Queen's University of Belfast

Doctoral Thesis

Exploring the Creative Potential of Physically Inspired Sound Synthesis

Author:
Michaël Dzjaparidze
BA, MA

Supervisors:
Dr. Paul Wilson &
Dr. Maarten van Walstijn

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

at the

Sonic Arts Research Centre
School of Creative Arts

SONICARTSRESEARCHCENTRE

August 2015

QUEEN'S UNIVERSITY OF BELFAST

Abstract

Sonic Arts Research Centre
School of Creative Arts

Doctor of Philosophy

Exploring the Creative Potential of Physically Inspired Sound Synthesis

by Michaël DZJAPARIDZE
BA, MA

This thesis accompanies a portfolio of compositions and, in addition, discusses a number of compositional approaches which use physical modelling and physically inspired sound synthesis methods for the creation of electroacoustic music. To this end, a software library has been developed for the purpose of the real-time simulation of systems of inter-connected 1D and 2D objects, which has proven to be indispensable for producing the music works. It should be made clear from the outset that the primary objective of the research was not to add any novel scientific knowledge to the field of physical modelling. Instead, the aim was to explore in depth the creative possibilities of technical research carried out by others and to show that it can be utilised in a form which aids my own creative practice. From a creative perspective, it builds upon concepts and ideas formulated earlier by composers Jean-Claude Risset and Denis Smalley, centred around the interpretation of timbre and sound as constructs which actively inform compositional decision-making and structuring processes. This involves the creation of harmony out of timbre and playing with the source-cause perception of the listener through the transformation of timbre over time. In addition, the thesis offers a discussion of gesture and texture as they commonly appear in electroacoustic music and motivates my own personal preference for focussing on the development of texture over time as a means for creating musical form and function.

Acknowledgements

First off, I would like to thank both of my supervisors, Dr. Paul Wilson and Dr. Maarten van Walstijn for their encouragement, advice and expertise. Paul has stimulated me to think more consciously and confidently about my own creative practice and Maarten's knowledge and advice has helped me immensely in navigating through the technical side of the research.

Furthermore, I would like to thank my family, especially my parents, for supporting me unconditionally and helping me in realising my objectives in life. Also, I'd like to note that my father has helped me considerably during the beginning stages of my research to understand some of the mathematical questions I wasn't able to solve on my own.

I would also like to thank all staff and colleague PhD Students I have met during my stay at SARC. Although I prefer to work on my own and have done so for realising the majority of the portfolio works, their comments on my work are greatly appreciated and I have enjoyed their company.

Lastly, I would like to extend my thanks to three of my former lecturers – Jorrit Tamminga, Marcel Wierckx and Hans Timmermans – from the Hogeschool voor de Kunsten Utrecht. It was their enthusiasm, expertise and inspirational teaching that initially fuelled my interest for sound synthesis, computer programming and compositional practice.

Contents

List of Works

Excite Me (2012)	Sonorities 2013 Festival, Belfast UK
	SuperCollider Symp. 2013, Boulder, Colorado USA
Stable Equilibrium (2013)	Symposium on Acoustic Ecology, Kent UK
Extase 1 (2013)	Sonorities 2014 Festival, Belfast UK
Extase 2, 3 & 4 (2014)	Sounds Alive Festival, Dublin IE
	ImmLib Concert at SARC, Belfast UK
Extase 3 (Version) (2014)	Soundscape Park, Belfast UK
Extase 4 (Version) (2014)	NYC Electroacoustic Music Festival, New York USA
	Soundscape Park, Belfast UK
PM01 (2015)	Forthcoming vinyl release on Will & Ink

List of Figures

Abbreviations

1D	**O**ne **D**imensional
2D	**T**wo **D**imensional
3D	**T**hree **D**imensional
DSP	**D**igital **S**ignal **P**rocessing
DWM	**D**igital **W**aveguide **M**ethod
FDM	**F**inite **D**ifference **M**ethod
FTM	**F**unctional **T**ransform **M**ethod
GPU	**G**raphical **P**rocessing **U**nit
MM	**M**odal **M**ethod
MSN	**M**ass **S**pring **N**etwork
ODE	**O**rdinary **D**ifferential **E**quation
PDE	**P**artial **D**ifferential **E**quation
PM	**P**hysical **M**odelling
VST	**V**irtual **S**tudio **T**echnology

Nomenclature

A	cross-sectional string area	m^2
$\mathbf{A}$	first coefficient matrix	
$\mathbb{A}$	state transition block matrix	
$\bar{\mathbb{A}}$	total state transition block matrix	
$\tilde{A}$	transformed total state transition block matrix	
b_1	frequency independent damping coefficient	
b_2	frequency dependent damping coefficient	
$\mathbf{B}$	second coefficient matrix	
$\mathbb{B}$	input block matrix	
$\bar{\mathbb{B}}$	total input block matrix	
$\tilde{B}$	transformed total input block matrix	
c	wave speed	m/s
$\mathbf{C}$	coupling matrix	
$\mathcal{D}$	general domain	
$\mathbf{D}$	difference operator in matrix form	
$\mathbf{D}_\Delta$	Laplacian operator in matrix form	
$\mathbf{D}_{\Delta,\Delta}$	biharmonic operator in matrix form	
$\mathbf{e}$	column vector denoting input/output distribution	
E	Young's modulus	$\mathrm{N/m}^2$
$\mathbf{E}$	matrix holding input/output distribution vectors in columns	
f	modal frequency	Hz
F	force divided by linear mass density	$\mathrm{m}^2/\mathrm{s}^2$
F_s	sampling rate	Hz
$\mathbf{f}$	force divided by linear mass density column vector	$\mathrm{m}^2/\mathrm{s}^2$
h	grid spacing	

H	membrane thickness	m
$H(z)$	transfer function	
$\mathbf{H}(z)$	transfer function matrix	
I	moment of inertia for bending string vibrations	m^3
$\mathbf{I}$	identity matrix	
k	sampling interval	s
l	horizontal grid index	
L	total string length	m
m	vertical grid index	
$\mathcal{M}$	mass ratio	
n	time index	
N	grid size	
N_q	size of state vector	
N_x	horizontal grid size	
N_y	vertical grid size	
$\bar{N}$	size of total state vector	
$\mathbb{S}$	output block matrix	
$\bar{\mathbb{S}}$	total output block matrix	
$\tilde{S}$	transformed total output block matrix	
t	time	s
u	transverse object displacement	m
$\mathbf{u}$	object displacement as a flattened column vector	m
$\mathbb{U}$	unit domain	
$\mathbb{U}_N$	finite domain	
$\mathbb{U}_\epsilon^2$	unit area rectangular domain	
$\mathbf{v}$	object velocity as a flattened column vector	m/s
$\bar{\mathbf{v}}$	velocity of all objects as a flattened column vector	m/s
$\mathbf{w}$	state vector	
$\bar{\mathbf{w}}$	total state vector	
$\tilde{w}$	transformed total state vector	
x	scaled horizontal spatial location	
y	scaled vertical spatial location	
$\mathbb{Z}$	infinite domain	

γ	scaled wave speed	s^{-1}
δ	difference operator	
δ_Δ	Laplacian operator	
$\delta_{\Delta,\Delta}$	biharmonic operator	
ϵ	aspect ratio	
κ	stiffness parameter	
λ	eigenvalue	
μ	linear mass density	kg/m
ν	Poisson's ratio	
ρ	material density	$\mathrm{kg/m}^3$
ω	angular frequency	rads^{-1}
$\langle\cdot,\cdot\rangle$	inner product	
$\lVert\cdot\rVert$	Euclidian norm	
$\otimes$	cross product	
$\cdot^\top$	transposition	

Chapter 1

Introduction

The main aim of the practical research associated with this thesis is to offer a creatively motivated exploration of sound synthesis methods that are physically inspired. This includes both physical modelling sound synthesis[1] (PM Sound Synthesis) methods as well as other synthesis techniques which are loosely based upon physical principles, but are otherwise firmly routed in the signal processing domain. Research into PM sound synthesis may be roughly divided into two main areas of interest. The first one is the simulation of existing musical instruments; either as a research tool to aid in a better understanding of the physical laws which underly a specific acoustic instrument or as a practical tool for synthesising high quality sound simulations of a particular instrument. The second one is the exploration of new sounds through the creation of abstract virtual instruments which are not necessarily based on an acoustic equivalent. This may be accomplished for instance, by inter-connecting more basic lumped or distributed elements in a (semi-)modular fashion in order to create complicated networks of virtual acoustic objects [1–4] or by creating hybrid instruments which aim at combining the properties of multiple existing instruments [5, 6] in novel ways. As far as the physical modelling part of this research goes, it is the latter field in which i am particularly interested, since the promise of being able to produce novel, complex sound material is very appealing from a compositional point of view. Aside from these matters which focus mainly on the generation of sound, there is the issue of how to offer easy and intuitive control over the creative music making process. Hence, a closely related area of research encompasses

[1] A more qualitative discussion of what physical modelling sound synthesis is exactly and how it is different from other digital sound synthesis methods will be pursued further in chapter 2.

the design and evaluation of dedicated interaction models (both in the form of software-based interfaces [7] as well as tangible digital controllers [8]) to aid in the performability of virtual instruments. The focus of this thesis is on creative aspects, and hence matters of controllability play a role to a certain degree. However, the topic of interaction will not be addressed explicitly, as the main goal of this thesis is to offer a discussion of the composed portfolio works, to place them in a wider artistic context and to illustrate how the development of a custom software library has been integral to their conception.

Although PM sound synthesis has been around for over thirty years now, the amount and quality of creatively oriented research in the form of compositional output seems surprisingly scarce. This can be partly attributed to the fact that the implementation of a PM sound synthesis model requires a certain level of scientific insight and knowledge most composers simply do not have. Similarly, scientific researchers are not likely to possess an equally well-developed ear for musical expressiveness a composer or musician would have. This discrepancy is not exclusively pertinent to the field of PM sound synthesis, but exists in the broader field of music production and sonic arts. One of the initial driving forces behind the conception of IRCAM [9] was exactly to try to bridge this gap by stimulating and facilitating collaborations between musicians and scientists[2]. However, apart from these collaborative endeavours, it seems that there are not many individuals who have successfully managed to balance their time and effort between the fields of musical invention and technical research in a mutually consistent way. Often the bias is either more towards scientific and implementation related matters, thereby neglecting the artistic side, or is mostly focussed towards creative practices (e.g. performance or composition) and consequently only scratches the surface of how to utilise the technology in a well thought out and original manner. Although I certainly acknowledge the importance of technical research, I am nevertheless of the conviction that such research in support of creative practices can only be genuinely successful if the development of the technology is in response to finding a solution for a certain creative problem, or in the words of Pierre Boulez: *"...musical invention must bring about the creation of the musical material it needs; by its efforts, it will provide the necessary impulse for technology to respond functionally to its desires and imagination"* [11, p. 11]. Additionally, I feel that the developed technology needs to be sufficiently accompanied

[2]Although it has been argued that in the specific case of IRCAM, collaborative research between composers and scientific researchers based on genuine mutual equality and respect was, and perhaps still is, an idealised concept rather than reality, due to the rigid hierarchical structure of the organisation and the reverence of the particular view on aesthetics of its leading figures [10].

by creative output, as this is the only way to validate its true artistic merit. Therefore it was decided early on that a portfolio of music works, demonstrating a variety of compositional strategies that utilise physically inspired sound synthesis approaches, should be the main focus in the evaluation of the research rather than the technology which was used to realise these compositions. Hence, the primary objective of the portfolio works is to demonstrate a thorough exploration of creatively utilised physical modelling and physically inspired sound synthesis techniques for the creation of novel sounding electroacoustic music generated entirely by the computer with a discernible character and identity. As such, it is envisioned that it will contribute to the fields of electroacoustic music composition, PM sound synthesis and digital sound synthesis in general, by emphasising the creative potential of technological and scientific constructs that are based upon physical principles.

1.1 Why Physical Modelling?

One question which might be raised immediately is why I have decided to incorporate PM sound synthesis techniques into my compositional practice. In other words: what does PM sound synthesis offer to a composer that other synthesis techniques or methods cannot? Although a possible answer to this question will inevitably be subjective to a large degree, it is a very relevant one to ask. I must confess that at the start of the research this choice was motivated purely by curiosity, since it was a synthesis technique I had very little experience with before and it seemed so different from abstract sound synthesis techniques[3] like FM or additive synthesis. Another aspect which initially appealed to me was being able to synthesise convincing, rich sounds of a certain complexity using scientifically grounded reasoning. As I became more familiar with the technique[4] during the progression of the research, a stronger argument for this choice started to develop.

In order to arrive at a satisfactory answer to the question raised in the previous paragraph, I first of all would like to put forward the argument that theoretically any sound synthesis method may be used to produce an (almost) identical sound. It is useful however, to evaluate how much design effort is required to arrive at a similar sound

[3] With abstract sound synthesis methods are meant those synthesis methods that are not necessarily based on a physically correct reasoning for generating the sound. Also see section 2.2

[4] Both technologically as well as practically speaking.

complexity and if the available synthesis parameters are manageable and intuitive. If the objective is the creation of sounds with a certain physical identity, the claim may be made that a PM sound synthesis approach offers more intuitive control than an abstract sound synthesis approach, as the available parameters for tuning the sound have a direct physical connotation and, hence, make it more obvious what characteristic of the sound might change in response. Take the simple case of synthesising a plucked string sound for instance. A physical model of such a system would contain only a handful of physical parameters (e.g. string tension, stiffness, length). One is able to more easily anticipate how changing these parameters will affect the sound as they can be related directly to perceptible parameters such as pitch, inharmonicity or decay time. Clearly, this offers much more intuitive control than having to manually fine-tune the amplitude and frequencies of hundreds of individual partials when instead using an additive synthesis approach or to have to adjust numerous modulation indices and carrier/modulator ratios in the case of an FM synthesis approach. For more complicated systems, like multiple coupled objects for instance, an abstract sound synthesis approach will become unwieldily and wholly empirical. A physical modelling approach however, although possibly pushing the limits of what is computationally allowable in real-time, will still have the advantage of providing easy and intuitive control by means of just a few physical control parameters. From a purely sonic point of view, PM sound synthesis offers the possibility to create sounds that, while still eluding to something physical, have an almost otherworldly unique, idiosyncratic quality. The ability of being able to interpolate between convincing instrumental sounds and various augmented versions by changing parameters that are routed in the material properties of the modelled object itself can produce unique, expressive sonic results that will be hard, if not impossible, to realise using a signal based approach. A prime example of this are the more adventurous experiments carried out by Orr and Van Walstijn in [12], which, among others, includes the online variation of the mass density of a plate coupled to a string. This allows one to interpolate smoothly from a piano-like tone to a percussive sound for instance. To obtain a similar sound transformation using a signal-based approach would either require an amplitude-based crossfade or some sort of spectral interpolation technique. However, it is not likely that either of these will produce a result that sounds as 'natural' as the physical modelling-based interpolation, simply because in the physical modelling case we are actually interpolating parameters related to the sound generating mechanism itself rather than some abstract representation of the sound.

While PM sound synthesis models are an excellent choice for generating high quality, physically plausible[5] results for sounds which sound generating mechanism obeys certain clearly definable physical laws, there also exist many natural sounds produced by vibratory patterns that are too complex to describe in terms of a fully deterministic theoretical model. This is especially true for natural sounds with very noisy or chaotic characteristics, like the sound of sea waves crashing on a beach or an active thunderstorm for instance. The modelling of these sort of complex natural sounds often will be more easily and convincingly accomplished by basing the modelling on empirical listening experience and loosely defined physical reasoning rather than a rigorously defined underlying physical model [13]. Hence, I have decided to model certain types of physically plausible sounds using a physically inspired, but nonetheless abstract sound synthesis approach. Most noticeable in this respect is the opening section of *Extase 2, 3 & 4*, which features a completely synthetically generated sound scene of waves crashing on a beach. The result has turned out so realistically, that much to my satisfaction, several people were tricked in believing the sounds of the waves were recorded as opposed to synthesised on first listen. Furthermore, by means of experimentation I have found out that interesting results can be obtained by using abstract sound synthesis methods in conjunction with physical modelling techniques. In this case the physical model acts as a means of adding additional coloration to a sound already containing a possibly complex morphology.

1.2 Background and Motivation

Besides trying to expose the creative potential of physically inspired sound synthesis methods for musical invention and the derivation of compositional strategies thereof, I intend to offer the reader of this thesis insight in my creative and compositional practice. Although perhaps a little strange to state in the light of a portfolio of compositions, I have to confess that I find it rather difficult to label myself as a composer. One reason being that besides composing and sculpting sounds, I am very much interested in more technical and scientific matters related to music, sound and art. Therefore, I strive to divide my time and efforts equally between technical and artistic undertakings. My

[5] A more precise definition of what I understand the term 'physically plausible' in the context of sound to mean exactly will be given in section 4.3. For now it will suffice to interpret this as a sound which the listener is likely to link to a physical source or cause.

interest in technology and science is driven by the fact that I have an intrinsic urge to learn and understand how things work on a fundamental level. Specifically related to electronic music composition, I feel rather uncomfortable when being confined to out-of-the-box tools of which I have no clear idea how these operate at a more basic level and which force one to think and operate in a pre-defined and often restricted way. I hardly find this to work creatively stimulating and it is therefore that I have developed a growing interest in the mathematics and physics of digital audio and sound, digital signal processing, computer programming and algorithmic composition techniques.

Another reason for my apprehension of calling myself a composer stems from the fact that my musical background is not rooted in any formal musical education, at least not in the strict traditional sense. Instead my background is in electronic dance music production and dj'ing; a field which in the academic art world is not taken seriously in general and is often deemed inferior compared to the more intellectually informed act of composing non beat-based, experimental music. It is not my intention to start a debate about the validity of this claim. It merely serves as a point to indicate that the stereotypical idea of a composer as somebody who explicitly writes out the music she pursues in the form of traditional notation structures is not something I am familiar with, nor aspire to. For me the act of composition is an intrinsically intuitive process in which sound itself is the vantage point from which to start to explore organisational and structural processes. This is usually accomplished through employing a combination of algorithmically informed and manual editing techniques to build composite structures out of individual layers of sound. Hence, the stages of software development, sound design and musical structuring are often intimately intertwined and feed in and out of each other continuously. More on this will be said in chapter 3, which deals specifically with a discussion of algorithmic composition techniques and the possibly altered role of the composer when it comes to musical decision making and aesthetic form. Rather than offering a comprehensive treatise of this field, my intention is to illustrate to what degree algorithmic principles are used in the composed portfolio works. In addition, I want to come to some sort of mindful conclusion as to how I (currently) position myself in this field from a creative perspective; namely as the entity who ultimately has the deciding hand in determining the shape of the final musical outcome as opposed to relent total creative control to the computer. It is important to stress that the concept of a compositional approach which is based on the organisation and structuring of sound rather than notes is hardly new, and can be traced back to the 1940's with Pierre Schaffer's introduction of 'musique concrète' [14].

Although I do not make use of any formal structuring techniques based on pitch relations, the harmonic dimension does play an important role in my work on a regular basis. However, the harmonic structure and the way this develops throughout a piece always is the result of intuitively guided experimentation (i.e. the (automated) layering of differently pitched sounds and assessing the result by ear) rather than adhering to some formal notational scheme. I find from experience that when dealing with processing recorded sound or sampled material, the issue of how to deal with pitch and harmony is sometimes easier, as this might already be embedded in the sound. As an example, I would like to refer to one of my older works. For *Granular Chaos 1* (2008)[6] I sampled parts from old jazz recordings that subsequently were time-stretched in a nonlinear fashion using an algorithm based on the Lorenz Attractor [15]. The results were layered to create harmonically rich clouds of grains and the final structuring was realised through simple editing techniques using a standard audio editing program. This example serves to illustrate that I was free from having to think consciously about which pitches or harmonic scheme to choose before generating any sounds, as the pitched source material dictated this for me. Producing pitched or harmonic sound material through the use of sound synthesis methods usually requires a little more thought however, since the frequency of an oscillator or filter has to be set explicitly[7]. In an attempt to get around this, I started experimenting with the derivation of pitch information from sound spectra[8] and, taking this a step further even, from spectra obtained from emission lines of the Hydrogen atom[9]. Ever since these first experiments I have become more and more interested in exploring the fusion between timbre and harmony. Hence, this notion plays a dominant role in the majority of the portfolio works. Harmonic structures derived from the timbral qualities of the sound itself are particularly prevalent, as most harmonic schemes are derived directly from the normal mode frequencies associated with the designed system of inter-connected virtual objects used to produce the sound material for each work. As such, there is a strong conceptual link between ideas and

[6]Visit `http://www.michaeldzjaparidze.com/audio.html` if one desires to audition this work and any other older personal works mentioned after.

[7]Leaving aside analysis/synthesis or cross synthesis methods.

[8]My work *t.01.07-01-08* (2008) may be considered as a prime example of this.

[9]I'd like to refer the interesting reader to my MA thesis [16] and preceding publication [17] which deal with this topic specifically.

compositional strategies which have been explored by composer Jean-Claude Risset in the past. Although, stylistically, my music is rather far removed from most of his work, I share his interest in unifying timbre and harmony. This is discussed in more detail in section 4.4.

Another subject which I find interesting from both a practical as well as conceptual point of view is that of transforming sound. On the simplest level for me this implies the generation of different variations of base materials in order to extent ones existing sound palette. Virtually all my former works which feature recorded or sampled materials are built up from differently processed versions of only a couple of source sounds. The rationale behind this is that by restricting myself to a limited amount of source materials, I will be forced to explore the creative possibilities of a particular processing technique more thoroughly in order to acquire variations that sound perceptually different enough, but yet still retain a certain homogeneity among each other. Concurrently, the word transformation suggests that it is some process which is unfolding over time, i.e. a transition between different states. Focussing on this aspect, a sound transformation may be interpreted literally as a sonic event, the perceptual parameters of which are changing over time in a (semi-)continuous fashion. Hence, one may create listening experiences where the listener is constantly challenged to assess if the presented material is real, abstract or somewhere in between these two extremes. In the context of this research, sound transformations are used for exactly this purpose; to create continuous transitions between synthesised sounds which appear to be real, or at least allude to something with a certain level of physicality, to sounds which may be perceived as abstract in the sense that the source or cause, responsible for their creation, is ambiguous. This will be expanded upon in section 4.5, after first having introduced a spectromorphological informed view on timbre that emphasises source-cause relationships rather than equating timbre to harmony or vice versa.

Something that will become apparent when listening to the portfolio, is that I have a strong interest in the development of texture over time as a compositional construct, rather than focussing on gesture as a means to create a causal relationship between successive events. In this respect I think it is fair to say that my compositional output and musical preference is influenced heavily by composers of ambient and drone music and 'neo-modernist' sound art [18]. This has mainly to do with the fact that I enjoy listening to the gradual development of the timbral aspects of sound and music over time, rather than being able to link individual musical gestures together in order to trace the music linearly through time as is common practice in many more gesturally oriented compositions. Chapter 5 expands on this and also serves to illustrate more clearly how the development of texture is reflected in the portfolio and the role the employed physically inspired sound synthesis methods play in this regard.

1.3 Structure of the Thesis

The reader has probably noted by now that I have decided to structure the thesis according to a number of themes, rather than presenting a chronological discussion of the composed portfolio works. The primary reason for this is to present to the reader with a clear overview of the technical and artistic choices I have made during the research and to expose the link that exists between the technical and creative aspects of the research. Each chapter deals with a general discussion of a specific theme, after which follows an illustration of how the portfolio addresses specific issues or interprets certain creative concepts sketched out in these discussions. The only exception to this is chapter 2. This chapter is concerned with providing a basic introduction to commonly employed PM sound synthesis techniques, contains a review of relevant PM sound synthesis software, gives a concise overview of the developed software library *PMLib* and concludes with an introductory tutorial on how to use *PMLib* for setting up, computing and simulating systems of inter-connected resonator objects. As such, it is the most technical chapter of this thesis. Chapter 3 acts as a transition chapter between the technical and creative sides of the research, whereas chapters 4 and 5 focus primarily on creative and artistic matters that have influenced me during the composition process and how these are ingrained in the portfolio works. Chapter 6 summarises the portfolio works and discusses the correlation between them. The thesis concludes with a short discussion on the issues that merit future investigation. Ultimately, I am convinced that the structure I have decided upon will make for a more interesting read, as it will demonstrate more clearly how my work fits into the wider context and, equally important, will show the unique, personalised way I have linked technical and creative research.

1.4 Contributions to the Field

As mentioned at the start of this chapter, the contribution this portfolio makes is situated in the creative field rather than the technical field. Although a large part of the investigation involved various technical matters[10], the creative use of those constructs to produce musically interesting results is what I have been concerned with most. As far as any contributions to the field of musical invention go, I want to be careful not to make any grand claims as firstly, this, by definition, is a field where subjectivity is an important factor and secondly, influences from other composers and musical works are always prevalent. However, what I am confident about is that I have developed a greater awareness of my own compositional practice and know better now where to situate it with respect to other composers of electroacoustic music. The portfolio demonstrates that I have a particular interest in the internal development of texture as the basis for musical structuring first and foremost and that the particular approach to PM sound synthesis, in conjunction with a creative use of other abstract sound synthesis methods and algorithmically informed sound production techniques, has allowed me to create unique, rich and dynamically varying sound textures which would have been difficult to create in any other way. Although I associate strongly with certain conceptual constructs formulated by more traditional composers of electroacoustic music (Jean-Claude Risset and Denis Smalley in particular), I feel a closer connection with music and composers who focus on a gradual exposition of the internal properties of sound by means of continuation and who manage to craft a fine balance between noise[11] and tonal elements, like Stephan Mathieu for instance. A recurring compositional construct in the portfolio is the use of sound transformations to create listening experiences which play with the source-cause perception of the listener in a time-varying manner. In itself this is nothing revolutionary. However, in the case of this portfolio it is interesting to note that everything is completely synthesis based, and hence a different set of possibilities is explored in comparison to various sound processing based transformations, as many of them involve the modification of sound generating parameters directly. Similarly, the simulation of environmental sound by means of synthesis also is not in any way new or

[10] E.g. research into different PM sound synthesis techniques and development of software and sound synthesis algorithms.

