Physical Modelling of Musical Instruments for Real-Time Applications

Ph.D. Dissertation

Silvin Willemsen

Dissertation submitted March 10, 2021

Thesis submitted:	March 10, 2021
PhD Supervisor:	Prof. Stefania Serafin Aalborg University
PhD Committee:	Prof. X, Y University Prof. X, Y University Prof. X, Y University
PhD Series:	Technical Faculty of IT and Design, Aalborg University
Department:	Department of Architecture, Design and Media Technology

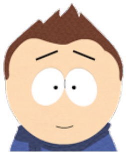
ISSN: xxxx-xxxx
ISBN: xxx-xx-xxxx-xxx-x

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

Printed in Denmark by Rosendahls, 2021

Curriculum Vitae

Silvin Willemsen



Here is the CV text.

Abstract

English abstract

Resumé

Danish Abstract

Contents

Curriculum Vitae **iii**

Abstract **v**

Resumé **vii**

Contents **ix**

Preface **xv**

I Introduction 1

1 Physical Modelling of Musical Instruments **3**
1.1 Physical Modelling Techniques 3
1.2 Thesis Objectives and Main Contributions 4
1.3 Thesis Outline 4

2 An Introduction to FDTD Methods **5**
2.1 The Mass-Spring System 6
2.2 The 1D Wave Equation 7
2.3 Energy Analysis 7
2.4 Stability Analysis 7

II Resonators 9

3 Stiff string **13**
3.1 Adding Losses 13

4 2D Systems **15**
4.1 2D Wave Equation 15
4.2 Thin plate 15

4.3 Stiff membrane 15

5 Brass **17**

III Exciters **19**

6 Unmodelled Excitations **23**
6.1 Initial conditions 23
6.2 Signals 23
6.2.1 Pulse train 23
6.2.2 Noise 23

7 Modelled Excitations **25**
7.1 Hammer 25
7.2 The Bow 25
7.2.1 Static Friction Models 25
7.2.2 Dynamic Friction Models 25
7.3 Lip-reed 26
7.3.1 Coupling to Tube 26

IV Interactions **27**

8 Connections **31**
8.1 Rigid connection 31
8.2 Spring-like connection 31

9 Collisions **33**
9.1 Classic models 33
9.2 Michele's tricks 33

V Dynamic Grids **35**

10 Dynamic Grids **37**
10.1 Background and Motivation 37
10.2 Method 37
10.2.1 Cubic interpolation 37
10.2.2 Sinc interpolation 37
10.2.3 Displacement correction 38
10.3 Analysis and Experiments 38
10.3.1 Interpolation technique 38
10.3.2 Interpolation range 38

10.3.3 Location 38
10.4 Discussion and Conclusion 38

VI Real-Time Implementation and Control 39

11 Real-Time Implementation 43

12 Control 45
12.1 Sensel Morph 45
12.2 Phantom OMNI 45

VII Complete Instruments 47

13 Large Scale Modular Physical models 51
13.1 Bowed Sitar 51
13.2 Dulcimer 51
13.3 Hurdy Gurdy 52

14 Tromba Marina 53
14.1 Introduction 53
14.2 Physical Model 53
14.2.1 Continuous 53
14.2.2 Discrete 53
14.3 Real-Time Implementation 53
14.3.1 Control using Sensel Morph 53
14.3.2 VR Application 53

15 Trombone 55
15.1 Introduction 55
15.2 Physical Model 55
15.2.1 Continuous 55
15.2.2 Discrete 55
15.3 Real-Time Implementation 55

References 57

VIII Papers 59

A Paper Errata 61

Todo list

Different title here? 23
check if is still true 37

Preface

Starting this PhD, I did not have a background in mathematics, physics or computer science, which were three equally crucial components in creating the result of this project. This is why I decided to write this thesis a bit more pedagogical than what could be expected. As I felt that the literature lacks a lot of intuition I wanted to give that to the reader. Some basic calculus knowledge is assumed.

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

Make physical modelling more accessible to the non-physicist.

Interested in physically impossible manipulations of now-virtual instruments.

Silvin Willemsen
Aalborg University, March 10, 2021

Part I

Introduction

Chapter 1

Physical Modelling of Musical Instruments

The history of physical modelling of musical instruments

Exciter-resonator approach.

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

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

1.1 Physical Modelling Techniques

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

1.2 Thesis Objectives and Main Contributions

The main objective of this thesis is to implement existing physical models in real time using FDTD methods. Many of the physical models and methods presented in this thesis are taken from the literature and it is thus not Secondly, to combine the existing physical models to get complete instruments and be able to control them in real time.

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

1.3 Thesis Outline

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

Notes

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

One over number $\rightarrow$ reciprocal of number

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

Chapter 2

An Introduction to FDTD Methods

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

Differential equations

As mentioned in Chapter 1 differential equations are used to describe the motion of dynamic systems. A characteristic feature of these equations is that, rather than the absolute position (or displacement) of an object, the time derivative of its position – its velocity – or the second-order time derivative – its acceleration – is described. From this, the displacement of the system can be computed.

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

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

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

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

all of which mean a second-order derivative with respect to time t, i.e., u's acceleration. In this document, the operator notation will be used.

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

Now that an equation has been established, how

Discretisation using FDTD methods

Finite-difference time-domain (FDTD) methods essentially subdivide a continuous equation in discrete points in time and space, a process called *discretisation*.

