A_t [m ²]	0.0707
Material Density	ρ_t [kg/m ³]	1.225
Wave Speed	c_t [m/s]	30
Radiation Damp. Ct.	α_1 [s·m ⁻¹]	0.008
Radiation Damp. Ct.	α_2 [m ⁻¹]	4.348
Grid spacing	h_t [m]	0.0011
Bowing Model		
Coulomb Friction	μ_C [-]	0.3
Static Friction	μ_S [-]	0.8
Normal Force	F_N [N]	$0 \leq F_N \leq 20$
Bow Velocity	v_B [m/s]	$0 \leq v_B \leq 0.2$
Stribeck Velocity	v_S [m/s]	0.1
Bristle Stiffness	s_0 [N/m]	10^5
Bristle Damping	s_1 [kg/s]	$0.001\sqrt{s_0}$
Viscous Friction	s_2 [kg/s]	4
Noise Coefficient	s_3 [N]	$0F_N \leq s_3 \leq 0.04F_N$
Pseudorandom Fct.	w [-]	$-1 \leq w \leq 1$
Breakaway Disp.	z_{ba} [m]	$0.7f_C/s_0$
Other		
Sample Rate	f_s [Hz]	44100
Time Step	k [s]	$1/f_s$

Table 1: Parameter values used for the friction drum simulation.

4. IMPLEMENTATION

The implementation of the finite difference scheme presented in Section 3 has been carried out in C++ using the JUICE framework [12] and a demonstration video is available at [13]. The parameters used can be found in Table 1, and have been chosen starting from the work of Serafin [3] and Willemsen et al. [5], but tuning them to achieve the desired sound for the instrument.

4.1. Prototype Model Results

Before implementing the audio application in JUICE, tests were done in MATLAB [14] to identify that the model is stable and that results are in line with expectations. Figure 4 shows a snapshot of the circular membrane being bowed with $F_N = 12$ [N] and $v_B = 0.1$ [m/s] coupled with the acoustic tube at some time step in the middle of a simulation. Looking at the tube, one can see the free and the radiating boundaries at its endpoints.

The next step was to test whether the vibrations of the membrane exhibit the Helmholtz motion, which is typical for bowed instruments and tends to produce triangular-shaped wave forms. Figure 5a shows the displacements at a location on the membrane

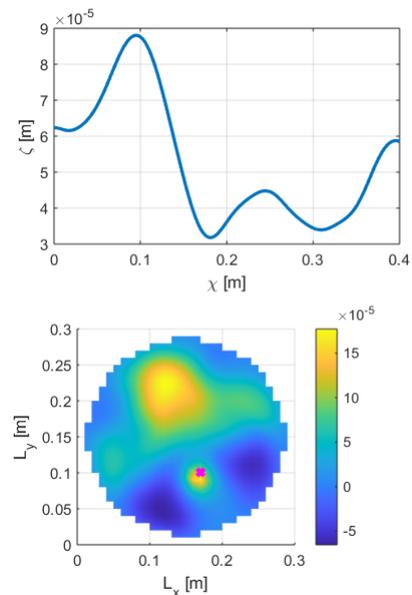


Figure 4: Snapshot showing the displacements of the friction drum’s components at a time step in the middle of a bowing simulation, top being the longitudinal displacements of the air column in the acoustic tube ζ and bottom being the transverse displacements of the membrane u . The magenta cross highlights the bowing position.

during a simulation, as well as the relative velocity and the resulting displacements at the open end of the tube. The membrane indeed shows a Helmholtz motion, while the relative velocity exhibits the stick-slip behavior with values hovering around zero followed by an abrupt drop after which a new portion of the bow sticks again and the cycle restarts. The wave form of the displacements at the open end of the tube is somewhat more complex and highlights the effect of the tube which distorts the triangular shape of the membrane’s displacement, due to interference.

Finally, the presence of a hysteresis loop in the force vs. relative velocity is investigated, which is an expected behavior as per experimental observations of bowed strings by Woodhouse and Smith [9]. This is illustrated in Figure 5b.