[11] In this regard, the word noise should not be interpreted as an unwanted or unpleasant sound, but rather has to be thought of as a sound that does not have an obvious tonal element, but nonetheless can present itself as valid musical material.

groundbreaking, but at the same time, to my knowledge, is not commonly utilised for compositional purposes.

Although this research doesn't involve any contributions directed to the advance of the field of PM sound synthesis, the particular approach followed is unique in the sense that it combines elements from various different physical modelling paradigms in order to be able to simulate in real-time complex systems of inter-connected objects in a relatively straightforward way, using an object-oriented design approach with the help of the SuperCollider and Python programming languages. Several physical modelling UGens exist within SuperCollider[12]. However, my approach makes accessible a unique array of new possibilities for SuperCollider users for exploring the timbral complexity of custom build virtual instruments through the inter-connection of simpler building blocks. The functionality of this library needs to be refined and can be extended, but I still expect that it will prove useful and inspirational to some. In light of assessing my research this is largely irrelevant, as it is the musical value of the portfolio which should be judged first and foremost. Nonetheless, I hope that whoever is assessing my work, recognises the effort and time I have put in developing the software part. The time invested in this clearly has proved to be of great value, as I feel it has allowed me to work in an unrestricted and musically stimulating way, which hopefully is reflected in the musical quality of the composed works. This then remains the most important part in judging my research efforts; that the music will be appreciated for what it is, independent of any technical aspects or conceptual presumptions.

[12]Most of them are ported form the STK [19].

Chapter 2

An Overview of PM Sound Synthesis and the Developed Software

This chapter is dedicated to providing more background information on a custom SuperCollider and Python based software library, used for the physical modelling of systems of inter-connected resonator objects by means of a hybrid finite difference/modal approach, called *PMLib*. This library has formed the foundation for the majority of the generated sound materials, but has also informed many of the developed compositional strategies. This chapter will first commence with a motivation for developing the software library, which will lead to the formulation of a number of creatively oriented criteria. This will then be followed by a concise discussion of some key concepts and commonly used terminology in the field of PM sound synthesis, without straying into too much technical and scientific detail. It is hoped that as such it provides the reader with enough basic understanding to follow PM sound synthesis related matters in this and the subsequent chapters. For a mathematical treatment of the followed approach and the numerical implementation details of the software library, the reader is referred to appendix A of this thesis. The chapter concludes with a general overview of the developed software and a tutorial which will demonstrate the creative possibilities of *PMLib* in a more detailed fashion.

2.1 Motivation and Criteria for Developing the Software

As stated at the start of chapter 1, my primary interest in using PM sound synthesis was for exploring new sound worlds as opposed to focussing on recreating the sound of a particular instrument as accurately as possible. In order to be able to accomplish this in an efficient and, for me, creatively stimulating way, a decision had to be made regarding what tools would prove to be most suitable. After investigating the already available options (a selection of which will be reviewed in section 2.3), it was concluded that in order to allow for the most creative freedom and efficient workflow during the composition stage, it would be more advantageous to develop a custom software tool for synthesising physically modelled sounds. One may argue that in the light of a music composition PhD, devoting significant time on matters not directly related to the compositional process is dangerous, since one could end up spending a considerable amount of time on the developmental stage without harvesting enough creative results. Although this certainly is something to be aware of, I would like to note that likewise, one could run the risk of wasting valuable time on learning how to work efficiently with existing tools. In addition, existing solutions are not likely to be tailored to one's own specific wants and needs. Consequently, it might prove to be necessary to adapt one's preferred way of working according to this, which could lead to additional frustration and creative blockages. Hence, it is my belief that the development of my own tools has ultimately resulted in a more deeply informed and intelligent exploration of them and, in response, has produced more personalised and therefore more interesting creative results. The following creatively motivated research aims and criteria were found to be particularly important with regards to the implementation, design and use of the software tool to be developed:

- The ability to generate a relatively wide variety of spectrally rich sounds with a dynamic timbral profile in an efficient manner.
- The ability to design arbitrary complex virtual instruments by connecting more basic elements in a relatively easy and efficient manner[1].

[1] It should be noted that in the context of this research the term *virtual instrument* does not correspond to an existing musical instrument necessarily, but rather to an abstract musical instrument (i.e. it might not have a direct acoustic counterpart) in the form of a system of inter-connected resonator objects.

- The ability to use the spectral description of a particular virtual instrument in musical meaningful ways, e.g. to determine the harmonic structure of musical gestures.
- The ability to generate high quality sound output in real-time.
- The ability to merge the design stage, the sound sculpting stage and the composition stage within a single environment so that one can easily switch between these different stages.
- The ability to change model parameters and audition any perceptual changes while running the simulation.

The above set of criteria has directly influenced the particular choice of the PM sound synthesis method that was employed for the developed software tools and the programming environment in which everything has been implemented as will be addressed in more detail in section 2.5. Furthermore, it is strongly believed that this particular choice has made it possible to produce a wide variety of different types of sounds and made it easier to write a substantial amount of music in a limited amount of time. It has to be duly noted again however, that in light of this research it is the compositional output which should be assessed first and foremost rather than any developed software, as judging a portfolio of compositional works makes more sense practically with regards to the main aim of this research. Furthermore, it is important to mention that it is in no way claimed that the developed software is adding any new scientific or technological knowledge to the field of PM sound synthesis. Instead, it was my aim to use existing PM sound synthesis research and frame it in a way so as to make it useful for expressing my personal artistic and aesthetic goals. This is not to say that the software part should be regarded as non-relevant completely, as the process of its development has directly influenced the employed compositional strategies and it is hoped that it might prove of practical use for others at some point.

2.2 PM Sound Synthesis

PM sound synthesis is a technique which produces sound by directly simulating the sound generating mechanism of an acoustical object or physical action. This physically-based foundation is absent in so-called abstract [20] or sound-based [21] synthesis methods. As such PM sound synthesis relies on a well-defined mathematical formulation of the underlying physical problem to be solved. Usually this is done in terms of one or more (coupled) partial differential equations (PDEs) with particular boundary conditions. The complete problem description is denoted as a boundary-value problem, which describes the dynamics of the system at any spatial point in the domain of definition $\mathbf{x} = [x_1, x_2, \ldots, x_{N-1}, x_N] \in \mathcal{D}^N$ and at any time $t \geq 0$. To make this more concrete, let us consider the example of a single 1D object, so that $\mathbf{x} = x$, in the form of a stiff string with frequency dependent and independent damping parameters. This may be described by the PDE [22]

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} - \kappa^2 \frac{\partial^4 u}{\partial x^4} - 2b_1 \frac{\partial u}{\partial t} + 2b_2 \frac{\partial^3 u}{\partial x^2 \partial t}. \tag{2.1}$$

Here the dependent variable $u(x,t)$ denotes the transversal direction of string vibration, $x \in \mathcal{D} = [0, L]$ is a spatial position along the length of the string, the constant c is the wave speed[2] in absence of the stiffness and loss terms, κ is a stiffness parameter giving rise to frequency-dependent wave velocity or dispersion and the coefficients b_1 and b_2 give rise to frequency independent and dependent damping respectively. The application of the operators $\partial^n/\partial t^n$ and $\partial^n/\partial x^n$ results in the n^{th} partial derivative of the dependent variable u with respect to time and space. In absence of the last three terms on the right hand side, equation 2.1 reduces to the 1D wave equation describing transverse wave motion on an ideal string [23].

Various different methods exist for arriving at a solution to equation 2.1 which are suitable for a direct implementation on a computer. Attempts to compare and evaluate different methods have been carried out by a number of people [21, 24–27], some of which are more complete and objective than others. Each technique has its own specific advantages and disadvantages and in general the optimal choice for a particular method depends on the specific context and the practical aspects one deems most important. A

[2] Higher values of c are the result of either more applied string tension or a smaller cross-sectional string area which will increase the pitch of the sound produced by the vibrating string.

comprehensive comparison of all methods to PM sound synthesis is outside the scope of this thesis and instead, a concise evaluation of three established PM sound synthesis techniques will be given. Attention is directed in particular to their prospective usefulness with regards to the criteria for the software which was to be developed as previously outlined in section 2.1.

2.2.1 Digital Waveguide Method

Compared to abstract sound synthesis methods, PM sound synthesis is computationally expensive in general. An exception to this rule is the 1D wave equation which can be simulated very efficiently with the digital waveguide method (DWM) by utilising the travelling wave solution formulated by D'Alembert (1747). In short, D'Alembert discovered that the solution to the wave equation can be expressed as a superposition of two waves travelling in opposite directions, which may be expressed mathematically as

$$u\left(x,t\right) = u_r\left(t - \frac{x}{c}\right) + u_l\left(t + \frac{x}{c}\right), \tag{2.2}$$

where u_r and u_l represent the waves travelling to the right and left respectively. A digital implementation, popularised and developed mainly by J.O. Smith [28] in the late eighties, consists of discretising these travelling wave solutions by sampling them every T seconds, where T is the sampling period or equivalently, the reciprocal of the sample rate f_s. The physical string displacement (or alternatively string velocity or force) at a specific location along the string may then be obtained by summing the output of a bi-directional delay line representing the discretised waves. In order to realise more complex, realistic structures, effects of damping and stiffness may be incorporated by introducing filters in the delay line structure. However, this may complicate a practical implementation significantly and furthermore computational costs might approach those of other methods. To give an example; effects of stiffness are often modelled by placing allpass filters at the termination point of a digital waveguide [29], thereby utilising the non-linear phase characteristic of the allpass filter to introduce frequency dependent wave velocities. Although this gives reasonable results for modelling moderate stiffness in strings, it proves problematic for the modelling of bars. The simple waveguide model has been extended to coupled, interleaved waveguide models which do make it possible to include a physically correct formulation of stiffness effects [30, 31]. However, they do not possess the same computational benefits as the simple digital waveguide structure. The efficiency of the simple 1D waveguide also doesn't carry over to higher dimensional problems as this requires a network of digital waveguides connected through junction ports which becomes similar or even worse in terms of computational efficiency when compared to an equivalent finite difference model[3]. Furthermore, the DWM can suffer from numerical dispersion issues[4], which translates to the modes of vibration of the simulation being lower in frequency than those of the true system. This error becomes larger with increasing frequency. For the current research this is only of minor importance however, since my aim is to explore interesting sounds by arbitrarily inter-connecting elementary structures as opposed to matching the sound of a specific instrument as accurately as possible. The modular inter-connection of different distributed objects is not so straightforward to implement unfortunately, since this requires the use of specialised linking elements in the form of the aforementioned junction ports or alternatively which could be based on wave digital filters or hybrid finite difference sections [33].

2.2.2 Finite Difference Method

The finite difference method (FDM) is probably the most direct way of arriving at a discretised solution to a given continuous boundary value problem which may be implemented directly on a computer. The method is based on replacing all the derivatives by finite differences, which can be done in a multitude of ways [34]. The resulting discrete approximation of the continuous PDE is then referred to as a difference scheme and the continuous function $u(x,t)$ is represented instead by the grid function u_l^n. Here l and n are arbitrary integers which are related to continuous time and space according to $t_n = nk$ and $x_l = lh$ respectively, where k denotes the time step (for audio applications this is equal to the sampling period T) and h denotes the spatial step. An approximate solution to the continuous time solution may then be obtained by rearranging terms and solving for u_l^{n+1} (i.e. the value of the grid function at the grid point l at the next time iteration $n+1$) directly when the difference scheme is explicit or by employing numerical solver techniques when the scheme is (semi-)implicit. It is important to keep in mind that in general $u_l^n \neq u(lh, nk)$, or in other words that the value of the grid function at the grid coordinates l, n is only an approximation to the true value of the corresponding

[3]Some exceptions do exist, most noticeably simple 2D structures of a rectangular geometry which can be modelled by 1D waveguides representing plane wave motion in a specific direction [32].

[4]The exception being the case of the ideal string under fixed or free boundary conditions [24].

continuous function. To analyse the performance of a certain scheme, frequency domain analysis methods[5] or analysis based on energetic principles [24] are often employed, which also offer a convenient means of monitoring the numerical stability of a scheme. Proving and guaranteeing numerical stability for a finite difference scheme can become a rather elaborate undertaking however. In comparison, as long as the underlying problem is linear, proving stability for systems modelled with the digital waveguide or modal method is almost trivial, as it is sufficient to ensure that the constituting components (i.e. individual filter structures and delay lines) are passive. The FDM is closely related to the DWM and it can be shown that in fact certain finite difference schemes can be expressed as an equivalent digital waveguide model when choosing the ratio between the time step and space step appropriately [24, 31]. Just as the DWM, the FDM can also suffer from numerical dispersion issues, but as mentioned previously this is not seen as a major concern in the light of this research. The biggest advantage of the FDM is that of all methods it is without a doubt the most generally applicable one and as such the simulation of more complicated model systems is relatively easy[6]. As long as the model system is representable over a grid it can be approximated by an associated finite difference scheme [24]. Inter-connecting different objects is intuitive conceptually, since this is expressed in a purely physical form, i.e. the forces the inter-connected objects impart on each other. Furthermore, due to the direct discretisation of the model equations the programmatic implementation of a scheme is quite straightforward.

2.2.3 Modal Method and Functional Transform Method

Instead of solving a given problem by direct discretisation of the model system as the FDM does, modal-based methods decompose the complex vibration pattern of a vibrating object into a set of basis functions or eigenfunctions which represent all the different modes of vibration of the model system. The shape of these eigenfunctions and their associated eigenvalues, representing the modal frequencies, depend on the boundary conditions and exist in closed form for a select number of simple problems only[7]. Given our PDE of a stiff string described by equation 2.1, the first step to a solution would be to consider the function $u(x, t)$ to be a product of two separate functions; one of which is a

[5]Also referred to as von Neumann analysis [34].

[6]Although it is relatively easy to formulate a scheme for more complicated and perhaps non-linear systems, proving stability inevitably becomes a much more involved matter.

[7]For instance that of the ideal bar or stiff string under simply supported boundary conditions.

function of time only and the other a function of spatial position only. Solving the problem may be accomplished through discretising the PDE using finite element [35] or finite difference [12, 36] methods and employing a diagonalisation process. This then result into a set of uncoupled ordinary difference equations, each of which may be easily updated at every subsequent discrete time interval. The functional transformation method (FTM) instead looks for an analytic solution to the spatial part of the problem using the so-called Sturm-Liouville transformation, which may be seen as a spatial analogue of the Fourier series [21]. The eigenvalues representing the modal frequencies are complex valued in general. However, for real-valued problems they appear in complex conjugate pairs and hence each mode may be considered to consist of such a pair of eigenfunctions and eigenvalues. These pairs may then be combined in order to represent the mode by a second order resonator after discretising the time part of the problem[8]. The location of excitation and picking up the response will influence the contribution of the individual modes to the overall vibration pattern. This is accomplished by projecting the modal data onto the excitation or readout state[9] which results in a weighted set of coefficients. A big advantage of the modal method (MM) over the DWM is that it inherently is very capable of modelling objects with a high stiffness. Ensuring stability is easy as long as the model system is linear and once modal data is computed offline, a real-time implementation is simply implemented as a parallel bank of second order resonators. Another attractive feature for compositional purposes is that a complete spectral description (i.e. all the normal modes of vibration of the model system up to half the sampling rate) of the virtual instrument is present. This spectral description may also be used for reducing the computational costs of the actual simulation part by discarding modes which contribute little to the overall vibration pattern of the model system. For instance, one may consider to discard modes which are above a certain frequency threshold, since the average human ear won't be able to hear these anyway. Similarly, one may decide to discard all modes which fall below a certain amplitude threshold[10] or which decay pattern is short enough so that they do not make a significant contribution to the overall vibration pattern in relation to other modes. Thus, although computational complexity

[8]Employing the impulse-invariant transformation for instance [37, p. 324].

[9]In the simples case this may be represented by a delta function located at a given point along the object.

[10]One needs to be aware that this only makes sense to do when the excitation point and/or readout point are not variable during the simulation as this will change the relative weight of the modes and hence their amplitude.

is approximately the same [24] for any of the mentioned methods in this section[11] in a linear, time invariant setting, the MM is the only method which provides an extra scalability when it comes to reducing computational complexity through the aforementioned mode filtering options prior to the simulation stage. However, the MM also has some drawbacks. For instance, the inter-connection of different objects in a semi-modular fashion is possible, but it does require one to re-calculate all modal data when making any changes to the model system[12]. Furthermore, the modelling of distributed nonlinearities, while possible [38–40], is problematic, as modes are not longer uncoupled and stability issues need to be dealt with in a more rigorous manner.

2.3 Relevant Software Tools

A rather thorough overview of compositional work involving the use of PM sound synthesis techniques appears in [41]. There are a number of important observations that may be extracted from reading this article, which have partly influenced the direction the conducted research has taken. Firstly, a lot of the discussed compositional work is using physically modelled sound material which is predominantly meant to complement or to extend the sonic qualities of acoustical instruments rather than for it to be used in a self-contained way. Secondly, the PM sound synthesis method which seems to be favoured first and foremost for the majority of the works discussed is the DWM. This is probably due to the fact that the DWM is by far the most efficient for modelling simple 1D objects, and hence has been used for the majority of practical sound synthesis applications to date [27]. However as mentioned in section 2.2.1, the efficiency of the DWM method does not carry over to higher dimensional problems and it has structural limitations when it comes to modelling more complex system. Lastly, although the article is discussing a rather extensive body of work, it is concerned with works from composers having a predominantly academic background. It thereby disregards creative applications of PM sound synthesis lying outside of the realm of the academic world, although it has to be said that these applications are generally not thoroughly documented or are in the form of a commercial product[13].

[11]With exception of the special case of the DWM as applied to the ideally vibrating string.

[12]E.g. when adding/deleting objects, changing inter-connection positions or changing boundary conditions.

[13]A plugin effect unit for instance (e.g. a VST or AU).

As delineated in Chapter 1, my personal interest in using PM sound synthesis was fuelled mainly by the promise of being able to produce captivating and idiosyncratic sounds [4, 24, 42, 43] rather than by any specific musical works. Therefore it seems more appropriate to discuss a selection of software tools for generating sounds through PM sound synthesis techniques to see which ones might prove useful in some way or other for the authors personal research aims, rather than to analyse specific musical works in depth. The most appealing feature of the selected tools for the discussion to follow, is that they offer a more general, open-ended framework for the design and implementation of arbitrary complex virtual object configurations, which allows the user to explore a wide variety of different timbres and sound behaviours.

TAO / Cymatic The non-realtime TAO synthesis environment developed by Mark Pearson [4] allows the user to design a custom built virtual instrument by inter-connecting objects like strings, pipes and circular and rectangular sheets. These distributed objects themselves are created out of combining more elementary, lumped building blocks consisting of cells and springs. Individual cells or groups of cells are related to each other through force and damping laws. The environment consists of a synthesis engine and a scripting language. The scripting language allows one to construct instruments and to provide a score as a means to play the instrument. In order to generate sound output, one applies a force to one or more small groups of cells and by writing the varying position of a cell (or possibly several) in response to this excitation to a sound file. Cymatic is the follow-up to TAO [44]. In contrast to TAO, Cymatic is implemented as a real-time sound synthesis environment which offers user-control through the use of gestural, haptic controllers rather than a scripting language.

The greatest promises of both TAO and Cymatic seems to be the relative ease by which one can construct complicated networks of inter-connected virtual objects without little formal or conceptual restriction. This modular approach provides the possibility for experimenting with the design of virtual instruments that have no direct counterpart in the real world and hence, makes possible the exploration of complex object interactions and novel sounding timbres. A disadvantage is the relatively high computational load which depends directly on the complexity of the virtual instrument to be modelled and the lack of means for fine compositional control.

CORDIS-ANIMA / Genesis CORDIS-ANIMA is a physical modelling environment which, like TAO, makes it possible to create complex virtual instruments out of more elementary objects like masses, springs and dampers [2]. Hence, it can be seen as one of the earliest attempts to develop a modular approach to PM sound synthesis using a Mass Spring Network (MSN) framework. In contrast to TAO, the user is allowed to establish connections between individual masses by a number of available connection rules directly. As such, it seems that more emphasis is placed on the behaviour and properties of the object material (i.e. the lumped building block), whereas in the case of TAO, emphasis is on waves propagating through the material [4]. Interesting to note however, is that while CORDIS-ANIMA is using a MSN approach initially, this is ultimately converted to a modal representation [45]. GENESIS is a graphical extension to CORDIS-ANIMA. Its purpose is to offer a more friendly musician-oriented interface to control virtual musical instruments [46].

Probably the greatest advantage is the high modular nature of the software and the complete freedom in which complex structures can be built up from simpler entities. Consequently this offers the possibility to not only build virtual instruments that produce sound output, but also more high-level objects which may be used to control the behaviour of these sound generating objects in a dynamic and potentially meaningful way, musically speaking [47]. The disadvantage of the MSN approach is the fact that difficulties arise when it comes to numerical issues such as accuracy, the implementation of more complex boundary conditions and highly nonlinear systems [24].

Modalys With the Modalys environment from IRCAM it is possible to create virtual instruments by inter-connecting simple physical objects like strings, plates, tubes, membranes, plectra, bows and hammers [3]. In this respect it seems to provide a similar, although a more maturely developed functionality as that offered by TAO and Cymatic. However, the conceptual basis and technical implementation of Modalys is completely different. Instead of a MSN approach to acoustical modelling, Modalys employs PM sound synthesis through mode decomposition of a vibrating physical structure [48]. Furthermore, the Mlys library provides a convenient interface to control Modalys from within the graphical programming environment Max/MSP.

Although less micro-modular[14] than CORDIS-ANIMA, the Modalys environment has its

[14]In the sense that one is not able to specify connection rules or material parameters on the level of individual lumped elements.

own specific advantages. Because every mode is implemented as an uncoupled second-order resonator, it is possible to adjust the damping, frequency and shape parameters of individual modes manually. This can be used to fine-tune a model, but also makes possible a whole range of spectrally motivated sound transformations like, for instance, interpolating between different mode parameter sets to obtain hybrid variants of two or more virtual objects, as has been demonstrated in the work *Eikasia* by composer Hans Tutschku [5].

Modular Percussion Environment As a last example of a more creatively oriented use of PM sound synthesis, it is essential to mention the work of Stefan Bilbao on a modular percussion synthesis environment and prepared piano synthesis. In [1] the author discusses the design and implementation of a network of inter-connected bars and plates in order to simulate a stable and energy conserving virtual percussion environment using the Finite Difference Method (FDM). The word modular in this sense may be a bit misleading, since it is not very easy nor efficient to add or delete individual objects to the system while the simulation is running. The procedure is based on collecting the numerical description of all individual elements into a global block matrix before the start of the simulation, which allows one to update the entire system at every subsequent time step. A recent publication [49] describes an extension of this environment, which allows for the modelling and inter-connection of distributed non-linear components, non-linear object interactions and embedding components in 3D space using a parallel implementation approach with the help of GPU's.

Although such a general and relatively easy manageable approach to building complex systems of interconnected virtual objects sounds very promising from a sound synthesis point of view, the presented sound examples [50] generated with the first incarnation of the modular environment are a little bit disappointing. Most sounds have a rather artificial and dull synthetic character, which is not helped by the sequence of fast, tightly timed impulses used to excite the system. The examples which incorporate plates and nonlinear interaction forces sound much more interesting though. A first untitled demo work composed by Gordon Delap, illustrates the musical potential of the environment [51]. Although some of the sounds in the work are interesting from a spectral point of view, many of the musical gestures and interactions sound a little bit too clean and machine-like. Delap has also composed a more recent multi-channel work entitled *Ashes to Ashes* (2014) [52], using the new functionality of the improved environment. It seems it is one of the first of a series of collaborations between a selection of invited composers and technical researchers affiliated with the NESS [53] project for exploring the musical potential of cutting-edge PM sound synthesis research.

2.4 Other Tools and Software

The tools mentioned in the previous section have directly influenced and inspired the development of my own PM sound synthesis library. In this section some other software packages and applications for sound synthesis through physical modelling will be briefly mentioned to complete the discussion. The majority of the discussed software in this section is restricted in the sense that it does not allow the user to (easily) construct virtual instruments by freely inter-connecting more elementary objects. The exception to this are various more experimental block-based approaches to PM sound synthesis [54, 55], where different subsystems (both exciters as well as resonators) are modelled individually and are inter-connected through specialised linking ports based on wave principles. Although open-source, these packages are rather complex and hence not so easy to set up. Furthermore, unlike the software tools discussed in the previous section it seems that the focus of the developers is largely on technical matters rather than displaying how they could be practically employed for creative sound design and music composition.

There are a number of open-source packages available which provide more fixed, closed-off alternatives to PM sound synthesis in the form of ready-build instruments. Most of these are in some way or other based on the Synthesis ToolKit (STK) [19], which offers a number of different virtual software instruments written in the object-oriented programming language C++. Some of these instruments have been ported to general purpose sound synthesis environments like SuperCollider, Csound, Max/MSP and Pd. However, most instruments are relatively simple and fixed and new models can only be created by implementing them in C++ directly. Hence this does not allow for a very modular and efficient design stage.

Lastly, it is worth mentioning that there are a number of commercial software applications available which employ PM sound synthesis either entirely or at least to some degree. There are a number of VST instruments which model the sound of acoustic instruments like the piano (Pianoteq) and various bowed and plucked string instruments

(Applied Acoustic Systems) as well as ones which seem to be geared towards the production of more unconventional sounds (Kaivo and The Kyma Physical Modeling Toolkit). Although some of these instruments offer high quality sounds, they don't allow for a lot of flexibility due to their rigid design structure.