Once an ODE or PDE is discretised using these methods it is now called a *Finite-Difference Scheme* (FDS).

Implementation

In the following

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

Something something newton's second law

2.1 The Mass-Spring System

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

Continuous-time

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

$$\frac{d^2 u}{dt^2} = -\omega_0^2 u \tag{2.1}$$

Discrete-time

The , u is approximated using

$$u(t) \approx u^n \tag{2.2}$$

where $t = nk$

Implementation

2.2 The 1D Wave Equation

The simplest and arguably the most important PDE in the field is the 1D wave equation

Continuous-time

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

$$\partial_t^2 u = c^2 \partial_x^2 u. \tag{2.3}$$

Discrete-time

Boundary Conditions

When a system is distributed in space,

Output sound

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

2.3 Energy Analysis

Debugging physical models.

2.4 Stability Analysis

Finding stability condition

Part II

Resonators

Resonators

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

- Bars and Stiff Strings
- 2D Models
- Brass

Chapter 3

Stiff string

Stiff string and stuff

Boundary conditions need to be extended...

3.1 Adding Losses

Chapter 4

2D Systems

4.1 2D Wave Equation

4.2 Thin plate

4.3 Stiff membrane

Chapter 5

Brass

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

Part III

Exciters

Exciters

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

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

Chapter 6

Unmodelled Excitations

Different title here?

6.1 Initial conditions

Hammer

Full raised cosine

Pluck

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

6.2 Signals

6.2.1 Pulse train

For brass

6.2.2 Noise

Noise input

Chapter 7

Modelled Excitations

7.1 Hammer

Hammer modelling

7.2 The Bow

The bow...

7.2.1 Static Friction Models

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

7.2.2 Dynamic Friction Models

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

7.3 Lip-reed

Lip-reed model

7.3.1 Coupling to Tube

Part IV

Interactions

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

Newtons third law (action reaction)

Chapter 8

Connections

Something about connections

8.1 Rigid connection

8.2 Spring-like connection

Chapter 9

Collisions

Something about collisions

9.1 Classic models

9.2 Michele's tricks

Part V

Dynamic Grids

Chapter 10

Dynamic Grids

10.1 Background and Motivation

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

check if is still true

Tension in a string is changed when tuning it

Some artists even use this in their performances [1].

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

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

10.2 Method

Iterations have been:

- Interpolated boundary conditions
- Linear interpolation

10.2.1 Cubic interpolation

10.2.2 Sinc interpolation

10.2.3 Displacement correction

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

$$\begin{aligned} u_M^{n+1} &= 2u_M^n + \lambda^2(u_{M-1}^n - 2u_M^n + u_{M+1}^n) - K\left(u_M^n - \frac{u_M^n + w_0^n}{2}\right) \\ w_0^{n+1} &= 2u_M^n + \lambda^2(w_{-1}^n - 2w_0^n + w_1^n) - K\left(w_0^n - \frac{u_M^n + w_0^n}{2}\right) \end{aligned} \tag{10.1}$$

$$\begin{aligned} u_M^{n+1} &= 2u_M^n + \lambda^2(u_{M-1}^n - 2u_M^n + u_{M+1}^n) + \frac{K}{2}(w_0^n - u_M^n) \\ w_0^{n+1} &= 2u_M^n + \lambda^2(w_{-1}^n - 2w_0^n + w_1^n) - \frac{K}{2}(w_0^n - u_M^n) \end{aligned} \tag{10.2}$$

with $K = K(\alpha)$

$$K = (1-\alpha)^\epsilon. \tag{10.3}$$

10.3 Analysis and Experiments

10.3.1 Interpolation technique

10.3.2 Interpolation range

10.3.3 Location

... where to add and remove points

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

10.4 Discussion and Conclusion

Part VI

Real-Time Implementation and Control

Real-Time Implementation and Control

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

Chapter 11

Real-Time Implementation

JUCE Give overall structure of code

Implementation of the physical models using FDTD methods

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

Usually, `MATLAB` is used for simulating

Here, we define an application as being real-time when

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

Also the application needs to be controlled continuously

Chapter 12

Control

12.1 Sensel Morph

150 Hz

12.2 Phantom OMNI

Part VII

Complete Instruments

Complete Instruments

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

Chapter 13

Large Scale Modular Physical models

In the paper "Real-Time Control of Large-Scale Modular Physical Models using the Sensel Morph" [2] we presented a modular physical modelling environment using three instruments as case studies.

13.1 Bowed Sitar

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

13.2 Dulcimer

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

13.3 Hurdy Gurdy

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

Chapter 14

Tromba Marina

14.1 Introduction

14.2 Physical Model

14.2.1 Continuous

14.2.2 Discrete

14.3 Real-Time Implementation

14.3.1 Control using Sensel Morph

14.3.2 VR Application

Chapter 15

Trombone

15.1 Introduction

15.2 Physical Model

Most has been described in Chapter 5

15.2.1 Continuous

15.2.2 Discrete

15.3 Real-Time Implementation

Unity??

References

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

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

Part VIII

Papers

Paper A

Paper Errata

Here, some errors in the published papers will be listed:
Tromba marina paper:

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

DigiDrum:

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

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

- Unit for membrane tension is N/m.