4.2. Real-time Application

Figure 6 shows snapshots of the friction drum audio application during use, where due to variation of the bowing position and force/velocity different modes of vibration are in resonance. Notice the different opacities of the bowing position square.

An important part in designing the real-time application was to have a natural type of interaction. Since there are 4 dimensions of input to the model, i.e., the bowing position $(x_B(t), y_B(t))$, bowing force $F_N(t)$ and velocity $v_B(t)$, it was desired to find a way to somehow control all these inputs simultaneously. An ideal match for this task was the Sensel Morph which is a tablet-sized pressure sensitive controller which is very fast and extremely sensitive [6]. The work of [15] provided an open source library for allowing easy communication between the Sensel and JUICE. This allowed to map the (x, y) touch position to the bowing position

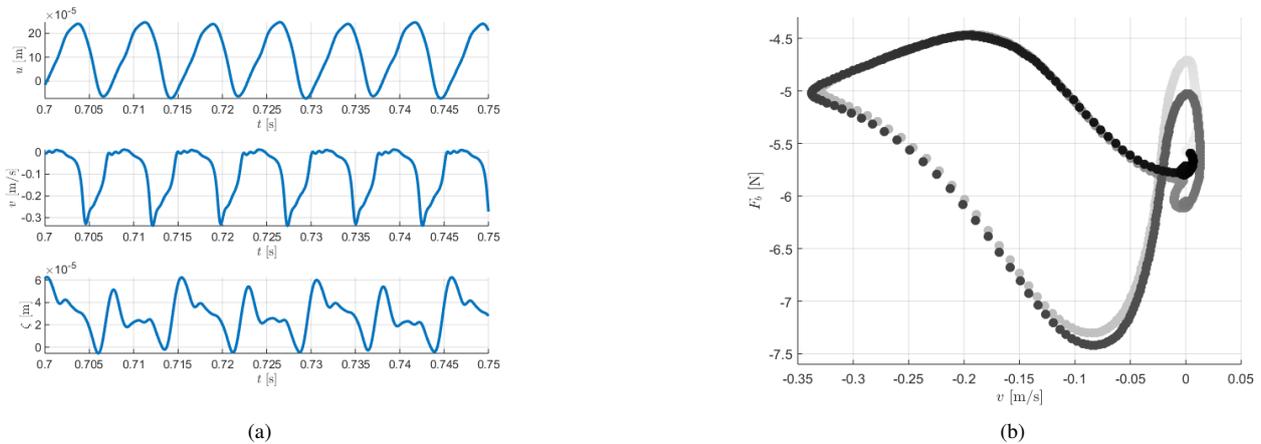


Figure 5: (a) A plot of the displacement wave forms at the membrane, u (top) and the tube, ζ (bottom) together with the relative velocity, v between the bow and the membrane at the bowing location (middle). (b) Hysteresis loop showing 700 points of relative velocity v and bowing force F_b , going from light to dark with increasing samples.

while the pressure was mapped to the bowing force and velocity, linearly coupled. The normal force is limited in the range of $F_N \in [0, 20]$ [N] while the bowing velocity is mapped in the range $v_B \in [0, 0.2]$ [m/s]. Naturally, the bowing position is limited in the range of $[0, L_x]$ and $[0, L_y]$. Of course this means one can bow in the edge of the square grid which is outside the circular membrane's calculation points but in that case, nothing will happen.

Other parameters which can be modulated via sliders are the tuning of the membrane, i.e. the wave speed $c_m \in [15, 150]$ [m/s], which is named "Tuning" thus allowing for a more intuitive understanding of the parameter by a non-technical user. Similarly the damping parameters $\sigma_{0,m} \in [0, 6]$ [s^{-1}] and $\sigma_{1,m} \in [0, 0.00266]$ [m^2/s] are combined into one value called "Damping". Note, that grid spacing in Eq. (20) is initialised using the highest values for c_m and $\sigma_{1,m}$ so that the stability condition is not violated. Also note, that even when the damping parameters are set to 0 the radiation damping parameters for the tube α_1 and α_2 are fixed. Hence, even with zero damping, there will still be decay present. A third slider at the top of the graphical user interface (GUI) window controls the $s_3 \in [0F_N, 0.04F_N]$ [N] term in the bowing force, and is called "Noise" as it adds some white noise to the friction force proportional to the normal force F_N .