2.5 Overview of the Developed Software

Based on the discussions from the preceding two sections it was decided that a hybrid FDM/MM approach to PM sound synthesis would be the most desirable option to consider for the development of the software library *PMLib*. The choice for a hybrid approach is based on the fact that in order to adhere to the aims formulated in section 2.1 as closely as possible, it was found that the inter-connection of different resonator objects in a quick and efficient manner was most easily accomplished by adapting the same approach followed in [1]. However, after a first trial version of the software was developed in the MATLAB programming language [56] using an all FDM approach, it was concluded that this didn't allow for a particular efficient workflow, since the nonreal-time nature of the default MATLAB distribution[15] makes it impossible to audition sounds while running a simulation and more extended virtual instrument configurations took a considerable time to compute due to the high computational costs of the FDM. Furthermore, as MATLAB is not an environment geared towards sound synthesis nor music composition, it proved rather cumbersome to use it in a way that stimulated the creative process. Hence, the second and, at this point, most recent version of the software is implemented in the programming languages SuperCollider [57] and Python [58] instead. The choice for SuperCollider is based on the fact that it provides a very powerful and efficient real-time sound synthesis engine and means for designing flexible musical control structures. Python was chosen because of its high-level approach to programming and the fact that it provides an open source alternative to MATLAB for scientific computing through the help of the NumPy and SciPy packages [59]. As both SuperCollider and Python are open-source initiatives, communication and the exchange of data between these two environments is relatively easy and can be accomplished without any extra necessary efforts from the user's side.

[15]There exists a Real-Time Audio Processor library for MATLAB. However, it doesn't seem like this library is actively developed at this stage and it can only be used on the Windows platform.

2.5.1 Alternatives for Implementing PMLib

It is important to point out that there are other alternatives to consider besides Python and SuperCollider. Instead of SuperCollider, any other programming environment or language specifically geared towards sound and music computing (e.g. Csound [60], Pure Data [61] or even the Web Audio API [62]) would be a viable option. The only requirement of such an environment is that it allows for the loading of external files in JSON[16] format and offers support for simulating parallel arrangements of second order filter sections. In fact, it would even be possible to keep the whole process within the Python domain by making use of Python signal processing modules like pyo [64] or scipy.signal [65]. An interesting alternative to consider instead of Python is the Julia programming language [66]. Julia contains an extensive mathematical function library and also offers support for various signal processing tasks.

2.5.2 Summarising the Procedural Flow of PMLib

The reader interested in specific numerical implementation details regarding the developed PM sound synthesis software is hereby referred to the appendices of this thesis. Appendix A contains a detailed mathematical and numerical breakdown of all the implemented numerical algorithms and appendix B contains all code documentation and instructions on how to install *PMLib*. A more detailed demonstration of how to use *PMLib* from a practical point of view will appear in the next section. For the sake of the current discussion it suffices to summarise the workings of the library by decomposing it into the following steps (also see figure 2.1):

1. The complete design specification of the system of inter-connected objects is to be specified by the user in SuperCollider using an object-oriented approach. This involves creating an array holding all 1D and 2D resonator objects the user wants to include in the system, an inter-connection[17] matrix where each column contains exactly two nonzero entrees denoting the relative inter-connection points between

[16] JSON (JavaScript Object Notation) is a lightweight, language independent data-interchange format. As such it provides an easy and convenient way for communicating between different programming languages and environments [63].

[17] Connections between any two arbitrary objects are assumed to be of a simple rigid type, meaning that forces are equal and opposite and that the objects move at the same velocity at their point of inter-connection.

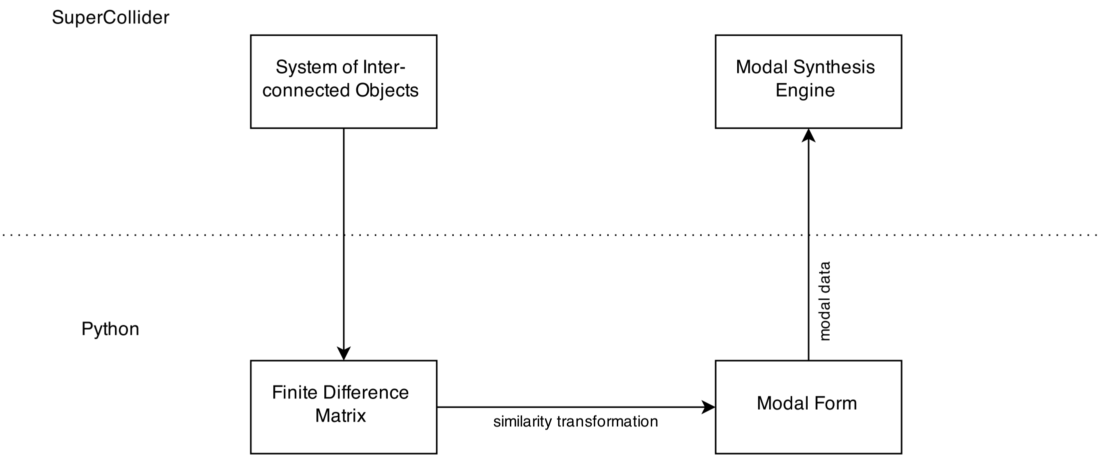

Figure 2.1: A block diagram representing the sequence of steps followed by *PMLib* for calculating and simulating the modal data belonging to a user-defined system of inter-connected objects.

two objects, a mass matrix where each column contains exactly two nonzero entrees denoting the total mass of each of the two objects, an excitation matrix where each column denotes a single excitation condition (giving the possibility to excite multiple objects at once) and a readout matrix where each column denotes a single readout condition (giving the possibility to pick up the summed vibrational response of multiple objects at once). Each row of a matrix thus corresponds to one of the virtual objects making up the system in the order in which they appear in the array.

2. The design specification is parsed into JSON format and saved to disk, after which the JSON file is read back into Python and parsed into a legal Python statement.

3. The Python statement is evaluated and in response the system description will be converted into a single finite difference state transition block matrix. This matrix is transformed into modal form by applying a similarity transformation along similar lines as described in [12, 36]. This will result in a sequence of complex conjugate eigenvalue pairs with associated eigenvectors.

4. Optionally, from the eigenvalues all normal mode frequencies and/or second order filter coefficients may be obtained. Each individual mode can then be simulated with a single second order biquad section. The coefficients of the second order sections may be obtained from all eigenvalues and a projection of the eigenvectors onto a specific combination of input and output vectors. The result will be a

transfer function matrix, where each transfer function corresponds to a specific combination of input and output conditions (i.e. specific excitation and readout location).

5. All modal data will be parsed into JSON and saved to disk. The JSON file is read back into SuperCollider and parsed into valid SuperCollider code.

6. The digital filter coefficients related to a particular transfer function may now be used to drive a parallel bank of second order filter sections, thereby simulating the vibrational response of the system for a specific combination of excitation and readout location.

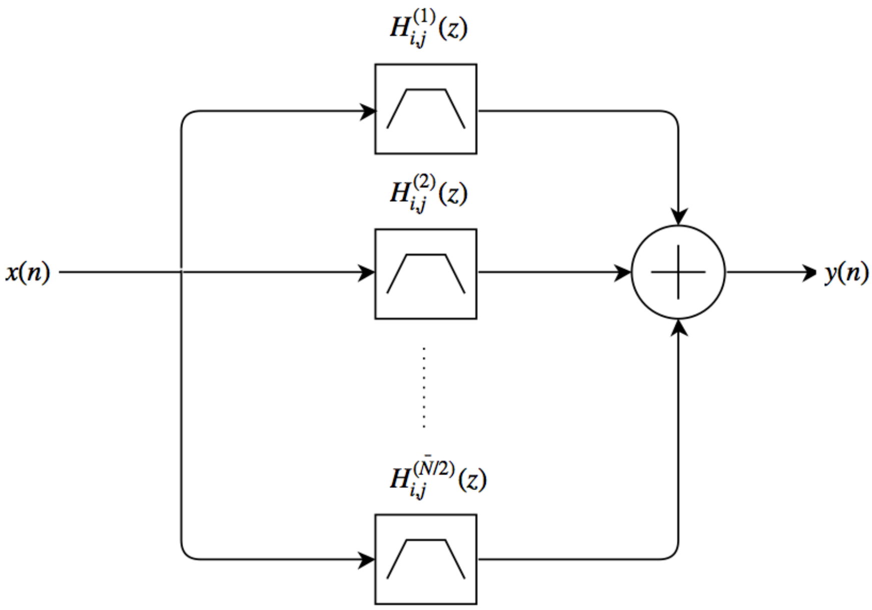

FIGURE 2.2: A block diagram of a resonator model simulating a system of inter-connected virtual objects for a specific combination of excitation and readout conditions.

Note that completion of steps one through five involves an inherently offline process. Hence, from the time the user defines the system of inter-connected objects until the software calculates and returns all modal data, the user is unable to interact with or modify the system. Depending on the amount of modal data to be simulated, the last step can be performed both in real-time as well as in non real-time. Figure 2.2 shows a block diagram representing the general form of the modal synthesis engine described by equation (A.48), simulating the system of inter-connected resonator objects designed by the user for a specific combination of excitation and readout conditions. A resonator is implemented as a parallel set of $\bar{N}/2$ second order sections, each of them modelling a single modal frequency. The subscript i refers to the i^{th} column in the readout point matrix and the subscript j refers to the j^{th} column in the excitation point matrix. The advantage of this hybrid approach is that the design of virtual instruments, sound sculpting and composition may all be accomplished from within the same programmatic environment in an intuitive and efficient way. Furthermore, the MM gives a complete spectral description of the virtual instrument, which has been exploited for various musical and sound processing purposes[18]. Some disadvantages of this approach are that as soon as the internal structure of the virtual instrument is altered, all modal data needs to be re-calculated and that one is restricted to the modelling of linear systems[19]. However, the ability to generate sound output in real-time, even for systems of inter-connected resonators of considerable complexity, and having access to its complete spectral description were the main reasons for favouring this approach over an all FDM one.

2.6 PMLib: A Tutorial

This section is meant to serve as an introduction on how to start creating sounds with *PMLib*. The tutorial is aimed at users who have no former experience working with the library, but have at least some experience working with a high-level, objected oriented programming language. For instructions on how to install *PMLib* and more extensive code documentation, see appendix B. The website accompanying this thesis contains a page which hosts the sound files associated with the code fragments which will appear throughout this tutorial[20]. In order to give the reader an idea of how the library may be used in conjunction with a programming environment other than SuperCollider, the first example will discuss how to set up the model system both in SuperCollider as well as directly from Python.

[18]This will be discussed in more detail in the consecutive chapters, in particular section 4.4.

[19]As mentioned in section 2.2.3, extending the MM to the modelling of nonlinear systems is possible, but this increases computational complexity and implementation efforts become considerably more involved. Hence it was decided to restrict the software to the modelling of linear distributed systems only.

[20]Alternatively, the sound files supporting the code fragments can be found on the USB memory stick at the path **/portfolio/PMLib/audio**.

2.6.1 Example 1: Two Inter-connected Strings

The first system we will consider is that of two stiff, inter-connected strings[21]. In order to keep matters clear and simple at first, we will consider input/output conditions where only one of the two strings can be excited at any time, both strings can be listened to potentially, and the strings are connected at a single point along their length only. We will first go through the steps of defining and calculating the system directly from Python[22]. After this, we will demonstrate how exactly the same thing can be accomplished from SuperCollider and then proceed to show how the system may be simulated in real-time using the computed modal data.

Listing 2.1 presents a fully working Python code example for creating a system consisting of two strings inter-connected at a randomly chosen relative location along their lengths. Saving this as a file with a .py extension and executing it from a terminal window should after a short pause produce a file named 'modalData.json' containing the computed modal data[23].

```
from ResonatorNetwork import ResonatorNetwork
from Resonator1D import Resonator1D
from Resonator2D import Resonator2D
import random

# define two 1D resonator objects with the given gamma, kappa, b1, b2
# and boundaryCond argument parameters
resonators = [Resonator1D(100, .2, .685698, .000515, 'BothClamped'), \
Resonator1D(101, .2, .685698, .000515, 'BothClamped')]

# the relative connection points on the two strings
connPointMatrix = [[random.random()], [random.random()]]

# the masses of the two strings
```

[21] A similar model has been used for generating the sound material for the work *Excite Me* (also see sections 3.4.1.1 and 4.4)

[22] It is assumed that the reader knows how to write and run Python scripts using a modern text editor like Atom [67] and the console window.

[23] The simplest way to assure that the script is able to reference the classes that are imported at the top if the script, is by saving the script in the same directory as the files in which these classes are defined (i.e. at **.../PMLib/python**).

```
15 massMatrix = [[1], [1]]
16
17 # excitation/input condition: excite first string at random position (0 - 1)
18 excPointMatrix = [[random.random()], [0]]
19
20 # readout/output condition: listen to second string at two random positions
21 readoutPointMatrix = [\
22 [0, 0, random.random(), random.random()],\
23 [random.random(), random.random(), 0, 0]\
24 ]
25
26 # create our system of inter-connected objects
27 resonatorNetwork = ResonatorNetwork(resonators, connPointMatrix, \
28 massMatrix, excPointMatrix, readoutPointMatrix)
29
30 # calculate modal data
31 resonatorNetwork.calcModes(20, 14000, .01)
32
33 # calculate the transfer function matrix
34 resonatorNetwork.calcBiquadCoefs(100)
35
36 # save all modal data to disk (only the filter coefficients in this case)
37 resonatorNetwork.saveAsJSON(include='ynnn')
```

Listing 2.1: Script for setting up a system of two inter-connected strings, computing its modal data and saving it to disk as JSON.

The first three lines of the script import the three classes we will use to set up and define our model system. Lines 8 to 9 create all the resonator objects which are going to be the basic building blocks of our system. In this case we create two strings with nearly identical material parameters. Each of the strings has an approximate fundamental frequency of 50 Hz, contains a moderate amount of stiffness, has a relatively low damping and is fixed at both ends according to clamped boundary conditions.

Line 12 defines the inter-connection of the two string objects. Each inner array appearing in the 2d list structure can be interpreted as a single row in a 2 dimensional matrix and corresponds to the connection points (a single point in this specific example) on a single object. Hence, by definition, the number of rows should equal the number of resonator objects which make up the complete system and subsequent rows correspond to subsequent resonator objects defined in the `resonators` array. The same holds for all other 2d list structures appearing in the script. Spatial positions along the length or across the surface of an object are relative, meaning that the domain of choice stretches from 0 - 1 always instead of the actual physical dimension in meters the object occupies in space.

Line 15 is used to define the mass of each object. It should be noted that for the computations it is really the mass ratio M_1/M_2 which is relevant. The mass ratio between two objects may be used to determine how much energy flows from one object to the other in response to exciting it. In our example above, the mass ratio between the two string is set to unity, meaning that the transfer of energy between the two strings would be exactly the same in either direction for identical objects.

Lines 18 and 24 define our input and output conditions. The total number of different input/output combinations is given by the product of the number of columns in each matrix. Since the excitation point matrix contains one column and the readout point matrix contains four columns, there are four input/output combinations to choose from in total. These conditions corresponds to listening to the second string at two randomly chosen relative positions in response to exciting the first string at a single randomly chosen relative position and to listening to the first string at two randomly chosen relative positions in response to exciting the first string.

Lines 27 to 28 collect everything together into a single resonator network object. This object may then be used to calculate the modal data associated with our system configuration. This is accomplished on line 31, where we request to discard any modes that lie outside the range 20 Hz to 14000 Hz or which have a T60 decay time smaller than 0.01 seconds.

For simulation purposes, it is convenient to have a transfer function representation of our system. Hence, we calculate the transfer function matrix associated with our system and boost the gain of the numerator filter coefficients on line 34 of our script. Note that since we defined a total of four different input/output combinations for the example above, this 4×1 transfer function matrix will contain four transfer functions[24].

The last thing we do on line 37 is to save the computed modal data to disk in JSON so that we can possibly use it in any other environment which understands this format. If a specific path is omitted (as in the example above) this file will be named 'modalData.json' and will be saved at the same location as the `ResonatorNetwork` class file. The second argument to the `saveAsJSON` function gives the user some control over the different forms in which the modal data can be saved. This is controllable through setting the individual characters of a four character string to either 'y' (yes) or 'n' (no). The first entree determines if the modal data should be saved in transfer function form, the second if the eigenvalue part of the modal data should be saved in polar form, the third if these eigenvalues should be saved in rectangular form and lastly, the fourth entree determines if the eigenvector part of the modal data should be saved to disk. In light of running a simulation, saving the data in transfer function form is the most convenient, as this will account for all combinations of different input and output conditions automatically.

```
1  (
2  var resonators, connPointMatrix, massMatrix, excPointMatrix,
       readoutPointMatrix;
3
4  resonators = [
5     Resonator1D(100, 0.2, 0.685698, 0.000515, \bothClamped),
6     Resonator1D(101, 0.2, 0.685698, 0.000515, \bothClamped)
7  ];
8
9  connPointMatrix = Array2D.fromArray(2, 1, [
10    1.0.rand,
11    1.0.rand
12 ]);
13
14 massMatrix = Array2D.fromArray(2, 1, [
15    1,
16    1
17 ]);
18
19 excPointMatrix = Array2D.fromArray(2, 1, [
```

[24]Since all individual transfer functions which make up the transfer function matrix will share the same poles, the denominator coefficients are only stored once.

```
    1.0.rand,
    0
]);

readoutPointMatrix = Array2D.fromArray(2, 4, [
    0,          0,          1.0.rand, 1.0.rand,
    1.0.rand, 1.0.rand, 0,          0
]);

~network = ResonatorNetwork(resonator, connPointMatrix, massMatrix,
    excPointMatrix, readoutPointMatrix);
~network.calcModalData(20, 14000, 0.01, 40.dbamp);
)
```

Listing 2.2: SuperCollider code for setting up a system of two inter-connected strings and computing its modal data.

Accomplishing the same thing using the SuperCollider interface looks very similar as may be deduced from listing 2.2 and hence it is somewhat redundant to discuss this in detail. Instead we will move on to discuss how our inter-connected string system may be simulated in real-time. In order to accomplish this, we will need to design an additional synthesis model which will act as the exciter for our model system[25]. Before doing this however, let us first discuss how the resonator simulating our model system may be implemented in SuperCollider using multiple `SOS` UGens, or alternatively, using a single instance of the `SOSBank` UGen[26]. Listings 2.3 and 2.4 present the code for defining a resonator synth using both these UGens with the help of the synth def specification[27].

```
(
// the total number of modes to simulate
var numModes = ~network.modalData["biquadCoefs"]["b1"].size;

// defining a 2-channel resonator synth def using SOS
SynthDef(\resonatorSOS, { arg in, out, preGain = 1.0, postGain = 1.0,
    dryWet = 1.0;
    var input, output, a1L, a2L, a1R, a2R, b1, b2;
```

[25] *PMLib* doesn't provide any ready to use excitation models by default, as it solely focusses on defining and setting up a (possibly complex) resonator object. Hence, the design and implementation of excitation models is completely up to the user of the library.

[26] The `SOSBank` UGen is not included in the standard SuperCollider distribution, but may be obtained from `https://github.com/michaeldzjap/SOSBank`.

[27] For more information on synth defs see the `SynthDef` help file accessible through the SuperCollider IDE.

```

   // allocate space for all filter coefficients
   a1L = \a1L.ir({ 0 } ! numModes);
   a2L = \a2L.ir({ 0 } ! numModes);
   a1R = \a1R.ir({ 0 } ! numModes);
   a2R = \a2R.ir({ 0 } ! numModes);
   b1 = \b1.ir({ 0 } ! numModes);
   b2 = \b2.ir({ 0 } ! numModes);

   input = In.ar(in, 1)*preGain;
   output = [
      Mix(SOS.ar(input, 0, a1L, a2L, b1, b2)),
      Mix(SOS.ar(input, 0, a1R, a2R, b1, b2))
   ]*postGain;
   Out.ar(out, ((((1.0 - dryWet)*input) ! 2) + (dryWet*output)).tanh)
}).add;
)
```

Listing 2.3: SuperCollider code for defining a synth def for simulating a resonator object using SOS.

```
// defining a 2-channel resonator synth def using SOSBank
SynthDef(\resonatorSOSBank, { arg in, out, bufnum_a1, bufnum_a2,
    bufnum_b1, bufnum_b2, preGain = 1.0, postGain = 1.0, dryWet = 1.0;
   var input, output;

   input = In.ar(in, 1)*preGain;
   output = SOSBank.ar(1, 2, input, -1, bufnum_a1, bufnum_a2, bufnum_b1,
    bufnum_b2, postGain);
   Out.ar(out, ((((1.0 - dryWet)*input) ! 2) + (dryWet*output)).tanh)
}).add;

// load all filter coefficients into buffer objects

// excite string 1, listen to string 2
~a1_1 = Buffer.loadCollection(s, [
   ~network.modalData["biquadCoefs"]["a1"][0][0],
   ~network.modalData["biquadCoefs"]["a1"][1][0]
].flat);
~a2_1 = Buffer.loadCollection(s, [
   ~network.modalData["biquadCoefs"]["a2"][0][0],
   ~network.modalData["biquadCoefs"]["a2"][1][0]
```

```
].flat);

// excite string 1, listen to string 1
~a1_2 = Buffer.loadCollection(s, [
    ~network.modalData["biquadCoefs"]["a1"][2][0],
    ~network.modalData["biquadCoefs"]["a1"][3][0]
].flat);
~a2_2 = Buffer.loadCollection(s, [
    ~network.modalData["biquadCoefs"]["a2"][2][0],
    ~network.modalData["biquadCoefs"]["a2"][3][0]
].flat);

~b1 = Buffer.loadCollection(s,
    ~network.modalData["biquadCoefs"]["b1"]
);
~b2 = Buffer.loadCollection(s,
    ~network.modalData["biquadCoefs"]["b2"]
);
)
```

Listing 2.4: SuperCollider code for defining a synth def for simulating a resonator object using SOSBank.

Note that the biggest difference between the `SOS` and the `SOSBank` approach is that for the former we have to pass in the filter coefficients directly to all individual `SOS` UGens, whereas for the latter we use buffer objects to pass the filter coefficients on to the UGen. Furthermore, note that the first two arguments of the `SOSBank` UGen specify the number of inputs and outputs to the UGen. We could have chosen to simulate all our possible input/output conditions (four in total) at once. However, in this case, we have decided to only simulate two of the four output conditions when creating a synth instance by using two different sets of buffers for the numerator coefficients. If one prefers to simulate all possible input/output conditions at once, it would be more efficient to use one set of buffers for the numerator coefficients (and to change the `numOutputs` arguments of the `SOSBank` UGen to four in response), as the recursive part of all second order filter sections is identical for all individual input/output conditions.

Now that we have the code in place for simulating the resonator part, we can define some excitation models. As the implementation of our system of inter-connected resonators will take the form of a parallel arrangements of second order filter sections, any audio input to this filter bank may act as a potential excitation signal. This could be a synthesised signal which is physically inspired in order to model a physical action like a pluck or a strike, but it also may be something more abstract like a sound file or the outcome of some abstract synthesis routine. In order to demonstrate the sonic diversity which may be obtained from a single resonator model, we will use two different synthesised excitation signals that were also used in the work *Excite Me*. The first one will be an impulsive model which may be used to model sounds ranging from water drops to a plucking type sound when used in conjunction with our coupled string model.

2.6.1.1 An Impulsive Excitation Model

Listing 2.5 shows the SuperCollider code for our impulsive excitation model. The model generates two de-correlated signals of filtered impulses that are distributed in time according to an exponential like distribution. Listing 2.6 shows how to generate a synth from the synth def template that produces water drop like sounds (01-drops-aiff) by using empirically chosen values for some of the synth def arguments and in addition suggests a variation that produces a continuous sound with a more stream like quality to it (02-stream.aiff).

```
SynthDef(\filteredImpulses, { arg out, minDelT, maxDelT, minDecT, maxDecT
    , minAmp, maxAmp, freqLo, freqHi, rq, minTrigFreq, maxTrigFreq, decT;
  var output, trig;

  // generate triggers that are (semi) exponentially distributed in time
     using a feedback loop
  trig = LocalIn.ar(2);
  trig = { Impulse.ar(TExpRand.ar(minTrigFreq, maxTrigFreq, trig.sum)) }
     ! 2;
  trig = CombN.ar(trig, 0.2, LFDNoise0.ir(3.14).range(0.07, 0.2).lag(5e3
    ), decT);
  LocalOut.ar(trig);

  // each time a trigger is received, a noise pulse will be generated
  output = Decay.ar(trig, TExpRand.ar(minDecT, maxDecT, trig),
   WhiteNoise.ar(TExpRand.ar(minAmp, maxAmp, trig)));
  2 do: { output = Resonz.ar(output, TExpRand.ar(freqLo, freqHi, trig),
   rq.lag(0.01)) };

```

```
    Out.ar(out, Limiter.ar(output*rq.sqrt.reciprocal, 0.97, 0.01))
}).add;
```

LISTING 2.5: SuperCollider code that defines a synthesis model for generating impulsive sounds.

```
// water drops
x = Synth(\filteredImpulses, [\out, 0, \minDelT, 0.001, \maxDelT, 0.1,
    \minDecT, 0.009, \maxDecT, 0.01, \minAmp, 0.1, \maxAmp, 0.9, \freqLo,
    1800, \freqHi, 19000, \rq, 0.26, \minTrigFreq, 0.1, \maxTrigFreq, 50,
    \decT, 0.2], 1, \addToHead);

// let it rain
x.set(\rq, 0.48, \decT, 12, \maxDelT, 0.1, \freqLo, 400, \freqHi, 12000,
    \minTrigFreq, 5, \maxTrigFreq, 20, \maxDecT, 0.01, \minAmp, 0.2);

x.free; // stop synth
```

LISTING 2.6: SuperCollider code for producing water drop like sounds.

We can now create a resonator synth and route the output of the excitation synth we just created to its input to finally hear how our coupled string system will sound in response to exciting it with the filtered impulses (03-drops_resonator.aiff, 04-stream_resonator.aiff and 05-stream_resonator_variation.aiff). We will first cover how to do this using the synth def which implements the resonator with multiple `SOS` UGens and then show how the same thing might be accomplished using the synth def with the single `SOSBank` UGen. With the excitation synth still running, execute the code within the curly brackets shown in listing 2.7. This is creating a new resonator synth instance and sets the filter coefficients of all second order filter sections. For this example, we are using the first two of our four transfer functions corresponding to listening to the second string at two different locations in response to exciting the first string with only one channel of our two channel excitation signal[28]. The first transfer function is used for the left output channel and the second one for the right output channel.

```
(
// make a resonator synth using the SOS synth def
y = Synth(\resonatorSOS, [
    \in, 10,
```

[28]Note that this is not the only possibility. Another option would be to use both channels of our excitation signal and to map each of them to one of the two transfer functions.

```
    \out, 0,
    \postGain, 24.dbamp,
    // listen to string two
    \a1L, ~network.modalData["biquadCoefs"]["a1"][0][0],
    \a2L, ~network.modalData["biquadCoefs"]["a2"][0][0],
    // listen to string two at another location
    \a1R, ~network.modalData["biquadCoefs"]["a1"][1][0],
    \a2R, ~network.modalData["biquadCoefs"]["a2"][1][0],
    \b1, ~network.modalData["biquadCoefs"]["b1"].neg,
    \b2, ~network.modalData["biquadCoefs"]["b2"].neg
], 1, \addToTail);

// route the excitation synth to the resonator synth
x.set(\out, 10);
)

y.free; // stop synth
```

Listing 2.7: SuperCollider code for making a resonator synth using SOS.

Exactly the same result may be obtained by using the `SOSBank` UGen. Instead of passing in the filter coefficients directly, we pass in the appropriate reference to the relevant buffer object as listing 2.8 demonstrates.

```
(
// make a resonator synth using the SOSBank synth def
y = Synth(\resonatorSOSBank, [
    \in, 12,
    \out, 0,
    \postGain, 24.dbamp,
    // listen to string two at two different locations
    \bufnum_a1, ~a1_1.bufnum,
    \bufnum_a2, ~a2_1.bufnum,
    \bufnum_b1, ~b1.bufnum,
    \bufnum_b2, ~b2.bufnum
], 1, \addToTail);

// route the excitation synth to the resonator synth
x.set(\out, 12);
)

y.free; // stop synth
```

Listing 2.8: SuperCollider code for making a resonator synth using SOSBank.

2.6.1.2 A Continuous Excitation Model

For the second example we will use a continuous, smooth excitation signal to excite our inter-connected string system. However, this time we will use the input/output conditions that correspond to listening to the first string in response to exciting the first string. The synth def code[29] for our excitation signal is shown in listing 2.9.

```
SynthDef(\filteredDrone, { arg out, amp = 1, freqLo, freqHi, freqModT,
    cutoff;
    var source = PinkNoise.ar(1 ! 2);
    18 do: { source = BBandStop.ar(source, LFDNoise1.kr(freqModT).exprange
    (freqLo, freqHi), ExpRand(0.5, 1.5)) };
    Out.ar(out, HPF.ar(LPF.ar(source, 1e4, amp), cutoff))
}).add;
```

Listing 2.9: SuperCollider code that defines a synthesis model for generating continuous, smooth drone sounds.

A synth created from this synth def produces a de-correlated 2-channel noise signal with a dynamically varying spectrum (06-drone.aiff). When used as an excitation for a resonator simulating our inter-connected string system, this produces a harmonic drone with time varying variations along its spectral dimension (07-drone_resonator.aiff). Listing 2.10 shows a code example which demonstrates this. Note that this time we are using the two input/output conditions which correspond to listening to the first string at two randomly chosen locations along its length in response to exciting the first string at another randomly chosen location with again, only one channel of our excitation signal.

```
// make a drone
(
s.makeBundle(nil,
    x = Synth(\filteredDrone, [\out, 10, \amp, 1, \freqLo, 50, \freqHi,
    12000, \freqModT, 3e-2, cutoff, 80], 1, \addToHead);
    y = Synth(\resonatorSOSBank, [
        \in, 10,
```

[29] The synthesis model for this excitation signal was adapted from a SCTweet originally created by Nathaniel Virgo: `https://twitter.com/headcube`.

```
        \out, 0,
        \postGain, 6.dbamp,
        \bufnum_a1, ~a1_2.bufnum,
        \bufnum_a2, ~a2_2.bufnum,
        \bufnum_b1, ~b1.bufnum,
        \bufnum_b2, ~b2.bufnum
    ], 1, \addToTail);
);

[x, y] do: { |synth| synth.free; };      // stop synths
)
```

Listing 2.10: SuperCollider code for generating a harmonic drone using the continuous noise excitation signal and a resonator model.

2.6.2 Example 2: A Multi-String, Multi Plate system

The second example we will discuss involves a more complicated system of inter-connected resonator objects. This system will consist of a 2d resonator (i.e. plate) with five differently tuned stiff strings attached. Listing 2.11 shows the SuperCollider script for setting up this system and calculating its modal data. Note that for this more complicated system, the amount of time necessary to calculate all modal data is considerably larger[30].

```
(
var resonators, connPointMatrix, massMatrix, excPointMatrix,
    readoutPointMatrix;

resonators = [
    Resonator2D(0, 20, 0.612, 0.00112, \allSidesSimplySupported),
    Resonator1D(130.81, 1.2, 0.724, 0.000815, \bothClamped),
    Resonator1D(196, 1, 0.724, 0.000815, \bothClamped),
    Resonator1D(293.66, 0.8, 0.724, 0.000815, \bothClamped),
    Resonator1D(392, 0.6, 0.724, 0.000815, \bothClamped),
    Resonator1D(523.25, 0.5, 0.724, 0.000815, \bothClamped),
];

connPointMatrix = Array2D.fromArray(6, 5, [
```

[30] To give an indication: on an iMac with a 2.7 GHz Intel Core i5 processor this takes approximately 35 seconds.

```
    {1.0.rand}!2, {1.0.rand}!2, {1.0.rand}!2, {1.0.rand}!2, {1.0.rand}!2,
    1.0.rand,     0,            0,            0,            0,
    0,            1.0.rand,     0,            0,            0,
    0,            0,            1.0.rand,     0,            0,
    0,            0,            0,            1.0.rand,     0,
    0,            0,            0,            0,            1.0.rand
]);

massMatrix = Array2D.fromArray(6, 5, [
    3, 3, 3, 3, 3
    1, 0, 0, 0, 0,
    0, 1, 0, 0, 0,
    0, 0, 1, 0, 0,
    0, 0, 0, 1, 0,
    0, 0, 0, 0, 1
]);

excPointMatrix = Array2D.fromArray(6, 2, [
    0,        {1.0.rand}!2,
    1.0.rand, 0,
    1.0.rand, 0,
    1.0.rand, 0,
    1.0.rand, 0,
    1.0.rand, 0
]);

readoutPointMatrix = Array2D.fromArray(6, 4, [
    {1.0.rand}!2, {1.0.rand}!2, 0,        0,
    0,            0,            1.0.rand, 1.0.rand,
    0,            0,            1.0.rand, 1.0.rand,
    0,            0,            1.0.rand, 1.0.rand,
    0,            0,            1.0.rand, 1.0.rand,
    0,            0,            1.0.rand, 1.0.rand
]);

~network = ResonatorNetwork(resonator, connPointMatrix, massMatrix,
    excPointMatrix, readoutPointMatrix);
~network.calcModalData(20, 14000, 0.01, 40.dbamp);
)
```

LISTING 2.11: SuperCollider code for setting up a plate-string system and computing its modal data.

Lines 4 to 11 of the script will, as before, define an array holding the resonator objects which will make up our system. In addition to the five 1D resonator objects, we now also introduce a 2D resonator in the form of a plate. Our connection matrix looks more complicated this time, as we have defined five different connections between each string and the plate at randomly chosen relative locations. For the plate we need two relative coordinates to describe these locations as opposed to just one for the strings. These two coordinates are simply packed into a two item array. Furthermore, note that the dimensions of the 2d array reflect the number of resonator objects (the number of rows) and the number of connections between any two objects (the number of columns). Hence, each column should contain exactly two non-zero entrees[31]. The same holds for the mass matrix. In this case we define the mass of the plate to be three times that of every string, but we could have chosen unique values for each individual connection.

Since there are two different excitation conditions and four readout conditions, there will be a total of eight different input/output combinations to choose from, or in other words, our transfer function matrix will be 4×2. To give some examples of the possibilities: we could listen to the plate in response to exciting all five strings at the same time (08-drone_plate.aiff) or conversely, we could listen to the summed output of all five strings in response to exciting the plate (09-drone_strings.aiff). Both these combinations sound very similar though when compared with the condition which corresponds to listening to the strings in response to exciting the strings (10-drone_strings_2.aiff). Listing 2.12 shows the SuperCollider code for realising these three different input/output conditions using the continuous excitation model we designed earlier in conjunction with the resonator synth def using the `SOS` UGens.

```
// listen to the plate, excite all five strings
(
// create the excitation
x = Synth(\filteredDrone, [\out, 10, \amp, 1, \freqLo, 50, \freqHi,
    16000, \freqModT, 3e-2, cutoff, 20], 1, \addToHead);

y = Synth(\resonatorSOS, [
   \in, 10,
   \out, 0,
   \postGain, 24.dbamp,
   \a1L, ~network.modalData["biquadCoefs"]["a1"][0][0],
```

[31] Establishing a connection at relative position 0 should be defined as a very small, but non-zero number (e.g. 0.0001).

```
    \a2L, ~network.modalData["biquadCoefs"]["a2"][0][0],
    \a1R, ~network.modalData["biquadCoefs"]["a1"][1][0],
    \a2R, ~network.modalData["biquadCoefs"]["a2"][1][0],
    \b1, ~network.modalData["biquadCoefs"]["b1"].neg,
    \b2, ~network.modalData["biquadCoefs"]["b2"].neg
], 1, \addToTail);
)

// listen to the 5 strings, excite the plate
(
y.free;
y = Synth(\resonatorSOS,[
     \in, 10,
    \out, 0,
    \postGain, 24.dbamp,
    \a1L, ~network.modalData["biquadCoefs"]["a1"][2][1],
    \a2L, ~network.modalData["biquadCoefs"]["a2"][2][1],
    \a1R, ~network.modalData["biquadCoefs"]["a1"][3][1],
    \a2R, ~network.modalData["biquadCoefs"]["a2"][3][1],
    \b1, ~network.modalData["biquadCoefs"]["b1"].neg,
    \b2, ~network.modalData["biquadCoefs"]["b2"].neg
], 1, \addToTail);
)

// listen to the 5 strings, excite the 5 strings
(
y.free;
y = Synth(\resonatorSOS,[
     \in, 10,
    \out, 0,
    \postGain, 24.dbamp,
    \a1L, ~network.modalData["biquadCoefs"]["a1"][2][0],
    \a2L, ~network.modalData["biquadCoefs"]["a2"][2][0],
    \a1R, ~network.modalData["biquadCoefs"]["a1"][3][0],
    \a2R, ~network.modalData["biquadCoefs"]["a2"][3][0],
    \b1, ~network.modalData["biquadCoefs"]["b1"].neg,
    \b2, ~network.modalData["biquadCoefs"]["b2"].neg
], 1, \addToTail);
)
)
```

```
[x, y] do: { |synth| synth.free };        // stop synths
```

LISTING 2.12: SuperCollider code for making different plate-string resonator synths.

2.6.3 Concluding Remarks

In summary, this section has served to demonstrate how one may setup, compute and simulate a system of inter-connected resonator objects using the *PMLib* library. In addition, some examples were given of how to excite these systems using two sonically different excitation signals. It should be noted that although in theory *PMLib* offers an almost unlimited number of possibilities for designing different systems of inter-connected resonator objects of various levels of complexity, the sonic diversity one is able to obtain is to a large degree dependent on how the system is being played, i.e. excited. Hence, and although not an integral part of *PMLib* at this point, the design and experimentation of suitable excitation models is vital in exploring the true sonic potential of the library and this could pose a challenge, although an interesting and hopefully creatively stimulating one, for anyone interested in experimenting with the library.

Chapter 3

Algorithmic Composition as a Tool for Artistic Expression

The purpose of this chapter is to cultivate a discussion about creative issues surrounding algorithmic and automated composition practices. It has to be made clear from the outset that a deep investigation of the field of algorithmic composition is beyond the scope of this thesis and is largely irrelevant. None of the portfolio works make explicit use of algorithmic procedures for determining the final form of the musical structure. However, automated processes have been used generously for generating individual sounds, and serve as an aid for traversing the sound space of a sound synthesis model in order to get a sense of the sonic possibilities. In addition, several physically inspired algorithms[1] have been designed to imitate a number of physical interaction behaviours and natural sounds scenes. By doing so, I strongly believe that I have been able to generate sound material which would have been (near to) impossible to produce if I had restricted myself to a manual approach exclusively. Hence, I think it is worthwhile to clarify when and how I have made use of automated processes for sound generation. The chapter will commence with a global overview of the field of algorithmic composition and touch upon a number of philosophical issues surrounding musical authorship and the treatment of obtained materials. Note that in order to focus the discussion, it will be restricted to the field of electroacoustic music primarily, thereby ignoring the use of algorithmic procedures in the confines of instrumental music and the possible esoteric questions it might raise

[1]Based primarily upon empirical analysis and re-synthesis of the timbral and micro-structural characteristics of real-world sounds rather than applying strict physical reasoning.

with regard to this field. Furthermore, I think it is important to note that this chapter is primarily concerned with how certain compositional strategies have been employed as opposed to why. A thorough discussion of the latter will be pursued in the following two chapters, hence providing a more artistically and aesthetically motived answer to why I have chosen for some of the strategies illustrated in section 3.4.

3.1 Creation versus Imitation

It is generally stated that algorithmic composition procedures may serve two different purposes; either that of style imitation or as a method for the creation of genuine musical expression [68]. In the case of the former, aesthetic considerations can be dealt with in a largely objective, analytic way, as the goal of the algorithm is to adhere as closely as possible to certain clearly defined stylistic criteria. The quality of an algorithm thus can be assessed based on how accurately the result obtained from the algorithm is meeting those criteria. However, in the latter case, matters of aesthetics are much more open to subjective interpretation and hence the 'success' of an algorithm is largely if not completely ambiguous. Furthermore, often a clear delineation between imitation and creation is obscured by the fact that *"...one must bear in mind that 'genuine methods of composition' cannot be defined precisely, since there may also exist style imitations of a proprietary style"* and *"...the integration of common algorithmic procedures of musical structure generation... ...may also be seen as style imitation on a structural level"* [68, p. 259]. In the light of this research the imitation of common algorithmic composition procedures or musical styles is for all intents and purposes extraneous. However, analogous to the imitation of a certain musical style, the simulation[2] of a specific physical interaction (e.g. bowing, rolling, scraping, bouncing), or natural sound scene (e.g. water waves crashing on a beach, the sound of rain drops falling on a surface) by means of some sound synthesis model may be interpreted as an algorithm for 'imitating' certain sonic features in order to arrive at a particular sound behaviour. Although one may argue that these two examples of imitation belong to two very different contexts, I would like to assert that for my objectives they serve exactly the same goal; namely the modelling of certain stylistic, respectively timbral and micro-structural features for the purpose

[2]For the sake of this discussion, a simulation implies both those which are based wholly on physical reasoning and those which are physically inspired, but are not necessarily derived from any governing physical laws.

of musical invention. It is important to stress though, that my main motivation for simulating certain types of sounds is for creative practicalities rather than imitation for the sake of demonstration only. Hence, rather than purely focussing on the imitative qualities of an algorithm, I am more interested in how the algorithms that make these imitative qualities possible, can be exploited as tools for genuine creative use and composition. This is something I will expand on in section 3.4 of this chapter, after I have explained more clearly how I generally view my musical responsibilities regarding the result of algorithmic composition procedures.

3.2 Algorithmic Autonomy: from Sound Generation to Musical Structuring

According to [68], in the most meticulous sense, an algorithmically composed work implies that its musical structuring is determined completely through algorithmic means, although it does not necessarily mean that all musical parameters are to be algorithmically determined as well. A more liberal interpretation of the term suggests that it may be used to signify methods which serve as tools to aid in the realisation of certain aspects of a composition from the micro level up to the macro level. This may vary from a procedure that generates musical material in the form of individual sounds, to routines which establish certain prescribed relationships between multiple sounds, to methods that deal with the organisation of materials into larger-scale musical sections. Irrespective of the level on which an algorithm operates, it is the composer who has to make a decision on how she positions herself in relation to the outcome of some algorithmic process. The most extreme and rigid stance would be to insist that the outcome of the algorithm is indisputable and rigid, regardless if it is living up to the aesthetic expectancies of the composer. A more compromising take would be to not adhere strictly to the outcome of an algorithm, but to see if, by modifying certain features, a more gratifying result may be obtained. Depending on the type of algorithm and the material to which it is applied, this may involve a more informed tuning of the original algorithm parameters, applying the same algorithm to different source materials, or to subject the result of the algorithm to additional editing or processing operations. In this regard Nierhaus mentions that: *"Algorithmic models, regardless of their use for the formal structuring or the generation* *of new material, may also only represent compositional possibilities...”* [68, p. 260]. After several years of listening to algorithmically composed works and experimenting with algorithmic composition methods on all different levels, I have come to appreciate this alternative representation of the term increasingly. My (prevalent) conclusion is that the majority of works that still allow for creative interference by the composer tend to sound more musically coherent than works that adhere strictly to the outcome of some algorithm.

To put this into context, one may consider the work *Funktion Rot* (1968) [69] by composer Gottfried Michael Koenig as an example of an algorithmically composed work with a fixed structure determined completely by the computer: *“The formal assembly of the final version was calculated by the computer, which determined for each circuit diagram the variants, the duration of the sound and its position (in time) in the piece”* [70, p. 13]. Although one has to acknowledge the period in time at which this work has been realised, I think it is safe to say that even a reasonably well-trained listener will find the near twenty two minute duration work a rather challenging listening experience. Apart from the harsh timbral characteristics of the presented sound world, it seems that in general a truly convincing musical reasoning for the flow of events is absent. Hence, one may question if the structural form of the final work would have turned out differently if some form of human intervention had occurred. In contrast, the electronic version of *Kontakte* (1958–1960) [71] by composer Karlheinz Stockhausen is an example of a work from the same time period which manages to present a much more natural flow of events, while still making use of structuring approaches which incorporate formalistic principles (although perhaps not to the same stringent extent). Although *Kontakte* can not be categorised as an algorithmic work in the strictest sense of the word, it is nonetheless interesting to compare it to *Funktion Rot*. Firstly, because it has been realised with the assistance of Koenig and secondly, because Stockhausen’s aspiration to unify all aspects of sound under a single compositional procedure suggests a certain formalistic approach to deriving larger-scale musical form from the properties of sound itself. In this regard Stockhausen states that: *“The temporal process by which a sound is transformed into a rhythm can, without a doubt, take a musical form. The aesthetic judgement of the listener can determine if the result of this process is successful, if it is significantly congruent with the total work, and if it has been accomplished with originality and imagination”* [72, p. 47]. The second part of the quote is particularly illuminating, as it hints at the acknowledgement of aesthetic considerations when it comes to judging the outcome of some formalistic structuring procedure.

This example should hopefully help to illustrate the inherent difficulty in ascribing musical meaning to the outcome of algorithms for musical structuring processes. Although I certainly do not want to argue that they are completely incapable of producing musical useful results, I am convinced that a more musically rewarding outcome will ensue if the result is treated as a creative possibility rather than a definitive, immutable product. To a large extent, the same issues are transferable to algorithms which are concerned with the shaping and articulation of individual sounds and events. Rather than taking for granted all the sounds and events derived from some algorithmic or automated process, it makes more sense to me to treat them as possibilities from which one is free to choose only those options that are the most promising from a sonic and/or musical perspective. This is the approach I have taken in most of my previous compositional works which rely on algorithmic procedures and it is for this reason that I can assert with confidence that, in general, they present themselves as genuine musical works as opposed to just an arbitrary sequence of sounds without a traceable musical thread running through them.

3.3 Musical Authorship and Reassessing the Role of the Composer

The use of algorithmic composition processes for creative purposes brings up questions about authorship and authenticity in relation to musical invention. One might say that by adhering rigidly to the outcome of an algorithm, the composer is avoiding the need to take any artistic responsibility by postulating that the music 'is what it is' as a result of the algorithm used. In addition, a claim may be made that the composer cannot really take credit for the musical result if she solely is the user rather than the inventor or designer of the algorithm [73]. However, genuine creativity and artistic expression often is born out of personal motivations which go beyond a clearly definable framework of rules and instructions[3]. Hence, doubts about authorship and the authenticity of the creative result should not necessarily pose a problem if one succeeds in fulfilling a certain creative objective that can be appreciated mostly, if not exclusively, for its musical merit

[3]Although one may always try to device algorithms which reflect embodied artistic methodologies as closely as possible of course.

rather than to merely serve as a sonic demonstration of the used algorithm. In this regard it is not much different from identical philosophical issues which tend to surface in discussions about sampling culture [74]. I am of the opinion that just as a work made up from sampled materials can present itself as original and genuine, so can a work which makes extensive use of (common) algorithmic composition methods if they have been applied with enough thought, skill and dedication in order to transcends a purely formalistic or algorithmic mode of listening. In this respect it is analogous with what Denis Smalley terms as technological listening: *"Technological listening occurs when a listener 'perceives' the technology or technique behind the music rather than the music itself, perhaps to such an extent that true musical meaning is blocked"* [75, p. 109].

By examining the musical potential of the outcome of some algorithmic process rather than to take it for granted, the issue of the artistic responsibility of the composer is resolved, as one cannot hide behind ascribing any musical imperfections to the algorithm any longer. However, a shift in the role of the composer is inevitable. This shift is likely to be different dependent on the degree of autonomy given to the algorithm and the level it will operate on. Instead of sketching out all possible roles (something which could encompass a whole thesis on its own), I will restrict the discussion to how I position myself regarding this. Let me start by saying that rather than to explicitly plan out all details related to the musical process from start to finish, my role as composer is characterised principally by engaging in the design and experimentation with algorithms for specific musical tasks. These tasks can range from the generation and shaping of individual sounds to the formation of longer duration musical gestures, but rarely do I employ algorithmic methods for fixing the overall musical structure, as I feel it is my artistic responsibility to direct this according to my own preferred musical preferences and aesthetic standards.

For me, algorithmic processes prove to be helpful especially for exploring the sonic possibilities of sound synthesis models by devising procedures that are used to automate model parameters dynamically over time. This allows me to generate a large number of sounds in a (semi)-automated way from which later I can select those which I consider to have the most interesting sonic characteristics. By documenting the evolution of the specific synthesis parameters giving birth to these sounds as well, I can go back later and make adjustments if I feel the generated sounds are lacking in certain respects, or

I might decide to use these settings as the basis from which to spawn a next generation of sounds. The exact form the automation of synthesis model parameters takes is dependent on a number of different things, but essentially reduces to the question if I desire a specific parameter value to be static or variable over the duration of the sound or event. In either case, the parameter needs to be assigned some initial value. This initial value and the way it will evolve over time (if not kept static for the duration of the sound/event) can be specified directly, or conversely can be set completely at random[4]. In between these two extremes there exists a whole range of possibilities that stretch from deriving the necessary information from algorithmic procedures which have no direct musical connotation, but nonetheless could potentially be interesting for some of their characteristic qualities[5] or from more musically and/or sonically informed procedures or rule sets[6].

3.4 The Appearance of Algorithmic Procedures within the Portfolio

Throughout the portfolio one can find numerous examples of synthesised imitation sounds, which range from natural sounds reminiscent of flowing water and rain showers, as can be heard in the works *Excite Me* and *Extase 2, 3 & 4*, to that of waves crashing on a beach in the opening section of *Extase 2, 3 & 4*, and that of physical contact sounds like bouncing, rolling, scraping and rubbing in *Stable Equilibrium* and the concluding section of *Excite Me*. The source of these sounds can be traced back to a selection of customary designed sound synthesis models of both the abstract, physically inspired, and physically modelled kind.

As explained at the end of chapter 2, the physical modelling methodology employed for the developed software library *PMLib* is based upon a hybrid finite difference/modal approach. Effectively, this means that all synthesis which can be classified as pure physical modelling is in the form of a linear time independent filtering operation on an

[4]Taking into account any sensible lower and upper boundary values of course. Sending a signal with an absolute amplitude value larger than one directly to the output of one's speakers is a bad idea in general.

[5]They might for instance model some natural or chaotic system, thereby imposing some form of regularity or naturalness without resulting into something too predictable.

[6]These might take into account certain desired harmonic features, psychoacoustic effects or include timbre specific considerations.

input signal. The physically correct way of interpreting this input signal is by identifying it as a time dependent force signal, whereas the output of the filter operation is to be interpreted as the vibratory response at a point along the surface of a system of interconnected 1D and 2D physical objects in response to applying the force signal to it at another point along its surface. In order to make a more clear functional difference between the two, the input signal is also commonly referred to as the excitation signal and the linear filtering operation as the resonator. These two different naming conventions will be used interchangeably throughout the remainder of this thesis, unless there is a specific need to adhere to one or the other.