Furthermore, a vibrato effect is added where one can modulate via a sine wave the tuning of the membrane by a chosen frequency and with a chosen amount. This is introduced in the GUI as the sliders, named "Variation" which adds an oscillation between $[0, 3]$ [m/s] to the wave speed c_m and "Rate", which controls oscillation frequency of the sine wave and is in the range $[0, 10]$ [Hz].

All the ranges mentioned above are mapped in the GUI to be in a $[0, 10]$ non-dimensional scale as to not confuse the user with different scales.

To add to the natural feel of the interaction another important addition is included in the GUI: the vibration of the membrane is plotted in real-time in a gray scale, inspired by [15], together with the bowing position, plotted with an orange color and an opacity given by the amount of pressure one applies to the Sensel.

The output sound is retrieved from the model by following the state of the open end of the tube (ζ_{R_x}) and amplified to the usual

range of amplitudes $[-1, 1]$. Since the amplitudes of the model states are higher when using a lower for c_m , an adjustable gain is used.

5. EVALUATION

Because of COVID-19 restrictions and the consequent impossibility to gather test subjects, the real-time simulation presented in the previous section was demoed by the first author during a Zoom session with 17 students enrolled in a physical modelling for sound synthesis class as part of the Master education in Sound and Music Computing at Aalborg University Copenhagen.

After a demo where the different parameters of the interface were explored, a qualitative interview and discussion took place.

To the question regarding which instrument it was, the answers were varied. One student said the sounds were inspired by the Theremin, another mentioned a gong, hand drum, metallic drum, bowed bar, low frequency saw, a cymbal that is "contact-miced" and bowed or even "a chair being dragged across the floor". Particularly, one student mentioned that the instrument sounded like you were "inside" the instrument or that it resembles the sounds a contact microphone might pick up. This was encouraging to hear as the sound is being picked up right at the end of the tube so in some way the listener is inside the drum itself.

All in all, the answers give some indications on how the sonories of the physical model reminds of a bowed inharmonic resonator and references to friction were abundant in the students' responses, as the "dragged chair" might suggest. Even if it was not possible for the viewers to play with the interface, they found the use of the Sensel intuitive and the sound produced felt natural.

This informal evaluation is obviously not ideal; the feeling of the quality of the instrument is better experienced from the viewpoint of the player. Nonetheless, it provided some indications for further development of the interaction and the GUI associated with the instrument.