In a more general context, the input signal can be any arbitrary digital signal. Although a complete physical justification might be lost in this case, there is nothing from preventing one to experiment with signals that have been generated through other (abstract) sound synthesis methods or even with recorded or sampled sound materials. In fact, in order to obtain a sufficient degree of timbral and structural variety from a given system of interconnected objects represented by the filtering operation, it was found to be necessary to let go of the strict interpretation of the input signal as a time varying force signal and instead utilise the possibility to use any digital signal as an input to the filter. In some instances this excitation is loosely based upon physical principles, as is the case with the various impact and contact type sounds appearing throughout the portfolio. In other instances the excitation signals are completely abstract and are primarily used for generating drones with dynamically changing spectral qualities when used as the input to the resonator. In some instances the same excitation models, in conjunction with a resonator model, are used to generate both physically plausible[7] sounds as well as completely abstract sounds and various morphing strategies have been devised to transform between these two extremes[8]. For the remainder of this chapter, I would like to focus on a purely qualitative discussion by giving some examples[9] of designed excitation and resonator models, and describe the way they are controlled in order to produce certain types of sounds and sound behaviours.

[7]As mentioned before in section 1.1, a more precise definition of the term 'physically plausible' as it applies to sound in the context of this research will be kept for section 4.3. For now it will suffice to interpret this as a sound which the listener is likely to link to a physical source or cause.

[8]A conceptual oriented discussion on the use of these types of sound transformations will be kept for section 4.5.

[9]I think it will suffice to only refer to a selection of the used sound synthesis models as otherwise the discussion may become overly complex and lengthy. However, I do think it is important to at least examine some of them as it will illustrate more clearly my way of working from a practical point of view and how algorithmic and automated processes play a part in this.

3.4.1 Some Examples of Sound Models and Control Strategies

As mentioned at the end of section 2.5, a drawback of the particular way the PM sound synthesis library has been implemented is that the parameters of the resonator are fixed during the simulation stage, since changing any material parameters or boundary conditions would require a re-calculation of all modal data. Some alternative strategies have been explored in order to let, or at least let it appear as if, resonator parameters change dynamically during the simulation. However, the majority of different sounds, obtained from using the same resonator model, have come about by changing the parameters of the excitation models exclusively. In general, the designed excitation models have evolved by starting from simple noise and/or impulse based models and adding desired functionality empirically along the way, in order to extent their timbral and structural qualities. Hence, most excitation models have a rather complex design and need the tuning of a considerable amount of parameters. The advantage of this, is that it offers a great deal of control for shaping the sound. However, it is not always apparent how these parameters can be set in a very organised, clear-cut way, as some of them are more intuitively interpretable than others and the number of possible parameter combinations can become rather daunting. Hence, in order to circumvent the laborious task of having to manually set dozens of parameters time and time again in order to explore the sonic space of a particular excitation model itself and in conjunction with a certain resonator model, various automated approaches have been devised. Some of these automated approaches have been directly incorporated into the compositional process[10], whereas others have served as tools to generate sequences of individual sound variants from which a selection has been made manually afterwards.

3.4.1.1 Example of an Automated Trigger System

Figure 3.1 shows a somewhat simplified block diagram of an excitation model that has been used to model sequences of short, filtered noise bursts dispersed in time according to an exponential-like distribution. Effectively, this makes the chance more probable that individual noise bursts group together into clusters, where these clusters themselves are perceived in time through the gaps of silence in between them. Empirically, this clustering effect was found to sound more natural when trying to imitate impact-like

[10] E.g. the automatic triggering of short duration events like plucks, strikes and impacts.

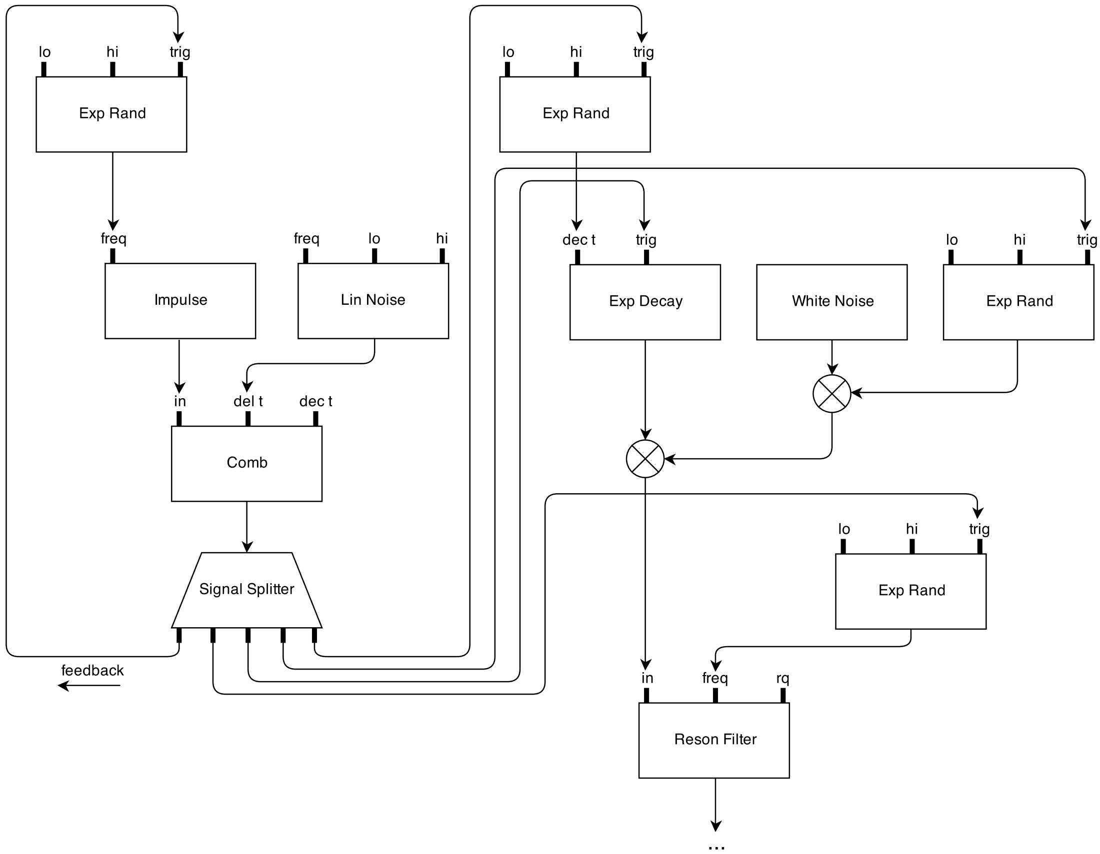

FIGURE 3.1: Simplified block diagram for a synthesis model producing filtered noise bursts in response to receiving a trigger.

events like water drops or small stones falling on a surface. Depending on the parameter settings, the types of sounds one might obtain from this model range from individual water drops to rain showers and quasi-regular, dense, grainy textures. When used as an excitation for a resonator module, it may serve as a simplified mechanism for plucking a string or striking a plate. This model has been used extensively in the works *Excite Me* and *Extase 2, 3 & 4* in order to generate just these types of sounds.

The opening section of *Excite Me* starts with a sound scene that is reminiscent of a rain shower, created by a gentle, sustained noise layer in the background and what sounds like the incidental water drop created by the noise burst model in the foreground. The water-like quality to the sound is the result of both spectrally related as well as temporally related features. At the basis of a single water drop is a noise source that is being multiplied with an exponentially decaying envelope (denoted by the 'Exp Decay' block in figure 3.1) in order to create a short noise burst. The decay time is randomly determined to lie between user specified 'hi' and 'lo' values for every new noise burst. Setting a suitable range for the 'hi' and 'lo' parameters was found to be critically important as decay times that were too long would make the result too harsh and rough, and too short decay times would tend to sound too glitchy and artificial, thereby in both cases breaking the illusion of listening to a water drop. Additionally, the 'wetness' of the final sound can be ascribed mainly to the effect of the resonant filter. Again, a careful tuning of the filter parameters proved critical in order to synthesis a convincing water drop sound. Relatively low (but not too low) values for the 'rq' (reciprocal of Q) or bandwidth parameter in combination with random centre frequencies within the mid-high to high frequency range were found to produce the closest sound possible to that of a real water drop.

Sticking with the specific example of how the noise burst model has been used in the opening section of *Excite Me*, the water drop sounds are slowly transformed into a stream of semi-regular grains that starts to take on a more substantial form around 2:00. This transformation is made possible through the use of several envelope generators with customary designed curves in order to control the 'lo' and 'hi' inputs for the top left 'Exp Rand' module[11] in figure 3.1 which in turn controls the frequency of an impulse generator. Similarly, envelopes are used to supply values to the 'dec t' input dynamically in order to control the decay time of the comb filter module in time, thereby supplying an additional control for increasing the density of noise bursts dynamically over time. At around 2:30 one starts to perceive a more harmonic element dominating the sound of the grain texture, which is the result of incrementally routing the texture through a resonator module simulating a system of two inter-connected strings. The resonator is used as a kind of send effect here, taking at its input an excitation signal and providing at its output the option to create a signal which is a mix between the unaffected input signal and the input signal after it has been processed fully by the resonator module. The grain texture persists until approximately 5:30, after which it has slowly faded away into a harmonically rich drone. The description of this particular sound event serves as a typical example of an element appearing in my work that has been generated with the help of algorithmic means, but at the same time is informed heavily by timbrally based decision making. Additionally, it demonstrates how a slowly evolving, long duration event, acts as one of several individually composed sound layers that together form a unified, homogeneous whole from a perceptual point of view.

[11] Incidentally, this approach of using two envelopes to control the lower and upper boundaries for (semi-)random selection purposes is what Koenig refers to as a tendency mask [70].

3.4.1.2 Example of an Automated Approach for Generating Sound Variants

In the previous section I have described an algorithmic approach for generating impulsive and sustained, structural sounds, used to simulate natural sounding settings (e.g. water drops and rain showers). Furthermore, it also has served as a plucking-type excitation signal for a resonator model corresponding to a system of two inter-connected strings. Hence, it is meant to serve as an example of an automated approach which was inherently part of the compositional process. In this section I want to present an example of an automated approach which, instead, was only indirectly part of the compositional process. It has been used as a generator of individual sounds exclusively and larger scale musical structures were obtained by manually combining and editing these results. To this end, a sound model used to generate all the sound material for the work *Extase 1* will be described. A simplified block diagram of this impulse-based model is represented by figure 3.2. The range of different sounds the model has to offer on its own is rather limited. When used as an excitation signal for a resonator however, the timbral diversity that may be obtained from it is much broader.

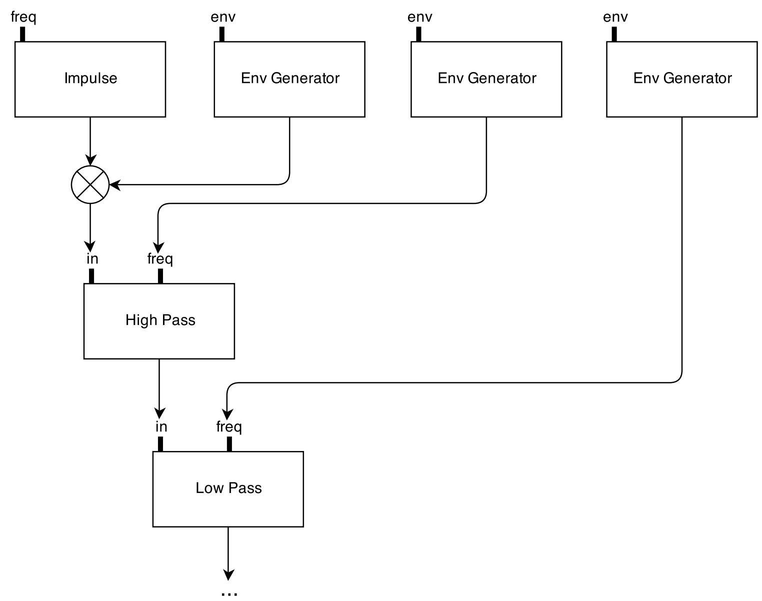

FIGURE 3.2: Simplified block diagram for a synthesis model producing a filtered impulse train.

The initial sound is produced by multiplying the output of an impulse generator with a user-specified amplitude envelope. Depending on the shape and total duration of this envelope, the impulse model may be used to generate impact-like to bowed-like sounds when used as an excitation for a suitably designed resonator model. For *Extase 1* the resonator model corresponds to one of a system of four inter-connected plates. The frequency argument of the impulse generator denotes the rate at which individual impulses follow each other up in time and, hence, it will produce a pitched tone when set high enough. For *Extase 1* it was decided to set the frequency argument equal to one of the first twelve modal frequencies of the plate[12], thereby utilising the modal data associated with the four plates as a musical scale[13]. The final excitation signal actually consists of two copies of the filtered impulse model. Each copy has almost identical parameters settings, except for the frequency argument of the impulse generator, which for one of the copies is set to half the value of the other one. Additional spectral shaping is provided by the high/low pass filter combination at the end of the signal chain and can be made to vary dynamically over time by setting the cutoff frequency of the filters through a user-supplied envelope shape. For the majority of the work, the filtered impulse model is used to produce sustained sounds with a total duration varying approximately between 3 and 30 seconds. The closing section, starting around 5:30, is dominated by impulsive sounds, each lasting for about 10 to 20 milliseconds. These are used to excite the plates for a short amount of time in order to let the sound evolve according to their natural decay pattern. The impulsive sounds make use of exactly the same filtered impulse model and envelope shapes, but now the total duration of the shapes are scaled according to the much shorter excitation duration.

The final harmonic and temporal structuring of *Extase 1* has been accomplished by layering individual sounds to form longer duration ‘chordal’ events that weave in and out of each other throughout the piece. Shaping each individual sound by hand would have been a very laborious and time consuming undertaking, as this would have required the repeated manual setting of break points, segment times and segment curvatures for all three envelope shapes. Since every ‘chord’ consists of up to three or more individual sounds, this quickly would have become an unwieldy and tedious process. Hence, the idea occurred to me that the task of generating potentially interesting sounds would work very well as a carefully designed automated process. Ultimately, this relieved me from having to do any of the ‘hard creative work’ [73] and instead allowed me to focus on shaping the larger-scale musical form to my exact liking by selecting the most promising

[12]Initial inspiration for doing this was taken from the section on simulating micro collisions appearing in [76].

[13]More on this will be said in section 4.4 when I will discuss in more detail the spectrally informed composition strategies which have been employed throughout the portfolio.

sounds and combining them together into larger structures. The automation process is a rather simple loop-based procedure. At each iteration one of the first twelve modal frequencies of one of the four plates is selected, three new envelopes shapes with some form of similarity are created, and, in response, the sound is synthesised in real-time and finally recorded to disk. The automation of the envelope parameters has resulted in a number of unique sounds, not likely to have been produced by a manual approach, as some of the resulting envelope shapes are not that obvious in relation to each other. However, as it turned out, this seemingly odd combination often (although not always) resulted in sounds with a certain uniqueness and edge to them.

3.4.1.3 Alternatives for Changing Resonator Parameters Dynamically

As mentioned before, the bulk of the resonator model parameters are fixed in time due to the used methodology. However, a number of alternative approaches have been utilised in order to have resonator parameters change with time or at least let it appear as if they are changing in time. The simplest of these approaches is to calculate multiple sets of modal data for a single designed system of inter-connected resonator objects, where each set has different values for one or more of the material parameters or inter-connection, excitation, readout or boundary conditions. To give a simple example: for the work *Excite Me* two sets of modal data corresponding to the same system were calculated, each with noticeably different values for the frequency dependent and independent damping coefficients. The first set then corresponded to a two string system with low damping characteristics, so that each string would take several seconds before decaying to silence completely, and the second set in turn corresponded to a two string system with high damping characteristics, thereby taking considerably less time to decay. Two resonators simulating both variations of the model system are used throughout the piece and at times their output is overlapping. This is especially apparent during the final section of the piece starting around 6:50. In the background one can hear a harmonic drone supposed to act as a faint echo of what was audible just before. In the foreground one can hear a chaotically moving, rubbing type of sound. The resonator used for the drone is the one corresponding to the low damping version of the model system, whereas for the chaotically moving sound the high damping version is used. The composite result presents a nice contrast between the resonant quality of the drone and the more direct, dry (but still faintly harmonic) identity of what is happening in the foreground.

Some alternative approaches for changing resonator parameters during the actual simulation stage have been experimented with, although it was found that in most cases the more difficult implementation and significantly increased computing costs did not weigh up to the creative benefits gained. For instance, some experimentation was carried out with calculating two sets of modal data corresponding to the same model system under low and high damping conditions. During the simulation stage, these two sets of modal data were linearly interpolated in order to be able to transform between sounds with different degrees of damping in a continuous fashion. Although I have not invested a huge amount of time on experimenting with this creatively, it was concluded that the 'discrete' approach previously described for *Excite Me* already provided enough in this regard. This is not to say that the 'continuous' approach could have its creative benefits, but with time considerations in mind, it was decided not to pursue this any further. Another interpolation approach which has been used for compositional purposes is that of interpolating between sets of modal data belonging to two different resonator objects, taking inspiration from the commentary of Hans Tutschku on his piece *Eikasia* [5]. This technique has been used to produce some of the sounds for the closing section of *Extase 1*, where after exciting one of the plates the filter coefficients of its resonator are linearly interpolated to that of another plate during the decay stage. Sonically, this will result in a spectral glissando caused by modal frequencies ramping up or down between a start and ending value. In *Extase 1* this effect is rather subtle, as an algorithm was devised which selects modal frequency pairs with the least distance in Hz between them. As such, it may be seen as an exploratory attempt to realise a sound transformation using an approach which is closely related to the physically modelling-based interpolation approach as previously mentioned in the second paragraph of section 1.1. Unfortunately, there was no time to test the robustness of the algorithm thoroughly or to experiment with other relations/rules on which to base the interpolation. Hence, although satisfied with the end result from an artistic point of view, I think this particular morphing strategy needs to be investigated more rigorously in order to be able to make any definitive statements about its proper application.

Chapter 4

Composing with Sound and Timbre

This chapter is meant to provide the reader with more insight in how essential it is for me to see sound and timbre as inherently part of the compositional process. I would even go as far as to say that without this integration it would be impossible for me to compose any music at all, since, as stated already in the introductory chapter of this thesis, I do not fit the role of a classically trained composer who writes music with the help of notational structures and thinks rationally in terms of precisely defined pitch relations, harmonic schemes and/or rhythmical grids. Instead, musical undertakings have their origin in the synthesis and processing of sound itself. Hence the goal is to illustrate how this way of thinking has been employed in the compositional processes that were used to realise the works in the portfolio. Before doing this however, I will first provide some background in order to situate my own approach to composing with sound and timbre in the broader context of electroacoustic music.

4.1 Timbre from a Spectral Perspective

Although I do not make explicit use of any formal musical schemes, the harmonic element is particularly present in the portfolio works. However, in virtually all cases the harmonic structure is dictated by the timbral features of the sound[1] rather than it being imposed by a 'musically correct' formal scheme. Timbre here is to be understood in terms of how a spectralist instrumental composer would be likely to approach this, namely as *"...an extension of harmony, or vice versa"* [77, p. 36]. Although I do not think it would be fair to categorise my work as spectral music in the strictest sense of the word, the idea of equating timbre to harmony has been utilised as a compositional approach frequently throughout the portfolio and hence deserves further explanation. In particular, there is a strong similarity with the musical ideas which have been explored by composer Jean-Claude Risset. In the mid sixties Risset, with the help of Max Mathews, began a detailed study involving the analysis, synthesis and perception of tones created by acoustic instruments, which ultimately resulted in a document entitled *An Introductory Catalogue of Computer Synthesized Sounds* [78, 79]. In addition to providing instructions on how to replicate these sounds through synthesis, the document also describes a number of more compositionally oriented applications. With reference to the last example of the catalogue John Chowning states that it *"...stands as a striking advance in computer music, although little recognised and little exploited"* [80, p. 4]. Specifically, this example is concerned with the fusion of harmony and timbre by first presenting pitched tones sequentially, thereby creating a melodic line, after which the pitched material appears simultaneously to create harmony and lastly, sinusoidal components with the same pitches are presented again simultaneously to invoke the idea of listening to the timbre of a single sound. With regards to this example, Risset says: *"...timbre becomes functional: it constitutes the musical material, but its specific intimate structure relates to harmony, it has implications over the syntax"* [78, p. 5]. Risset typified the process just described as the 'spectral analysis of a chord': *"it is as though the harmonics of the notes of the chord were selected through a frequency window moving from the high to the low frequencies"* [78, p. 5]. The work *Mutations* (1969) [81, t. 6] is the first work in which Risset used the idea just discussed in a compositional context. The first ten seconds constitute of a progression from a melodic line, to a chord and concludes with timbre in the form of a gong-like sound using the same pitch information for all three stages. Similarly, for the work *Inharmonique* (1977) [81, t. 5], Risset layered sinusoidal components with an instant attack and exponential decay to create synthetic imitations of struck bell sounds. However, there are also sections where the

[1]More specifically, all harmonic decision making is informed directly by the modal data associated with a particular custom built virtual instrument.

envelope shapes controlling the amplitude of the individual components are modified and consequently, the bell sounds are turned into fluid textures having the same spectral characteristics. Other than pure synthesis, Risset's work also incorporates striking examples of hybrid materials obtained by transforming recorded, natural sounds using various signal processing techniques. For his work *Sud* (1985) [81, t. 1–3] the dynamic profile of various natural sounds is projected onto the spectrum of synthetic sounds by means of cross-synthesis. In addition, resonant filters tuned to a specific scale are used to gradually imprint specific harmonic characteristics onto the spectra of various natural sounds, like that of sea waves and birds.

These three examples serve to illustrate that Risset's work has contributed significantly to the advance of computer music. Not only did he display the potential of digital sound synthesis for creating idiosyncratic materials and that it offers unprecedented control over sonic detail, but he also managed to expose in an imaginative way, how the creative use of sound synthesis methods suggests an integration of sound and timbre into the process of music composition naturally. Hence, creating musical form and function reduces to the composition of sound and timbre itself, or in the words of Risset: *"With synthesis, one can compose spectra and timbres just as musical chords, and one can attribute a harmonic function to timbre"* [78, p. 6]. Ideas similar to those just discussed have been used throughout the portfolio in order to relate the spectral representation of a virtual instrument[2] to the harmonic structure of the music. In this sense one may say that every designed virtual instrument associated with a specific musical work no longer is used merely as a generator of sound, but also serves a functional purpose; namely to determine the harmonic content and progression of the work. This is made possible exclusively by the specific approach to PM sound synthesis which has been employed for *PMLib*. The ability to construct arbitrary complex virtual instruments out of more elementary building blocks allows one to explore a rich and diverse palette of spectral possibilities. Furthermore, as the result is expressed in terms of modal information, one has the additional opportunity to utilise this to make harmonic decisions that can be related directly to the timbral characteristics of the sounds produced in response to exciting and picking up the vibratory response of the virtual instrument at arbitrary

[2]The terms 'virtual instrument' and 'system of inter-connected (resonator) objects' will be used interchangeably throughout this thesis. In essence, they both correspond to the same thing. However, the context of the discussion will sometimes determine which one of the two is more appropriate.

locations along its surface. A more detailed exposition of how this manifests itself in the composed portfolio works will appear in section 4.4.

4.2 Transitioning between Spectral and Spectromorphological Perspectives

Other than Risset's contributions to computer music and digital sound synthesis, it may be argued that his concern about the treatment of timbre in relation to harmony built the foundation for the experiments carried out in the seventies and early eighties by the second phase of predominantly French spectral composers. Several of them became fascinated by writing instrumental music which aimed at replicating harmonic and inharmonic spectra obtained from FM and ring modulation sound synthesis techniques, but also from real-world bell sounds and multi-phonics [82]. It is generally stated that spectral music is not so much about technique or style, but rather is to be interpreted as an attitude towards the treatment of materials. Hence, it seems easier to define what spectral music is not as opposed to what it is. Referring to the aims of the spectral composer, Joshua Fineberg states that *"the only true constant for all these composers is that they consider music to ultimately be sound and see composition as the sculpting in time of those sounds that a listener will hear"* [83, p. 3]. Although this may be true, it hardly can be used to make a clear distinction between spectralist and non-spectralist composers, as this statement can be wholly seen in the context of a spectromorphological approach to composition. Denis Smalley defines the term 'spectromorphology' as *"the interaction between sound spectra (spectro-) and the ways they change and are shaped through time (-morphology)"* [75, p. 107]. Although Smalley claims that spectromorphology is not to be interpreted as a compositional theory, he suggests that it can be used to inform compositional thinking. From both Fineberg's as well as Smalley's quote we may infer the same focus on shaping sound over time as pivotal to the compositional process. However, timbre in spectromorphological terms is to be interpreted as the means by which to identify the supposed source sounds are emanating from [77] rather than to see it strictly in the context of harmony as would be common in spectral composition.

It is relevant to point out this conceptual similarity with regards to treating sound in the compositional process, but differing interpretations of the word timbre, as aside from the clear presence and importance of timbre with regards to harmony, an emphasis on timbral aspects from a spectromorphological point of view is also prevalent in the portfolio. Many of the digitally created sound materials are derived from custom designed sound models which try to simulate, or at least capture, some of the sonic characteristics of a certain natural sound scene or physical interaction, thereby placing emphasis on their distinctive timbral and micro-structural features. It is hoped that by doing so, I have managed to create a certain complexity to the music which transcends the purely abstract, internal relationship between sounds and events commonly ascribed to computer music, but which instead allows the listener to connect with 'the world of lived experience' [84]. At the same time I want to make perfectly clear that I think it is somewhat of a narrow-minded presumption to state that for music to have any form of complexity it has to be situated in the context of the real world. On the contrary, music which is abstracted from the real world has the potential to invoke unique associations and imagery that go beyond what one is able to relate to rationally. The act of abstracting offers one the ability to tap into a whole area of meta-realistic experiences which are likely to be very personal and hence unique for everybody. It is exactly because one might not be able to make a direct association with the real world and common experience that one can become fascinated with some abstract phenomenon. The interplay between abstraction and realism has been used as a recurring compositional strategy throughout the portfolio, which will be discussed in more detail in section 4.5 in relation to the use of sound transformations as a means of traversing the space between the real and abstract. Before doing this however, I want to give a more clear definition of what real and abstract means in the context of this research. In other words: how are these rather subjective gradations to be interpreted when discussing material generated through both abstract[3] and PM sound synthesis techniques?

[3]It is rather unfortunate that the same word is used in two different contexts. Here the word 'abstract' is solely meant to differentiate between physical modelling and other, possibly physically inspired, sound synthesis methodologies [20], and not to characterise the perceptual qualities of the actual sound produced by either technique.

4.3 Defining Physical Plausibility of a Sound

In an attempt to quantify better the concept of realism as it applies to the generated materials for the portfolio, it seems useful to introduce the notion of the 'physical plausibility' of a sound. Before defining this more definitively, let us first consider the Schaefferian concept of 'reduced listening'; i.e. the idea of listening to the intrinsic properties of a sound without making any external references [14]. Such a formalistic approach to analysing sounds is often refuted, since *"it proves very difficult to hear sound only in terms of an appreciation of its shape and spectral properties as Schaeffer seemed to advocate"* [85, p. 136]. It is more generally believed to be true that the human ear tries to relate sounds (either artificially generated or recorded) to a possible source and cause naturally. This is reflected in Denis Smalley's concept of 'source bonding' [75]. It is particularly interesting that Smalley's definition of bonding allows for a relative interpretation by including highly personalised bonding experiences which might be completely different from what was envisioned by the composer. Hence, this is supporting my belief that an existing engagement between listener and musical material is not critically dependent on having to situate this in the confines of the real world, as external associations may be imagined and distinct for everyone. Similarly, Trevor Wishart has put forward the notion of 'landscape' to denote our recognition of the supposed (i.e. imagined or real) source of the sound [86]. Additionally, he makes a further difference between the recognition of the sound as an object and the perceived acoustic space these sounding objects reside in. Hence, this offers the possibility of making a distinction not only between real and abstract sounds, but also between real and abstract spaces. The concept of a resonator model of a virtual instrument acting as a medium through which to transmit both seemingly real and abstract sound will prove interesting later on when discussing some of the portfolio works in more detail. More specifically, some examples will be given where resonator modules have been used to add additional coloration to sound already possessing a certain complexity, effectively imposing the spectral profile of a virtual instrument onto that of a sounding object. Furthermore, unique approaches to spatialisation become possible as sound output may be taken from multiple points along the surface of a virtual instrument simultaneously, providing the opportunity for disseminating the music over multi-channel setups according to a spectrally motivated de-correlation of the sounding materials. This will be discussed in more detail in section 4.6.

Returning to the topic at hand, the 'reduced listening' concept is perhaps more easily applicable to sounds generated through (abstract) sound synthesis methods. Since the real source of the sound is not a physical one, the listener might be more tempted to focus on its internal qualities at first instance. However, alluding to the earlier given definition of source bonding, even abstract sound is not completely free from association. In relation to this, Risset states that *"synthesised sounds will be more easily accepted by listeners and have a better profile when they lead the subject to think they were produced in some physical manner"* [87], thereby implying that by being able to perceive a certain level of physicality, it will contribute to the level of realism that may be ascribed to a synthesised sound. In [26] the notion of 'physical plausibility' of a sound is introduced to which end the authors stress that: *"The important feature for a musical sound is not to cause the listener to infer its physical cause, but to present a set of subtle dynamic variations among perceptual parameters that lead the listener to think it was produced in some physical manner"* [26, p. 1]. Initially this statement seems a little contradictory, since one may argue that by supplying subtle dynamic variations among perceptual parameters in order to 'trick' the listener into thinking the sound was produced in a physical manner one is effectively creating a deliberately modelled physical cause for it. However, this modelled source or cause doesn't need to correspond to any real-world equivalent necessarily, while still being able to possess a certain physically plausible character. To this end, Trevor Wishart has formulated several archetypes (e.g. turbulence, crack, shatter etc.) to identify those sounds which source-cause relationship might appear to be ambiguous, but still possess a complex, physical plausible morphology [88]. Either way, Smalley's definition of 'source bonding' still applies to these type of hybrid sounds. Hence, in the context of my research, the term 'physical plausibility' is proportional to the level of realism one is able to ascribe to a synthesised sound. At one end of the extreme, this implies sounds which are instantly recognisable to be of a physical nature, even though they have been realised entirely through artificial means. At the other end, we have total abstraction and hence sounds which cannot be clearly associated with a definitive physical interaction or material object. In between these two extremes there exists a continuous domain which may be partitioned according to Smalley's idea of increasing orders of surrogacy or remoteness from known sources and physical gesture [89]. Abstraction may then be interpreted as *"a measure of the psychological distance between a sound which displays source-cause ambiguity and an assumed source-cause model"* [90, p. 79] and thus provides a means of navigating between the different levels of surrogacy. The notion of physical plausibility hence proves to be a useful conceptual construct for assessing the realness/abstractness of a synthesised sound, although one has to bare in mind that a certain level of subjectivity is unavoidable. Lastly, a distinction has to be made between the synthesis of physically plausible sounds and sounds which just have a time-varying dynamic and timbral profile, since these are not necessarily congruent. The latter clearly involves a more general category of sounds which includes physical plausible sounds as well as abstract sounds which have no direct physical connotation and hence could benefit from a more general analytic description[4].

4.4 Harmony from Timbre: Spectrally Informed Decision Making

Several different compositional approaches have been investigated in order to produce harmonic structures and progressions, all of which can be traced back to the modal data associated with a particular virtual instrument. Since for every work a dedicated virtual instrument has been designed, the harmonic diversity of each of the works is more or less proportional to the complexity of the designed system. Each virtual instrument consists of a system of inter-connected 1D and 2D resonator objects. Each of these objects in turn may be excited and/or listened to. Therefore in principal, the more objects the system contains, the greater the number of different timbral options will be, as each object has its own spectral identity it imparts on the total system. However, by designing excitation models whose spectral characteristics are made to vary dynamically, even relatively simple systems may exhibit complex variations in timbre over time.

To give a first example; for the work *Excite Me* the designed virtual instrument consists of a system of two strings connected to each other at one of their end points, while their other end points are fixed according to simply supported boundary conditions[5] (see figure 4.1). Only the first string is excited and sound is obtained from the vibratory response of the first string in response to this excitation only, the idea being that the

[4]Gottfried Michael Koenig's proposition of *physical complexity* versus *aesthetic complexity* [91] is an interesting alternative to consider alongside Denis Smalley's spectromorhological framework in this regard.

[5]At the time of composing *Excite Me*, the software library *PMLib* was still under development. At that time it was only possible to inter-connect 1D objects and to excite and pick up the vibrational response of the strings at a single location along their surface.

effect of the second string would only be indirectly present through sympathetic vibrations[6]. The inter-connection of the two strings in combination with the chosen values

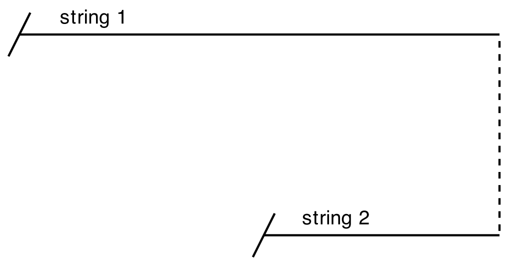

FIGURE 4.1: A simple system of two inter-connected strings. The dashed line represents the rigid connection between the right end points of the strings.

for their material parameters has quite an interesting and unique effect on the spectral characteristics of the system. In an attempt to expose the harmonic complexity of the two string system dynamically over time, an approach inspired by Risset's concept of the 'spectral analysis of a chord' was used to create a continuously changing drone texture, appearing in the section of the work which approximately starts at 4:15 and ends at 6:50. The underlying mechanism which produces this continuous change in timbre is most easily understood in terms of focusing on the interpretation of the resonator simulating the response of the first string as a set of filter sections connected in parallel. Each section[7] is 'listening' for a specific frequency in whatever signal it receives at its input. Hence the magnitude of the output of a single section is proportional to the amount of energy present in the input signal around that frequency. Therefore, if the input to the parallel arrangement of filter sections consists of a spectrally rich signal which contains significant, time-varying variations in energy along its spectral dimension, the magnitude/gain of the output of each individual section will be time-varying as well[8]. This may be seen as a variation of Risset's approach earlier described with regards to his work *Sud*. It is as if one is moving a sonic magnifying glass along the spectrum of, in this case, a physically modelled object in order to emphasise and de-emphasise certain of its characteristic timbral features in a dynamic, time varying way. The most noticeable difference between my approach and Risset's, is that firstly, in my case the input

[6]Sympathetic vibrations occur when an object starts to vibrate in response to exciting another object due to a mechanism which allows energy to be exchanged between those objects (in this case the rigid connection between the two strings).

[7]Recall that in the context of *PMLib*, each section is corresponding to a single second order resonating filter simulating the response of a single modal frequency.

[8]This is an effect which is in addition to the different, but fixed magnitude of each mode which is determined by the projection of the computed modal shapes onto a particular choice of excitation and readout location along the surface of the virtual instrument.

signal for the filter bank is synthesised as opposed to recorded, and secondly, that the number of filter sections is considerably larger and, when summed together, represents a virtual impression of an actual physical object, arguably giving a much richer, and physically plausible end result. In order to illustrate this process visually, figure 4.2 shows a spectral representation of a small portion of the drone section. One can clearly see the continuous curve-like trajectories corresponding to time-varying gaps in the spectrum of the drone, which sonically translates into a continuous, rich, but dynamically varying fluid sound texture. This serves as a good example of a sound which tends towards the abstract end of the physically plausible sound scale, as I think that the uninformed listener will have difficulty in determining both the cause (i.e. the action which supplies the continuous amount of energy to keep the sound going) as well as the source of the sound, as it is not very likely that one will associate the character of the sound produced by the modelled string to that of a real string any longer.

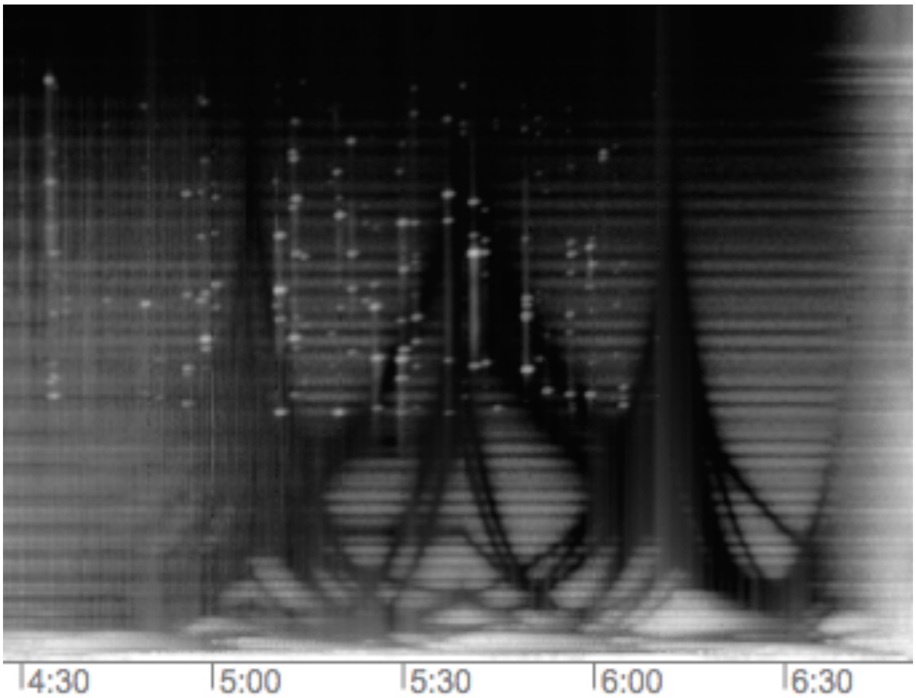

FIGURE 4.2: Spectrum of a section of *Excite Me*. The horizontal axis denotes time running from approximately 4:30 to 6:40, the vertical axis denotes frequency running from 0 to 22050 Hz. White indicates the areas of the spectrum which contain the most energy, black indicates the areas not containing any energy at that time instant.

An identical, slightly enhanced approach as just discussed has been used to create one of the continuous sound layers for *Extase 4*, which starts to take shape around 14:30 and lasts until the end of the work *Extase 2, 3 & 4*. It would be a little redundant to reiterate the aforementioned approach in the context of *Extase 4*, as I feel it will not add anything new to the current discussion. Instead, I want to focus on another way of deriving harmony from timbre utilised for *Extase 3*, which can be seen as a combination of Risset's additive approach used in *Mutations* and his resonant filtering approach employed for *Sud*. In terms of technical implementation, it is very similar to the technique previously discussed in section 3.4.1.2 with regards to generating the materials for *Extase 1*. However, the main difference between the excitation model used for *Extase 1* and that for *Extase 3* is that, instead of using filtered single sample impulses, the source for the excitation model is generated using the FOF synthesis technique [92], as it was found empirically that this gave a more convincing impression of a bowing-like interaction when used to excite a 1D or 2D resonator module. The designed virtual instrument for *Extase 2, 3 & 4* is considerably more complex than the one designed for *Excite Me*. It consists of two plates with different material parameters which are inter-connected to each other through the end points of six differently tuned strings (see figure 4.3). This more complex system allows for a greater harmonic diversity compared to the simpler two string system, since obviously, many more possible excitation and readout location combinations exist, each having their own distinct spectral profile. The rationale behind dividing the complete work up into three separate parts is based upon the fact that each of them focusses on deriving the harmonic content from a specific excitation and readout location along the surface of the virtual instrument. As such, each part has a clearly recognisable timbral (i.e. timbral implying harmonic here) identity. For *Extase 3* (which starts around 5:00 and ends around 12:45) the generated sound material is made up of different layers, each of which correspond to the vibrational response of one of the six strings in response to exciting that string using the FOF synthesis model. The frequency input of the FOF model is set to modulate randomly and slowly around one of the first twelve modal frequencies of one of the six strings, thereby generating a spectrally rich, pulse-like signal with a slight vibrato that has energy mainly at the fundamental and integer multiples of this modal frequency. By using this as an excitation for a resonator module simulating the string itself, the spectral profile of the string is imprinted onto the signal, resulting in an extra resonant character of that modal frequency and any of its harmonics if they happen to coincide with one of the other pronounced modal frequencies of the string. As such, the six possible pitch choices resulting from the specific tuning of the six strings is extended to include the first twelve harmonics of each of the six strings as well. Practically speaking however, most

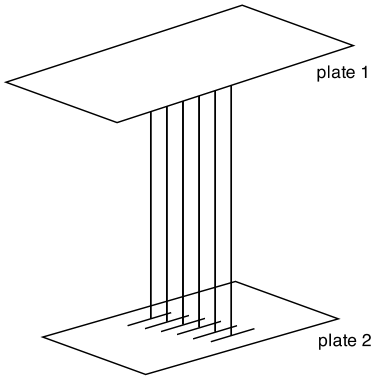

FIGURE 4.3: A system consisting of two plates with different material parameters inter-connected by means of six differently tuned strings. Note that the depicted connection points between plate and strings appear structured to make the visualisation more clear. In reality the connection points between the plates and strings are randomised.

of the higher harmonics are hardly used, as they produce very high pitched and rather undefined tones. Nonetheless, by layering the sounds of the different strings and sounds from the same string excited at different modal frequencies, this approach allows for the creation of rich, spectrally informed harmonic textures that expose the inner timbral detail of the strings. The effect from driving a single string at different modal frequencies is best perceivable in the second, rather inharmonic sounding section of *Extase 3*, starting around 10:30. This section features sounds obtained from the first and third string only, thereby providing a striking example of the tonal diversity which may be obtained from the fixed spectrum of only two of the six strings. As mentioned before, *Extase 1* uses the same spectrally informed technique to produce differently pitched sounds from the first twelve modal frequencies associated with one of four inter-connected plates with different material parameters (see figure 4.4). In fact, most sound material that is being used in the final version of the work originates from the first two plates predominantly, as these were found to create the most harmonic homogeny. Whereas for *Extase 3* the harmonic development could still be partly attributed to the different tuning of the six strings, the different harmonic progressions that appear in *Extase 1* are derived entirely from the spectrum of the plates. The spectrum of a rectangular plate is inharmonic by definition. Hence, the harmonic sounding results were produced through layering individual sounds with a clear pitch obtained from driving the plates at one of the first few modal frequencies and using simple tape speed variation techniques in order to create copies which are transposed down in octaval ratios. The opening event of *Extase 1*, spanning the first thirty seconds, consists of a single, untransposed, pitched sound with a bow-like quality produced by this method, and serves as a good example of how one of these basic harmonic building blocks sound in isolation.

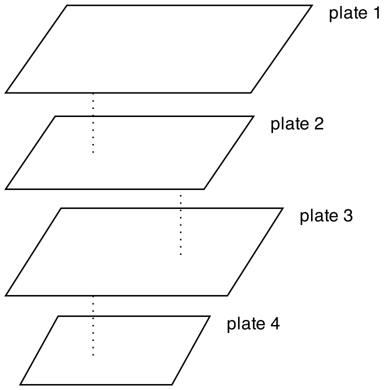

FIGURE 4.4: A system consisting of four inter-connected plates with different material parameters. The dashed lines represent the rigid connection between two consecutive plates.

As a last example of yet another different way of deriving dynamically changing harmonic textures from the spectral characteristics of a designed virtual instrument, I want to discuss the approach followed for the drone layer, starting around 2:15 and lasting until 8:00, making up most of the middle section of the work *Stable Equilibrium.* The virtual instrument designed for *Stable Equilibrium* is a system consisting of a single plate with six differently tuned strings attached through one of their end points, while fixing their other end points according to simply supported boundary conditions (see figure 4.5). During the drone section the stiffness of the plate is set considerably lower than for the opening section, resulting in an increase of lower modal frequencies. This is clearly audible when one compares the first two minutes of the work with the drone section as the harmonic centre is noticeably lower for the latter. The approach for creating the dynamic variations in timbre throughout the duration of the drone is based on somewhat of an inverse approach to that used for the drone section of *Excite Me* described earlier. The gaps in the spectrum of the excitation signal for *Excite Me* were being created by notch filtering in series a noise source, whereas for *Stable Equilibrium* the excitation signal consists of a noise source which is band pass filtered in parallel. The centre frequencies of the band pass filters are fixed and are set away from any modal frequencies of the plate deliberately, in order to avoid overly strong resonances in combination with the resonator module simulating the plate. The frequency bandwidth is set quite narrow in order to produce a pitched tone and the gain of each band pass filter is slowly modulated over time, thereby giving the impression that the different bands weave in and out of focus throughout the duration of the drone. Furthermore, instead of starting and stopping all the band pass filters at the same time, the start time of the band pass filters branch out from the middlemost center frequency. This can be perceived aurally by comparing the timbre of the drone towards the start of the event around 4:00, where the spectral focus is concentrated in the mid frequency range, and towards the end of the event around 7:00, where it is mainly the low and high frequency range that dominates. For the entire duration of the drone, the parameters of the resonator are fixed. Hence, all perceivable changes in the timbre of the drone are caused solely by the carefully composed spectral evolution of the excitation signal, using again a combination of layering tonal components and physically reasoned filtering (i.e. the MM approach to PM sound synthesis). In terms of realism and physical plausibility, I imagine that most listeners would place the drone sound in the abstract category, although it may be argued that the rather resonant quality does imply that the sound could be emanating from a physical source, which perhaps is just 'played' (i.e. excited) in an unusual way, which in fact is exactly what is happening. Hence, this approach is likely to produce a more 'realistic' sounding result than the aforementioned one, used for creating the drone layer for *Excite Me* and *Extase 4*. However, both approaches seem valid from a creative perspective in my opinion, as they have shown extremely useful for producing dynamic, time-varying variations in timbre and the harmonic dimension.

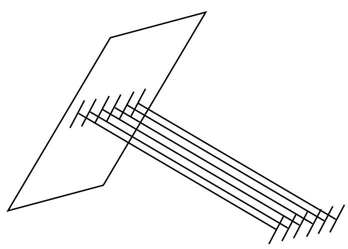

FIGURE 4.5: A system consisting of a plate with six differently tuned strings attached through one of their end points. The other end points of the string are fixed according to simply supported boundary conditions.

The discussed examples so far bring up a potential issue which I think is necessary to address appropriately. One could argue that investing time and effort into the physical modelling of an object like a string is somewhat pointless if ultimately one is content with an end result which is not relatable to the representative sound of that object anymore, thereby possibly inadvertently implying that any other, perhaps simpler synthesis or processing method would have sufficed. Nonetheless, I believe that even in an extreme case of remoteness from the 'typical' sound of the representative physical object[9] the approaches I have followed are still valid from an artistic point of view. Firstly, and as will be discussed more attentively in section 4.5, in the majority of the works there is a constant interplay between seemingly real and abstract sounds through carefully composed transformations. Many of these transformations start from a realistic sound scene and through continuous variations in model parameters are allowed to morph into something more abstract, but hopefully at least still faintly physical plausible. Secondly, the chosen physical modelling approach, in fact makes things more easy from a practical point of view. It frees me from the task of having to specify any harmonic related decisions explicitly, since these are largely the byproduct of designing systems of inter-connected objects and experimenting with different sounding excitation models. Thus, complex timbre and resulting harmonic relations arise from specifying a handful of physically relatable parameters only, in combination with an emphasis on the spectral shaping of excitation signals. This is convenient for me, as it allows me to focus on designing sounds and timbre and assessing the results by ear, rather than having to rely on some abstract musical formal scheme or rule set. My arguments are meant to demonstrate my intent to focus on the expressive power and potential of the presented models and techniques in this section with regards to generating unique materials through exploring the boundaries of their sonic possibilities. If one is at all interested in creating something novel or unexpected, it is more likely that this will be found by venturing out to the extremes rather than to remain in the safe confines of what is already known and expected.

4.5 Linking the Real and Abstract through Transformation

It seems appropriate at this point to move on from focussing on timbre from a harmonic perspective only, to some examples that will deal with timbre in a more general, spectromorphological setting. In particular, I want to illustrate the different ways in which

[9] However that may be defined rigorously in the first place.

sound transformations have been used to modify timbral aspects dynamically, and to what extent this can change the perception of the listener. A sound transformation may involve the creation of sonic hybrids by fusing two or more characteristic features of distinct sound objects [93]. For instance, one may impose the spectral profile of one sound onto the dynamic profile of another. Examples of these kind of transformations have in fact already been discussed in the previous section with regards to creating the continuously evolving drone layers. However, the word transformation also suggests that it is some process that evolves over time and by doing so, will offer the possibility of changing the source-cause perception of the listener with time. Although in the previous examples one can indeed speak of a transformation in terms of producing a sonic hybrid, which will certainly influence the listener's ability to determine the supposed source and cause of the sound, it is not very likely that the perception of the listener will change over time. In other words, the possible source-cause ambiguity resulting from an abstract, imposed morphology (created by the sustained, but continuously filtered noise excitation) in combination with the realistic, intrinsic morphology of the resonator modelling (a part of) the system of inter-connected objects, is static throughout the duration of a drone. Here, imposed and intrinsic morphology refers to Trevor Wishart's proposition of expressing the difference between the internal resonating properties of a sound source and the continuous or iterative energy input (i.e. cause) required to continue sound production [88]. However, there are also examples throughout the portfolio which try to utilise sound transformations in order to alter the perception of the listener regarding the supposed source or cause in a time-varying way. This is accomplished by transforming between seemingly real and abstract sound settings where the listener is constantly challenged to assess if the presented material is real, abstract or something in between. In the context of this research, these type of transformations can be divided roughly into two different categories: continuous horizontal (i.e. time-varying) transformations on excitation and abstract synthesis signals only and continuous horizontal transformations which involve the gradual imposition of the spectral profile of a virtual instrument onto natural sounding events, which can have a dramatic effect on the source-cause perception of the listener.

As an example of a transformation procedure used on materials obtained from excitation models exclusively, I would like to refer to the opening section of *Extase 3* which starts at about 4:00. At this time instant a sound layer reminiscent of flowing water starts to develop, which reaches its apotheosis around 7:20 and then gradually merges into the bowed string-like sound until it vanishes completely around 8:00. At the start of the event, my intent was to focus on a realistic sounding setting, as in combination with the additional layers of sound the impression of a rain storm is envisioned. However, as the section progresses, the parameters of the synthesis model are modified in a continuous manner in order to transform the sound layer into something which still retains some of the perceptual stream-like qualities of water, but yet doesn't sound like water anymore towards the end. As such, the perception of the listener is encouraged to change from a conviction of listening to something physical plausible (i.e. the natural sound of flowing water) to something still having a water-like quality to it, but at the same time sounding sufficiently abstracted away from water to the point that the source of the sound attains a certain level of ambiguity. Another example may be found in the opening section of *Extase 2*, which consists of a sonification of an imaginary beach. From the opening until 1:00, I wanted to explore the idea of synthesising a realistic sounding setting, which leaves no doubt regarding the imagined source of the sound. However, during the following thirty seconds, the same wave model is used to slowly fade in a secondary, less realistic sounding layer, which serves to introduce an increased sense of ambiguity regarding the origin of the sound and was found to accentuate rather effectively the abrupt transition to the next, resonant sounding section starting at 1:40.