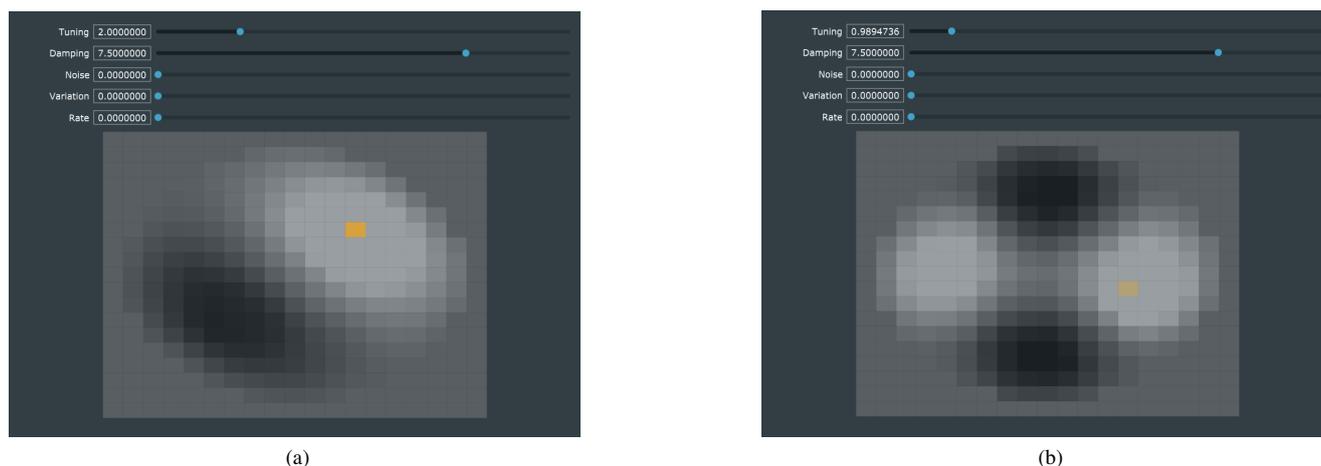


Figure 6: A screenshot of the real-time audio application where a resonance occurs with (a) mode 2 of vibration and (b) mode 3.

6. CONCLUSION

In this paper, the development of a real-time audio implementation of a friction drum using physical modelling has been presented. FDTD methods are used for simulating the friction drum as a bowed membrane connected to an acoustic tube. Furthermore, an advanced elasto-plastic friction model is used which is shown to exhibit physically consistent behavior observed in experiments on real music instruments such as the presence of a hysteresis loop in the resulting bowing force-velocity plane.

Future work may involve the investigation of other possible mappings to the various parameters, the use of different type of controllers and perhaps extending the model to a stereo output, with the output being picked-up at various locations inside the tube or on the membrane. Another important direction should be the optimization of the C++ implementation with the aim of reducing the grid size intervals in the numerical model and working closer to the stability conditions which would produce results of broader bandwidth. Additional optimization may allow for the application to be implemented on a micro-controller or single-board computer and developed as a stand-alone digital instrument.

7. REFERENCES

- [1] H. Balfour, “The friction-drum,” *The Journal of the Royal Anthropological Institute of Great Britain and Ireland*, vol. 37, pp. 67–92, 1907.
- [2] M. Müller, “Folk-folkelig-folkelige musikinstrumenter i danmark,” *Folk og kultur, årbog for dansk etnologi og folke-mindevidenskab*, vol. 23, no. 1, pp. 5–20, 1994.
- [3] S. Serafin, *The sound of friction: Real time models, playability and musical applications*, Ph.D. thesis, Department of Music, Stanford University, 2004.
- [4] 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. 1073–1081, 2005.
- [5] 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 *22nd International Conference on Digital Audio Effects*, 2019.
- [6] Sensel Inc., “Sensel morph,” Available at <https://sensel.com/>, Accessed: 2021-03-27.
- [7] P. Dupont, V. Hayward, B. Armstrong, and F. Altpeter, “Single state elastoplastic friction models,” *Automatic Control, IEEE Transactions on*, vol. 47, pp. 787 – 792, 06 2002.
- [8] S. Serafin, F. Avanzini, and D. Rocchesso, “Bowed string simulation using an elasto-plastic friction model,” 2003.
- [9] J.H. Smith and J. Woodhouse, “Tribology of rosin,” *Journal of The Mechanics and Physics of Solids - J MECH PHYS SOLIDS*, vol. 48, pp. 1633–1681, 08 2000.
- [10] S. Bilbao, *Numerical Sound Synthesis*, John Wiley and Sons, Ltd, 2009.
- [11] B. Hamilton, *Finite Difference and Finite Volume Methods for Wave-based Modelling of Room Acoustics*, Ph.D. thesis, The Univeristy of Edinburgh, 2016.
- [12] JUCE ROLI, “Juce,” Available at <https://juce.com>, Accessed: 2021-03-27.
- [13] M. G. Onofrei, “Friction drum physical modelling simulation demo,” <https://www.youtube.com/watch?v=3UEUyYc6PF0>.
- [14] MATLAB, *version (R2020b)*, The MathWorks Inc., Natick, Massachusetts, 2019.
- [15] 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 Conference*, 2019.