Sticking with the synthesised wave model for the moment, it was used as an excitation signal for a resonator model simulating one of the two plates for *Extase 4* (see figure 4.3) to create a sound layer subjected to a gradual transformation that radically changes the source-cause perception of the listener over its duration. From a technical point of view the transformation involves nothing more than dynamically controlling the ratio of synthesised wave signal routed through the resonator module to 'dry', unaffected signal send directly to the output. At the start of the event, around 16:15, all the listener is able to perceive is a harmonically rich drone, produced by the intrinsic morphology of the plate resonator in response to the energy supplied by the synthesised wave model. At this point, until approximately just before 17:55, the nature of the imposed morphology is completely hidden from the listener. It is only after 17:55, when the ratio between resonator and direct output begins to decrease beyond unity, that one starts to perceive consciously the characteristic sound of the waves again, albeit with a harmonic sounding tail at first. From start to finish this particular transformation will accomplish two things. Firstly, by removing the source-cause ambiguity gradually it is envisioned that the event will transition from the most remote order of surrogacy all the way up to first-order surrogacy and hence makes possible a continuous transition from complete abstraction back to simulated reality. Secondly, by slowly exposing the characteristic timbral qualities of the waves, the listener is encouraged to make a connection with the start of the work, thereby letting the listening experience come full circle again. Along the same lines, the work *Extase 4 Version* (which as its name suggests, is a variation on *Extase 4*) is exploring a similar type of transformation, although this is happening over an extended duration and is staying in a more abstract sound setting altogether. In this case the transformation takes place not so much with the aim of transitioning between seemingly realistic and abstract sound settings, but predominantly serves as a means of transitioning smoothly between non-tonal and tonal materials. As a last example of a transformation belonging to this category as well as to the one discussed first, I want to redraw attention to the excitation model described previously in section 3.4.1.1. Worth mentioning in the current context, is that this example serves again as an illustration of how a transformation of this kind can alter the perception of the listener regarding the imagined source and cause of the resulting sound. Additionally, it is interesting to note that during the progression of this particular event, the imagined source is allowed to switch from one realistic setting (i.e. rain drops at 1:30) to another (i.e. a dense collection of string plucks at 3:20) before condensing into total abstraction at 5:00.

4.6 Spatialisation by Spectral De-correlation

Throughout the portfolio different spatialisation techniques have been employed, which are similar in the sense that they all try to present the musical material using a fixed, wide spatial image by assigning de-correlated audio signals to the physical output channels (i.e. the speakers). De-correlated audio signals in the context of this research are to be interpreted as multi-channel signals which are different, but perceptually similar [94]. Acquiring a de-correlated stereo signal from a monophonic source, may be achieved by duplication and simply introducing a slight time delay between the two copies, which is commonly referred to as temporal inter-channel de-correlation [95]. When it comes to sound synthesis, de-correlated signals may be obtained by allowing for slight differences in the sound synthesised for each separate output channel. To give a simple example: a completely random, but monophonic signal presented over a 2-channel setup will appear to emanate from a location somewhere in between the two speakers, whereas placing a fundamentally different, but perceptually identical (i.e. de-correlated) random signal on each speaker will give a highly diffuse effect, not favouring one particular direction in terms of the assumed source of the signal. Likewise, randomly distributed, impulse-like sounds or very grainy textures will be perceived completely de-correlated when the synthesis process producing individual grains/pulses is independent for every output channel. These two simplified examples are not chosen at random, as all source signals used for the basis of the designed excitation signals are noise or impulse based. In virtually all cases, the highest possible de-correlation of the presented material was desired in order to not favour any one spatial direction in particular, the intent being to augment the impression that the listener is immersed by the music from every angle, regardless of ones vantage point. As such, this is closely related to Denis Smalley's concept of 'immersive space': *"The filling of spectral and perspectival space in circumspace so that the listener feels immersed in the image"* [96, p. 55].

The choice for this approach to spatialisation is primarily based on the fact that the majority of the portfolio is emphasising the development of texture over time. I wanted to maximise the impression that the listener is situated inside the sound, and hence is oblivious to any particular direction these textures are originating from. I think that the immersive experience would have been much less effective if instead the listener would have been able to trace the path of the sound through space according to some clearly definable trajectory. However, there is something to be said for the employed approach as well by arguing that it contributes positively to the amount of realism which may be ascribed to many of the synthetically generated natural sound scenes which appear throughout the portfolio. Take the opening section of *Extase 2* for instance, which as previously mentioned, consists entirely of digitally modelled sea waves crashing on an imaginary beach. The source of the wave sounds can be traced back to a noise generator, producing as many de-correlated versions of the noise signal as there are output channels. The fact that multiple de-correlated noise signals are used instead of one monophonic noise source, ensures that an extra and important spatiality is added to the sound. I strongly believe that this helps towards creating a more realistic sounding experience, as in a way it better captures the width in the sound one perceives when standing on a real beach. This width can be explained partly due to the fact that the sound of a real wave crashing on a beach is not simply emanating from a single point source. Instead, it is the cumulative effect of the whole spatial extent of the wave interacting with the beach which is creating multiple sound sources, reaching the ear at slightly different times and intensities. Hence, and although hugely simplified, it makes sense to try to account for this naturally occurring effect to some degree.

Using the same idea of de-correlating the separate output channels, another approach to spatialisation, made possible uniquely through PM sound synthesis, is something I will refer to as spectral de-correlation for lack of a better term. The difference between the aforementioned approach and that of spectral de-correlation, is that the latter is introducing differences in the spectral composition of a sound in order to create distinct, but perceptually similar variants instead of using differences in time. This is accomplished by projecting a monophonic excitation signal through a virtual instrument at different excitation and/or readout locations along its surface. Changing the excitation and readout location will have an effect on the timbre of the sound obtained from a resonator modelling a virtual object. This may be understood best by considering a real-life scenario of plucking a string, fixed at both ends, at different locations along its length. Plucking positions near the end point of the string will emphasise the higher modal frequencies and suppress the lowest modal frequencies, whereas a plucking position exactly one third along the length of the string will suppress every third integer multiple modal frequency of the string. The sounds produced under all these different excitation conditions will still be perceived as that of a string, even though there is likely to be a noticeable difference in timbre. From the variations in timbre due to different excitation and readout locations, a listener may infer spatial information related to the object being modelled, like the relative position at which an excitation signal is applied for instance. One way to accomplish this is by assigning multiple, fixed readout points to separate output channels arranged in the same spatial configuration as the readout points and moving the excitation location at the same time sound is projected through the virtual instrument [97]. In the context of this research, spectral de-correlation has been used alongside, and at times in conjunction with, temporal de-correlation to enhance the immersive experience. It has proven especially effective when applied to textural sounds (like the drone layers described in section 4.4) to make them appear more spatially wide, without favouring any one direction in particular. As such, it could be reasoned that one is listening from within rather than to the virtual instrument. Creating multi-channel versions of works also becomes much easier, as in principal all one needs to do is to simulate the vibratory response of the virtual instrument at as many excitation or readout points as there are output channels. This is exactly the approach followed for extending the 2-channel version of *Extase 1* to the 8-channel version. The 24-channel version, which is supposed to be arranged according to three levels of different height, each containing eight channels per level, uses the 8-channel version for every level, but makes a further distinction by assigning the sounds obtained from one readout location to a specific level based on its frequency content.

One other approach to spatialisation, developed by Miguel Negrão as part of the *ImmLib* library for spatial composition [98], deserves brief mentioning as it was used for creating the multi-channel version of *Extase 2, 3 & 4*, which can be seen as a further extension of the idea of spatialisation by audio de-correlation. In short, multiple de-correlated audio signals generated by the same synthesis process are each assigned a position in space. The main idea is that the evolution of a synthesis parameter at a specific point in space is controlled by a so-called ‘parameter field’, which is a mathematical function defined on a surface embedded in three dimensional (3D) space. This synthesis parameter can be anything in principle, although in the case of *Extase 2, 3 & 4* only amplitude is affected. It proved to be too computationally expensive to assign to each point in space a dedicated resonator module in real-time. Hence, it was decided instead, to spread the sound obtained from a single readout point across the multiple points in space. The spectral de-correlation effect is even more extreme in this case, as modal frequencies are simulated at one location in space only, whereas for the aforementioned approach, a single modal frequency could still be present at multiple points in space[10].

[10]They all would have different intensities of course, thereby creating the difference in timbre caused by the non-similar excitation and readout locations.

Chapter 5

Composing with Texture

The ideas and concepts expressed by the composers referred to in the previous chapter have been of major influence regarding my own thoughts about actively involving sound and timbre in the compositional process. As such, I deeply respect and acknowledge their technical and theoretical contributions to the fields of computer and electroacoustic music. However, if one were to ask me which composers I find truly inspiring from a purely musical point of view, I would have to confess that none of the aforementioned composers would come to mind naturally. Despite respecting their creative efforts, the satisfaction I hope to get from listening to a piece of music generally is found elsewhere. On the one hand this might seem a little strange, as one would expect that if there is such a close connection with creative ideas and concepts on a theoretical level, surely there must be at least some genuine appreciation for the way these ideas are perceived sonically. Nonetheless, a difference exists between agreeing on an interesting theoretical concept and how this concept materialises itself through creative expression. Now matters of aesthetics and personal preference come into play and this is exactly where a composer is able to inject a certain level of authenticity and originality in a work. This was in fact already concluded in section 3.3 more or less while discussing the shift in the role of the composer with regards to the use of algorithmic procedures for creative purposes. In an attempt to expose the contributions of my research to the field of electroacoustic and computer music, this chapter is meant to provide further insight in the aesthetic choices I have made with regards to the portfolio and where these choices are based on.

5.1 Gesture and Texture in Electroacoustic Music

It is not my intent to give a thorough and precise definition of musical gesture. Instead I will be guided by what Smalley says about gesture in the context of electroacoustic music: *"The notion of gesture as a forming principle is concerned with propelling time forwards, with moving away from one goal towards the next goal in the structure... Gestural music, then, is governed by a sense of forward motion, of linearity, of narrativity"* [75, p. 113]. Although a very gestural motivated approach to composition indeed may prove very effective for creating a sense of forward motion, there also exists a risk of saturating the listening experience by bombarding the listener with a seemingly never ending flow of events which do not exhibit much internal interest, but are characterised mainly by the gestural shape they attain. As such, a careful treatment and exposition of the inner details of the sounding materials is neglected and the resulting music can sound overly methodic. This methodic approach to electroacoustic music composition is reflected in Manuella Blackburn's proposition of turning Smalley's descriptive spectromorphological framework into a functional tool to direct compositional decision making. With regards to this, Blackburn states that: *"...vocabulary no longer functions descriptively; instead the vocabulary precedes the composition, directing the path the composer takes within a piece"* [99, p. 1]. To me, this approach seems a little dubious as I do not see how abstract textual or visual constructs can inform musical creativity directly if, at least initially, they are not to be associated with an actual sounding material. This is in stark contrast with how I (inspired by people like Risset) prefer to work; namely to let the sounding material dictate how it should be used structurally rather than to impose this through some artificially designed construct. In fairness, Blackburn elaborates that her spectromorphological informed approaches should be regarded as aids rather than solutions or formulas and that they merely form starting points which consecutively are subjected to intuitive decision making processes [100]. Indeed, when listening to her music, one has to acknowledge that she has an acute ear for sonic detail and musical shaping. It is important to stress that I am not set out to pose that, by definition, all gesturally-minded electroacoustic music lacks a certain care for the subtleties and intrinsic sonic qualities of its sounding materials. However, I do think that a composer needs to be aware of the fact that the gestural shaping of sounds according to spectromorphological informed, but nonetheless pre-determined, functional constructs will be much more interesting and powerful from a sonic point of view when they are directly influenced by, or somehow related to, these intrinsic sonic qualities. A good example in this regard is the work *...et ainsi de suite...* (1992) [101, t. 2] by composer Jonty Harrison. The base materials for this work consist of impact sounds of several wine glasses. These sounds are subjected to various transformational processes, some of which change the timbre of the original sound more radical than others. What strikes me instantly, is that Harrison's careful treatment of the materials has resulted into a gestural work that manages to enhance the experience of listening to the particular resonant qualities of the original materials. I especially appreciate the first one and a half minutes of the work, where the source of the sound is kept recognisable and musical gestures are directly derived from the impact behaviour of the wine glasses. Consequently, the first section of the portfolio work *Stable Equilibrium* was inspired by this. I have attempted to model a similar inter-action behaviour between colliding objects, using a combination of physical modelling sound synthesis to model the resonances of the objects and a physically inspired synthesis approach for modelling the impact behaviour (i.e. to determine the rate of change and 'hardness' of successive impacts). More on this will be said in section 5.2.

Smalley denotes texture as the opposite of gesture, in that: *"a music which is primarily textural concentrates on internal activity at the expense of forward impetus"* [75, p. 113–114]. Drone[1] music may be considered to fall into this category, as in the most strict interpretation of the genre the exterior of the music will appear to change very slowly or not at all. A commonly heard preconception about drone music is that it sounds static or overly repetitive and therefore is by definition boring. However, this reasoning is too simplistic I find, as a well composed drone is able to *"...eventually sharpen other modes of perception by refocusing the listener's attention on the subtle fluctuations in timbre or pitch that accrue greater importance against an otherwise static background"* [103, p. 93]. Clearly, listening to drone music requires a very different attitude from the listener than when listening to music in which gestural shaping dominates. For drone music, one is encouraged to listen 'in' rather than 'to' the sound. Only then it becomes possible to fully appreciate the sometimes minute fluctuations in timbre and that one may start to detect the evolution of seemingly individual elements making up the composite sound. The concept of *Deep Listening*, conceived by Pauline Oliveros, feeds into this. She

[1] In the context of this research I will restrict the discussion of drone music to the electroacoustic domain. However, it should be noted that the concept of a drone (i.e. a sustained note or chord) is thought to originate from the Ancient Near East region and occupies a central role in the indigenous music of many (ancient) cultures around the world (e.g. the tanpura in indian music, medieval organum and Tuvan throat singing) [102, p. 94].

describes this as a way of *"...listening to everything all the time, and reminding yourself when you're not"* [104]. She stresses the difference between hearing and listening, in that the latter is an essentially active process; something that stimulates the mind to think more consciously about what one is actually perceiving. In order to be able to access this active mode of listening, one needs to be exposed to a sound for a long enough amount of time however, and hence, I think it is reasonable to assume that drones and longer duration textural sounds allow for this process to happen more easily than when exposed instead to sequences of gesturally shaped, short duration sounds with possibly widely differing timbres. This different attitude towards perceiving sound and music thus hints at a difference in the perception of the passage of time, as Monty Adkins points out: *"Any discussion of sonic material that is directionless, devoid of predictable change, that creates an auditory aura perceived as continuing 'present' is inherently concerned also with issues of temporality"* [105, p. 3]. In the case of linear oriented acousmatic music, gestural constructs prove important for establishing causal relationships between otherwise disjointed musical events, whereas for textural oriented music, this need for causality seems less important and hence, the perception of time can take on a more fluid form. In reply to a question about his sense of time with respect to composition, La Monte Young notes rather poetically: *"I think that this kind of sense of time has to do with getting away from the earthly sense of direction which goes from birth to death... ...and has to do with static form and moving... ...up through the sound... ...using this to create a drone state of mind as I described. By using this to create a drone state of mind, it provides a means toward achieving a state of meditation or an altered state of consciousness that can allow you to be more directly in touch with universal structure and a higher sense of order"* [106]. This idea of reaching a sort of transcendental state through the act of listening is something that appeals to me immensely. Listening to sound and music thus becomes an almost spiritual undertaking, and cultivates a deeper appreciation for the characteristic sonic features of a sound and how these evolve and transform over time.

Ambient music, which may be considered to largely focus on texture as opposed to gesture as well, treats time and timbre in a way similar to drone music. In fact, in many cases they may be considered as interchangeable terms for the same music as far as I am concerned. For instance, the work *Static Nocturne* (2010) [107] by composer Eluvium might be seen as a prime example of a hybrid drone/ambient music. Although it is oppressive, noisy and loud at times, it also has a strong melancholic, meditative quality to it. According to one of the most important pioneers of the ambient genre, Brian Eno, ambient music, just as drone music, takes the inner qualities of the sound as the starting point for the development of musical form and has as its goal the creation of an immersive listening experience: *"...immersion was really the point: we were making music to swim in, to float in, to get lost inside"* [108, p. 95]. However, whereas drone music might be quite oppressive, loud and hence would be hard to remain oblivious to, ambient music normally is characterised by a certain calmness and pleasant tonality and as such *"...must be as ignorable as it is interesting"* [108, p. 97]. As just pointed out with regards to Eluvium though, there are plenty of artists for which this distinction clearly does not uphold. Another difference can be found in the complete absence of any form of obvious pulse in drone music, whereas in certain types of ambient, pulse or some form of rhythm might be present, as is the case with many of the works by the electronic music duo Boards of Canada for instance. However, even here exceptions to the rule exist. Wolfgang Voigt's Gas moniker[2] is a good example of a music that combines rich, repetitive drone textures with rather lo-fi sounding, monotonous kick and/or sub bass patterns that marches the music forward, favouring no one particular direction, seemingly suspending it in time. Hence, although pulse is present, it appears to somehow blend in with the continuity of the drone, adding another kind of meditative and hypnotic layer to it, rather than that it functions as an obvious indicator of a linear progression of time. The *Disintegration Loops* (2002 – 2003) [110] series of works composed by William Basinski deserves special mentioning in this respect as another prime example of where pulse, in the form of a singly looped sound fragment helps the listener to focus on the minute fluctuations in timbre and the overall structure of the work and, at least for me, ingrains a deep appreciation for the intrinsic sonic qualities of the used loops and the way the playback over the tape medium makes them deteriorate slowly and subtlety over time.

5.2 The Appearance of Gesture in the Portfolio

As Smalley points out, most electroacoustic music contains both gestural as well as textural elements. Additionally, he uses the term 'gesture-framing' to illustrate settings

[2] The work *Königsforst* (1998) [109] for instance.

where gestures are containing textural interiors. Hence, the listener is conscious of both gesture and texture, although the former is more perceptually significant. Conversely, with the term ‘texture-setting’ he denotes situations where texture provides the focus from which individual gestures can emerge. I think it is fair to say that the portfolio is primarily texture oriented and therefore relates to the latter category more closely. However, there are instances where gestural shaping proves important for emphasising simulated physical interactions. Perhaps the most appreciable example in this respect are the first fourty five seconds of *Stable Equilibrium*. Clearly, gesture dominates here in the form of a structured sequence of individual events simulating the physical interaction between a resonant object and a rigid surface. However, instead of obscuring the inner timbral qualities of the resonating object[3] by additional processing or heavy editing, the result is left quite pure. As such, the gestural outcome is purely the result from the particular way the plate is being ‘played’, i.e. excited.

Another interesting thing to note, is how a transition from gesture to texture can cause a shift in the perceived amount of realism one may ascribe to the sound. For instance, in *Excite Me* an event acting as the foreground layer, starting around 6:51, suggests a physical gesture not unlike that of rubbing or shuffling, which slowly transforms into something more noisy and textural by increasing the rate at which the individual interactions persist with time. At the start of the event, gesture dominates. Its spectromorphology hints at something physical, yet the cause and source of this interaction are somewhat ambiguous, placing it in the third-order surrogacy[4] category. Gesture is transformed into texture towards the end of the event however, and since now it is no longer possible to link some physical interaction to the cause of the sound (although one may still argue that the resonant quality of the texture ensures a certain level of physical plausibility), the event has progressed into remote surrogacy.

Another moment where gesture and texture meet each other in the portfolio is in the most recently composed work *PM01*. The various percussive elements act as a mechanism to drive the music forwards in time, although due to their repetitiveness not so much with the intent to move from one clearly distinguishable goal toward the next. Rather they are used collectively to build up a groove that slowly modulates over time.

[3]The resonating object in this case refers to the response of the plate in reaction to exciting the plate visualised by figure 4.5.

[4]See [75] in order to review the concept of surrogacy as it applies to the perception of sound in electroacoustic music.

Hence, one may argue that the concept of gesture is somewhat oblique in this case, as the rhythmical elements contain sonic characteristics which may be ascribed to both gesture as well as texture. In addition, it is interesting to note that the percussive 'gestures' give rise to texture, as they are used to trigger various textural layers of longer duration which create a slowly evolving atmospheric background.

5.3 Gestural Shaping in the Frequency Domain

As pointed out earlier in this chapter, gesture in electroacoustic music is often associated with propelling the music forward in time, resulting from a causal relationship between events. As such, musical structuring and evolution is a highly directional process that predominantly draws attention to the perception of a linearly structured passage of time, whereas in the case of texture one is more likely to focus on subtle changes in the timbre of the sound and hence, the perception of time can take on a more fluid form. As Wishart notes; *"Textural perception... ...takes over... ...when the succession of events is both random and dense, so we no longer have any perceptual bearings for assigning sequential properties to the sound stream"* [93, p. 66]. However, by introducing dynamic changes in the timbral properties of a texture, one can inject a certain flow in the music which suggests a sense of movement and evolution in the sound. In [111] the author proposes the idea of using texture as the root of structure and gesture in electroacoustic music composition. However, Nyström's focus is on the emergence of *"...a distinctly directional, propulsive and gestural flow of time"* [111, p. 46], whereas I am more intrigued by the idea of using the concepts of movement and flow to modulate the timbral characteristics of the sounding materials in a semi-structured way without prioritising any directionality per se, at least not in a strict linear sense. In order to accentuate this nuance in both approaches, I will refer to my approach as gestural shaping in the frequency domain. This idea is influenced mainly by listening to the work of composer Stephan Mathieu, in particular the works[5] *Theme for Oud Amelisweerd* (2002) [112, t. 2], *The Falling Rocket* (2013) [113], and *Radioland (Panorámica)* (2012) [114]. Characteristic are the continuous, slowly evolving and fluid sound textures and an absence of melody, pulse and abrupt change. The fluidity in timbre and texture makes Mathieu's music quite

[5]The majority of Mathieu's work is provided in its entirety through `https://soundcloud.com/schwebung` (last visited: August 2015).

distinct from the drone music of, for instance, La Monte Young[6]. The music is in a constant flux, where different harmonic and non-tonal textures shift in and out of focus, ensuring that the music is not sounding static or repetitive at any point, yet without making it sound like a *"...consciously directed construction moving from a 'beginning' to an 'end' and passing from one to another"* [115, p. 178]. Musical structure thus consists of a complexly evolving continuum of sound, where serene harmonic moments alternate with more intense, noisy settings in a mutually complementary way.

The interpretation of gesture as a means for carving out the frequency domain over time plays a central role throughout the portfolio. Examples of this have in fact already been illustrated in section 4.4 by describing the unique ways in which the modal approach to PM sound synthesis has been utilised to craft the dynamically evolving, harmonically rich drones out of static noise source signals. This approach has been applied to a maximal extent for the creation of *Extase 3 (Version)* and *Extase 4 (Version)*. Whereas in other works, gesture in the traditional sense of the word has played a role to some extent (most noticeably by using it simultaneously along texture to create contrast) these two variational works treat sound and musical structure as a continuum in time completely. Both are composed out of slowly evolving textural sound layers with carefully selected timbral properties that cause a gradual shift in spectral perspective over time, while still ensuring a certain homogeneity to the overall listening experience. For *Extase 3 (Version)*, these textures are built up exclusively from harmonic sounding materials that initially were produced for *Extase 3*, but remained unused in the end. It is envisioned that the continuous structure of *Extase 3 (Version)* helps the listener in attaining a greater appreciation for the way in which individual harmonic components evolve from start to finish and how they collectively produce the impression of a homogeneous, but ever changing mass of sound. In addition to this, *Extase 4 (Version)* is using non-tonal and resonant high frequency textures to produce a time-varying contrast by constantly flowing from tonal material to noise and vice versa in order to create a carefully composed balance between pure, contemplative and more intense, assertive experiential moments. The harmonic palette of *Extase 4 (Version)*, although constantly moving, is deliberately left to be quite simple as I wanted to emphasise a perceptually wide heterogeneity in timbre using a minimum amount of tonal diversity in order to

[6]Consider La Monte Young's work *Composition 1960 #7* for instance, which just exists by the virtue of holding two single notes for a very long time, thereby maximising the idea of minimal change and stasis.

draw listening attention to the constant changes in the spectrum of the sounding materials first and foremost. Both of the works discussed in this paragraph apply the idea of gestural shaping in the frequency domain in a practical manner. I am convinced that if I had not employed this conceptual construct consciously, both works probably would have sounded similarly rich in terms of overall harmonic content, but much more static and considerably less energetically flowing. However, the continuous dynamic changes in the spectral domain are not meant to simply subdivide the music into a structured sequence of events that can be traced down a line from a definitive start to finish, but rather are meant to enhance the idea of listening to a music which is in constant flux without overly favouring any one direction in particular.

Chapter 6

Conclusion

All the works which are part of the portfolio are fixed-media works, generated entirely with the help of the computer using both physically inspired (but abstract) and PM sound synthesis methods. As such, it may be categorised as computer music, although the more general term electroacoustic music has been used predominantly throughout this thesis. Traditionally, the term computer music is associated with music that relies heavily on formal structuring processes and which may therefore produce abstract results far removed from musical intuition or real-world experience. I have strived to demonstrate that abstraction does not necessarily mean that there cannot be a certain complexity to the music or that the listener is not capable of relating to the music on a deeply personal level. On the contrary, *Extase 4 (Version)*, perhaps the most abstract work in the portfolio, is the work I personally connect to most strongly. I am convinced that it is exactly this high level of abstraction that appeals to me, as I feel that I can get caught up in the sound, be in awe of how its timbral properties evolve over time, without getting distracted by having to relate it to some source or cause situated in the real world. In addition, I have put forward the argument that musical decision making should be based on artistic considerations and personal preference rather than relying solely on some formal construct or automated process, regardless of whether the musical outcome is situated in the real world, wholly abstract or somewhere in between. Furthermore, I think that the portfolio effectively demonstrates how the use of sound transformations to transition between simulated reality and abstraction can be utilised as a compositional strategy and may serve as an effective 'something to hold onto factor' [116], as the source-cause perception of the listener is encouraged to change alongside the timbral variations in the music. PM sound synthesis has proved to be useful for this, as it makes possible unique approaches to the creation of physically plausible sound materials in different stages of abstraction from reality. Sometimes PM sound synthesis has been used to increase the level of realism which may be ascribed to the sound, but contrastingly, it has also served to increase the level of abstraction, while still ensuring a certain physical plausibility as was discussed in the context of one of the sound transformations featuring in *Extase 4*, transforming from the harmonic drone back to the realistic sound setting of waves crashing on a beach. Lastly, my intent was to demonstrate how a focus on texture, and a carefully thought out shaping of the frequency domain can result in a music which is dynamically changing, constantly evolving, but is not overly concerned with a forward motion of time in a linear sense and stressing a causal relationships between events, but which, instead, serves to develop a greater appreciation for, and awareness of, the unique timbral details of the presented sound world. Again, the particular approach to PM sound synthesis followed (namely the ability to form complex systems of inter-connected objects) proved very valuable in this respect, as each of the objects involved adds its own idiosyncratic contribution to the complexity of the possible timbres that can be produced from such a system in response to exciting it.

6.1 Relating the Portfolio Works

Throughout this thesis, several instances have occurred where certain aspects of portfolio works are related to each other. This section is meant to centralise all these observations and will attempt to summarise their overarching aims. The first work composed during the research was *Excite Me* (2012). Several artistic as well as technical ideas which were explored in this work initially, have been re-used or re-interpreted in successive works. Most noticeably in this respect are:

- the use of continuous, horizontal sound transformations to traverse between different levels of abstraction from simulated reality and to transition from gesture to texture.

- the shaping of the spectra of textures over time to expose the inner timbral details of the sounding materials dynamically and to create harmony from timbre in the process.
- the creation of several abstract and physically inspired sound synthesis models for generating a wide variety of excitation signals.
- the use of simulated environmental sounds as the basis for musical form and function.

Although the idea of incorporating environmental sound into the sound processing and music making process is by no means revolutionary [84], I envisioned that by synthesising the environmental sounds instead of using recordings, a different realm of creative opportunities could be uncovered. Inspiration for this was taken mainly from listening to recordings of water drops and rain showers and analysing by ear how these sounds were developing in both the spectral and temporal domain. The design of some of the earliest excitation models was an attempt to create virtual impressions of these type of sounds, not so much with the aim to approach the original recordings as accurately as possible, but rather to capture and exaggerate some of the recognisable timbral and temporal features of these sounds. This then evolved into playing with the idea of transforming these dynamically changing natural and physically plausible sounding textures into harmonic sounding materials using PM sound synthesis techniques. After completion of the piece it was concluded that the compositional ideas and the creative results derived therefrom offered a strong basis for exploring each of them more in depth in subsequent works.

Stable Equilibrium (2013) is the second portfolio work composed and is primarily concerned with a further exploration of the use of continuous sound transformations for creating a listening experience that transfers the listener gradually from simulated reality to abstraction. Additional excitation models where designed in order to simulate convincing sounding collision interactions and sustained sounds resembling to some extent that of a bowed string. As mentioned in section 4.4 already, an inverse approach to the one used for *Excite Me* was investigated in order to produce a continuous, but dynamically varying drone layer, using band pass filters instead of band reject filters to carve out dynamically varying patterns in the spectrum of the excitation signal. Both the approach followed for *Excite Me* as well as for *Stable Equilibrium* may be seen as a different interpretation of Risset's 'spectral analysis of a chord' concept. Furthermore, the idea of using sound transformations as a means to link different sections together and more interestingly; to play with the source-cause perception of the presented material by the listener over time has been explored using different, physically inspired sound models.

Extase 1 (2013) tries to combine certain compositional ideas that were explored in the previous works. It is characterised by layers of different events of a relatively longer duration as a means to generate spectrally rich textures which are constantly evolving. However, whereas the harmonic development in *Excite Me* and *Stable Equilibrium* was residing in a fixed domain of harmonic possibilities, *Extase 1* tries to explore a more diverse harmonic development throughout the whole work by utilising the modal data associated with the different objects that are part of the designed virtual instrument as a spectrally informed musical scale. All pitch decisions and harmonic constructions are thus directly informed by the spectral characteristics of the sounding materials, creating a functional relationship between sound timbre and the harmonic structure of the music. The idea of gesture in the frequency domain is present as well, although at the time of composing, I was not yet consciously aware of thinking about it in this way. Texture is clearly dominating in *Extase 1*, even in the last section, made up of struck sounds which are slowly decaying over time. There are clearly recognisable events emanating from this texture at times that accentuate harmonic turning points and spectral changes in the music, without drawing attention to themselves too much.

Each portfolio is likely to contain a central work; a work that demonstrates an extension or refinement of previously explored ideas and as such acts as a functional representation of all creative, conceptual and technical efforts made by a composer throughout an extensive period of research and dedication. For this portfolio, *Extase 2, 3 & 4* acts as this work. It is the longest and most timbrally diverse of all portfolio works. After completion, the work seemed to divide naturally into three different sections that transform into each other smoothly. Each section focusses on exploring the timbral properties of a different object making up the complete system of inter-connected objects. As each object not only determines the timbre of the sounding materials, but also is used for making harmonic decisions, each section is clearly recognisable by its unique harmonic footprint. Carefully designed sound transformations on excitation signals serve as practical means to link the different sections together in a homogeneous way and to produce dynamic variations in timbre and harmony in each individual section. Just as for *Excite Me*, simulated environmental sounds are used as musical materials on their own and as dynamically varying excitation signals for resonator modules. Furthermore, a refined model for simulating bowed-like sounds was introduced for *Extase 3* alongside the simpler model used previously for generating all the sound material for *Extase 1*.

The biggest difference between *Extase 3 (Version)* and *Extase 4 (Version)* and previously composed works is that the focus is entirely on the creation of one continuous sound mass made up of individual textures having carefully selected timbral properties. The way in which these individual textures evolve over time and at the same time collectively combine into one homogeneous whole is what causes the music to evolve continuously over time, without overemphasising a forward motion to the music in a strict linear sense. *Extase 3 (Version)* concentrates on the idea of interpreting gesture as a means for carving out trajectories in the frequency domain in order to create movement and flow in the timbre of the sounding materials. *Extase 4 (Version)* takes this a step further by also incorporating a continuous, but gradual transition between non-tonal and tonal elements. Both of these works are prime examples of musical situations where a focus on texture and continuity not inevitably means that the end result has to be perceived as static or stationary, but instead injects a sense of flow into the musical experience without directing it forward per se.

PM01 is the most recent work composed. It clearly differs from the other portfolio works in the sense that there is a strong rhythmical element present in the form of repetitive percussive patterns that slowly change and interact with each other over time. However, there is also a certain similarity with the other portfolio works in that structurally, it is build up out of sound layers which timbral properties evolve relatively slowly over time. My aim with this work was to merge in some sense my background in techno music production with more experimental approaches to sound generation and musical structuring and to use the developed software to produce a different realm of sounds compared to the other portfolio works.

6.2 Future Developments

In general I am very happy with how my research turned out and the practical results produced in the process. As it stands now, the portfolio seems to make sense as a musical whole from start to finish and demonstrates clearly the different compositional ideas that are discussed in this thesis. One thing in particular I would like to explore further when I will be given the chance and opportunity, involves a more detailed exposition of the idea of shaping the timbral properties of a sound continuum in order to create movement and timbral diversity over an extended period of time. Hence, I would like to create a continuous work of at least forty, but ideally sixty minutes that tries to explore this more in depth using a combination of abstract sound synthesis methods and the PM sound synthesis software developed during this research period, ideally combining this together with visuals synthesised from the audio data to create an extended, immersive audio-visual experience. In addition, I would like to extend my efforts in researching the creative potential of PM sound synthesis through composition by investigating how dynamically changing material parameters can be utilised creatively and by looking into known methods for a more accurate modelling of nonlinear behaviour, in particular that of distributed non-linearities in plates and non-linear inter-action between (multiple) distributed objects.

Appendix A

Mathematical Formulation of PMLib

A.1 Preliminaries

This section provides the necessary background to understand the basic operations involved in the formulation of finite difference schemes corresponding to the PDEs of the 1D and 2D objects that form the basic building blocks of the developed PM sound synthesis library *PMLib*. To this end, it offers a summary of chapters 5 and 10 from [24] and effectively borrows a lot from the same mathematical framework developed in it, as well as from [1]. Hence, if the reader finds that the summary provided here is not sufficient to fully grasp the material discussed later on, he/she is advised to consult these references instead. With implementation considerations in mind, certain features are described in a little more detail in comparison to the aforementioned references, in particular when it comes to the explicit form of the used difference matrices.

A.1.1 Grid Functions and Difference Operators in 1D and 2D

A 1D continuous function $u(x, t)$ defined over the finite spatial domain $\mathcal{D} = \{x \mid 0 \leq x \leq L\}$, e.g. the solution of the partial differential equation 2.1, may be approximated by a grid function u_l^n defined at discrete spatial location $x = lh$ and at discrete time instant $t = nk$, where h is the grid spacing and k is the time step and $l, n = 0, 1, 2, \ldots$. When the domain of the continuous function $u(x,t)$ is specified as $\mathcal{D} = \mathbb{U} = \{x \mid 0 \leq x \leq 1\}$ – which may always accomplished by employing scaling techniques – the grid spacing h may be chosen as $h = 1/N$, so that the finite domain of the grid function u_l^n becomes $\mathcal{D} = \mathbb{U}_N = \{l \mid 0 \leq l \leq N\}$. For the moment we will assume the grid functions are of infinite extent in order to simplify the discussion in the next section. We will return to the unit domain once explicitly defined boundary conditions make their appearance. Now that we have defined the 1D grid function u_l^n, discrete approximations to various continuous time and space derivative operators may be defined as

$$\delta_{tt} u_l^n = \frac{1}{k^2}\left(u_l^{n+1} - 2u_l^n + u_l^{n-1}\right) \tag{A.1a}$$

$$\delta_{t\cdot} u_l^n = \frac{1}{2k}\left(u_l^{n+1} - u_l^{n-1}\right) \tag{A.1b}$$

$$\delta_{t-} u_l^n = \frac{1}{k}\left(u_l^{n} - u_l^{n-1}\right) \tag{A.1c}$$

$$\delta_{t+} u_l^n = \frac{1}{k}\left(u_l^{n+1} - u_l^{n}\right) \tag{A.1d}$$

$$\delta_{xx} u_l^n = \frac{1}{h^2}\left(u_{l+1}^{n} - 2u_l^n + u_{l-1}^{n}\right) \tag{A.1e}$$

$$\delta_{xxxx} u_l^n = \delta_{xx}\delta_{xx} u_l^n = \frac{1}{h^4}\left(u_{l+2}^{n} - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^{n} + u_{l-2}^{n}\right). \tag{A.1f}$$

Similary, a 2D continuous function $u(x,y,t)$ defined over the unit area rectangular domain $\mathbb{U}_\epsilon = \{(x,y) \mid 0 \leq x \leq 1, 0 \leq y \leq 1\}$ of dimensions $\sqrt{\epsilon} \times \sqrt{1/\epsilon}$, where $\epsilon = L_x/L_y$, may be approximated by a grid function $u_{l,m}^n$. Assuming equal grid spacings in both the x and y directions – i.e. $h_x = h_y = h$ – the grid function is defined at discrete spatial locations $x = lh$, $0 \leq l \leq N_x$ and $y = mh$, $0 \leq m \leq N_y$. Just as for the 1D grid function, we will assume the 2D grid function $u_{l,m}^n$ to be of infinite extent for the moment as well. Discrete approximations to the Laplacian and the biharmonic operator may then be defined as

$$\delta_\Delta u_{l,m}^n = \delta_{xx} u_{l,m}^n + \delta_{yy} u_{l,m}^n \tag{A.2a}$$

$$\delta_{\Delta,\Delta} u_{l,m}^n = \delta_\Delta\delta_\Delta u_{l,m}^n = \delta_{xxxx} u_{l,m}^n + 2\delta_{xx}\delta_{yy} u_{l,m}^n + \delta_{yyyy} u_{l,m}^n. \tag{A.2b}$$

A.1.2 Grid Functions and Difference Operators in Matrix Form

With an eventual modal implementation in mind, we need to be able to express grid functions and difference operators in matrix form. Assuming a 1D grid function defined over the infinite domain $\mathcal{D} = \mathbb{Z} = \{l \mid -\infty < l < \infty\}$, this may be expressed as a column vector of infinite size $\mathbf{u}^n = [\ldots, u^n_{-1}, u^n_0, u^n_1, \ldots]^\top$. Along similar lines, a 2D grid function defined over the infinite domain $\mathcal{D} = \mathbb{Z}^2 = \{l, m \mid -\infty < l, m < \infty\}$ may be arranged into a column vector of infinite size by concatenating all the columns corresponding to increasing index m to get $\mathbf{u}^n = [\ldots, \mathbf{u}^n_{-1}, \mathbf{u}^n_0, \mathbf{u}^n_1, \ldots]^\top$, where $\mathbf{u}^n_l = [\ldots, u^n_{l,-1}, u^n_{l,0}, u^n_{l,1}, \ldots]^\top$. Hence this will allow us to put the spatial difference operators introduced in the previous section in matrix form. For 1D problems the operators δ_{xx} and δ_{xxxx} may be written as the infinite matrices $h^{-2}\mathbf{D}^{(1)}_{xx}$ and $h^{-4}\mathbf{D}^{(1)}_{xxxx}$ respectively:

$$\mathbf{D}^{(1)}_{xx} = \begin{bmatrix} \ddots & \ddots & & & & 0 \\ \ddots & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & \ddots \\ 0 & & & & \ddots & \ddots \end{bmatrix} \quad \mathbf{D}^{(1)}_{xxxx} = \begin{bmatrix} \ddots & \ddots & \ddots & & & & 0 \\ \ddots & 6 & -4 & 1 & & & \\ \ddots & -4 & 6 & -4 & 1 & & \\ & 1 & -4 & 6 & -4 & 1 & \\ & & 1 & -4 & 6 & -4 & \ddots \\ & & & 1 & -4 & 6 & \ddots \\ 0 & & & & \ddots & \ddots & \ddots \end{bmatrix}. \tag{A.3}$$

For 2D problems the operator δ_{xx} and δ_{yy} may be written as the infinite block matrices $h^{-2}\mathbf{D}^{(2)}_{xx}$ and $h^{-2}\mathbf{D}^{(2)}_{yy}$ given by

$$\mathbf{D}^{(2)}_{xx} = \begin{bmatrix} \ddots & \ddots & & & & \mathbf{0} \\ \ddots & -2\mathbf{I} & \mathbf{I} & & & \\ & \mathbf{I} & -2\mathbf{I} & \mathbf{I} & & \\ & & \mathbf{I} & -2\mathbf{I} & \mathbf{I} & \\ & & & \mathbf{I} & -2\mathbf{I} & \ddots \\ \mathbf{0} & & & & \ddots & \ddots \end{bmatrix} \quad \mathbf{D}^{(2)}_{yy} = \begin{bmatrix} \ddots & & & & \mathbf{0} \\ & \mathbf{D}^{(1)}_{yy} & & & \\ & & \mathbf{D}^{(1)}_{yy} & & \\ & & & \mathbf{D}^{(1)}_{yy} & \\ \mathbf{0} & & & & \ddots \end{bmatrix}, \tag{A.4}$$

where $\mathbf{D}^{(2)}_{yy}$ is of the same form as $\mathbf{D}^{(1)}_{xx}$. The non-zero diagonals of the infinite block matrices representing the Laplacian operator δ_Δ and the biharmonic operator $\delta_{\Delta,\Delta}$ are then given by

$$
\mathbf{D}_{\Delta} = \left[\begin{array}{c|c}
\begin{matrix} \ddots & \ddots & & & & \\ \ddots & -4 & 1 & & & \\ & 1 & -4 & 1 & & \\ & & 1 & -4 & 1 & \\ & & & 1 & -4 & \ddots \\ & & & & \ddots & \ddots \end{matrix} &
\begin{matrix} \ddots & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{matrix} \\ \hline
\begin{matrix} \ddots & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{matrix} &
\begin{matrix} \ddots & \ddots & & & & \\ \ddots & -4 & 1 & & & \\ & 1 & -4 & 1 & & \\ & & 1 & -4 & 1 & \\ & & & 1 & -4 & \ddots \\ & & & & \ddots & \ddots \end{matrix}
\end{array}\right] \tag{A.5a}
$$

$$
\mathbf{D}_{\Delta,\Delta} = \left[\begin{array}{c|c|c}
\begin{matrix} \ddots & \ddots & \ddots & & & & \\ \ddots & 20 & -8 & 1 & & & \\ \ddots & -8 & 20 & -8 & 1 & & \\ & 1 & -8 & 20 & -8 & 1 & \\ & & 1 & -8 & 20 & -8 & \ddots \\ & & & 1 & -8 & 20 & \ddots \\ & & & & \ddots & \ddots & \ddots \end{matrix} &
\begin{matrix} \ddots & \ddots & & & & \\ \ddots & -8 & 2 & & & \\ & 2 & -8 & 2 & & \\ & & 2 & -8 & 2 & \\ & & & 2 & -8 & \ddots \\ & & & & \ddots & \ddots \end{matrix} &
\begin{matrix} \ddots & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{matrix} \\ \hline
\begin{matrix} \ddots & \ddots & & & & \\ \ddots & -8 & 2 & & & \\ & 2 & -8 & 2 & & \\ & & 2 & -8 & 2 & \\ & & & 2 & -8 & \ddots \\ & & & & \ddots & \ddots \end{matrix} &
\begin{matrix} \ddots & \ddots & \ddots & & & & \\ \ddots & 20 & -8 & 1 & & & \\ \ddots & -8 & 20 & -8 & 1 & & \\ & 1 & -8 & 20 & -8 & 1 & \\ & & 1 & -8 & 20 & -8 & \ddots \\ & & & 1 & -8 & 20 & \ddots \\ & & & & \ddots & \ddots & \ddots \end{matrix} &
\begin{matrix} \ddots & \ddots & & & & \\ \ddots & -8 & 2 & & & \\ & 2 & -8 & 2 & & \\ & & 2 & -8 & 2 & \\ & & & 2 & -8 & \ddots \\ & & & & \ddots & \ddots \end{matrix} \\ \hline
\begin{matrix} \ddots & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & \ddots \end{matrix} &
\begin{matrix} \ddots & \ddots & & & & \\ \ddots & -8 & 2 & & & \\ & 2 & -8 & 2 & & \\ & & 2 & -8 & 2 & \\ & & & 2 & -8 & \ddots \\ & & & & \ddots & \ddots \end{matrix} &
\begin{matrix} \ddots & \ddots & \ddots & & & & \\ \ddots & 20 & -8 & 1 & & & \\ \ddots & -8 & 20 & -8 & 1 & & \\ & 1 & -8 & 20 & -8 & 1 & \\ & & 1 & -8 & 20 & -8 & \ddots \\ & & & 1 & -8 & 20 & \ddots \\ & & & & \ddots & \ddots & \ddots \end{matrix}
\end{array}\right]. \tag{A.5b}
$$

A.1.3 Inner and Outer Products of Column Vectors

The inner product between two finite column vectors $\mathbf{u}$ and $\mathbf{v}$ of the same length N may be defined as

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{v}^\top \mathbf{u} = \sum_{i=0}^{N} u_i v_i \qquad \langle \mathbf{u}, \mathbf{u} \rangle = \|\mathbf{u}\|^2 \tag{A.6}$$

which results in a scalar quantity. The outer product of two finite column vectors $\mathbf{u}$ and $\mathbf{v}$ of length N_1 and N_2 respectively may be defined as

$$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^\top \tag{A.7}$$

which results in a $N_1 \times N_2$ matrix $\mathbf{W}$ where the entrees of $\mathbf{W}$ are given by $w_{i,j} = u_i v_j$.

A.1.4 Input and Output

Assuming that any excitation forces and forces arising due to the inter-connection of distributed objects are expressed as a scalar quantity – i.e. the forces are acting on a single point along the surface of an object only – one may define the column vector

$$\mathbf{e} = [0, \ldots, 0, 1, 0, \ldots, 0]^\top, \tag{A.8}$$

which may be used to transform the scalar quantity into a vector quantity. The location of the single non-zero value in the 1D case is defined by $\lfloor x_i/h \rfloor$ and in the 2D case by $P_y \lfloor x_i/h \rfloor + \lfloor y_i/h \rfloor$, where P_y denotes the actual number of grid points in the y direction. Note that we use the variable P_y here rather than N_y as the actual number of grid points depends on the boundary conditions. For instance, under fixed or simply supported boundary conditions one may omit the first and last grid function values as these will always be zero. Hence, $P_y = N_y - 1$ in this case, whereas in the case of free boundary conditions the first and last end points are non-zero in general and thus cannot be omitted. Therefore, in the case of 2D free boundary conditions $P_y = N_y + 1$. As free boundary conditions are only implemented for 1D objects for now we can safely assume that in our case $P_y = N_y - 1$ always. Obtaining output at a single location may similarly be expressed as an inner product between two vectors, for which we may use $\mathbf{e}^\top$.

A.1.5 Spatial Domains

Until now we assumed that the spatial domains were of infinite extent. A more realistic setting is to define a bounded region. In the 1D case this may be chosen as the finite domain $\mathcal{D} = \mathbb{U}_N = \{l \mid 0 \leq l \leq N\}$, which will correspond to the unit interval if we choose $N = 1/h$ and employ appropriate scaling techniques. For 2D problems we may define the unit area rectangular domain $\mathcal{D} = \mathbb{U}^2_\epsilon = \{l, m \mid 0 \leq l \leq N_x, 0 \leq m \leq N_y\}$ if we choose $N_x = \lfloor\sqrt{\epsilon}/h\rfloor$ and $N_y = \lfloor 1/\left(\sqrt{\epsilon}h\right)\rfloor$. Here $\epsilon = L_x/L_y$ denotes the aspect ratio of the rectangle and as stated previously, it is assumed that the grid spacing in the horizontal and vertical directions is the same, i.e. $h_x = h_y = h$. Defining a bounded region also means boundary conditions come into play and hence these need to be accounted for explicitly in the difference matrices defined in section A.1.2. This will be discussed in more detail in section A.2.2.

A.2 Finite Difference Schemes for Strings, Bars, Membranes and Plates

An explicit finite difference scheme for a stiff, linearly vibrating string including loss is given by the PDE

$$\delta_{tt}u = \gamma^2\delta_{xx}u - \kappa^2\delta_{xxxx}u - 2b_1\delta_{t\cdot}u + 2b_2\delta_{t-}\delta_{xx}u + \frac{1}{h}\sum_q \mathbf{e}^{(q)}F^{(q)}, \tag{A.9}$$

where u^n_l is a spatially scaled[1] 1D grid function representing transverse string displacement at time instant $t = nk$ and the scaled location $x = lh$ for integers $n = 0, 1, 2, \ldots$ and $l = 0, 1, \ldots, N-1, N$, the parameter $\gamma = c/L$ in absence of the stiffness and loss terms is the spatially scaled wave speed in units of Hz, $\kappa = \sqrt{EI/\left(\mu L^4\right)}$ is a spatially scaled stiffness coefficient with E being Young's modulus, I being the moment of inertia and μ being the linear mass density of the string. The terms b_1 and b_2 give rise to frequency-independent and dependent loss respectively. The last term on the right hand side denotes the sum of all point-wise applied forces in units of Newton divided by the linear mass density of the string. For the moment we will assume that these forces can

[1]A spatial scaling of the original continuous function $u(x,t)$ may be obtained by introducing the dimensionless coordinate $x' = x/L$, where L is the length of the string, and differentiating with respect to x' instead. Note that in (A.9) all primes have been removed from the difference operator arguments for clarity.

either be externally applied excitations or coupling forces between two objects due to their inter-connection. Note that in absence of the first term on the right hand side of equation (A.9), i.e. when $\gamma = 0$, the scheme reduces to that of a linearly vibrating thin bar with loss. Along similar lines, an explicit finite difference scheme for a stiff, linearly vibrating membrane is given by

$$\delta_{tt}u = \gamma^2\delta_\Delta u - \kappa^2\delta_{\Delta,\Delta}u - 2b_1\delta_{t\cdot}u + 2b_2\delta_{t-}\delta_\Delta u + \frac{1}{h^2}\sum_q \mathbf{e}^{(q)}F^{(q)}, \tag{A.10}$$

where $u^n_{l,m}$ is a 2D grid function representing transverse membrane displacement at time instant $t = nk$ and the scaled location $x = lh$, $y = mh$ for integers $n = 0, 1, 2, \ldots$, $l = 0, 1, \ldots, N_x - 1, N_x$ and $m = 0, 1, \ldots, N_y - 1, N_y$, just as for the string the parameter $\gamma = c/L$ in absence of the stiffness and loss terms is the spatially scaled wave speed, $\kappa = \sqrt{EH^2/\left[12\left(1-\nu^2\right)\rho L^4\right]}$ is a spatially scaled stiffness coefficient with H being the membrane thickness and $\nu < 1/2$ being Poisson's ratio. Setting $\gamma = 0$ reduces the scheme to that for a linearly vibrating thin plate with loss.

A.2.1 Stability and Boundary Conditions

In absence of any external or connection forces, the scheme given by equation (A.9) is stable when

$$h \geq \sqrt{\frac{1}{2}\left(\gamma^2k^2 + 4b_2k + \sqrt{\left(\gamma^2k^2 + 4b_2k\right)^2 + 16\kappa^2k^2}\right)}. \tag{A.11}$$

Stability for the scheme given by equation (A.10) in absence of any external or connection forces is guaranteed whenever

$$h \geq \sqrt{\gamma^2k^2 + 4b_2k + \sqrt{\left(\gamma^2k^2 + 4b_2k\right)^2 + 16\kappa^2k^2}}. \tag{A.12}$$

Both stability bounds follow from employing energy or von Neumann analysis methods. Energy analysis may also be used to arrive at numerical boundary conditions. For the scheme of the string (A.9) these are given by

$$\begin{aligned} u_0 &= \delta_{x-}u_0 = 0 \\ u_N &= \delta_{x+}u_N = 0 \end{aligned} \qquad \text{clamped,} \tag{A.13}$$

$$\left.\begin{aligned} u_0 &= \delta_{xx} u_0 = 0 \\ u_N &= \delta_{xx} u_N = 0 \end{aligned}\right. \quad \text{simply supported} \tag{A.14}$$

and

$$\left.\begin{aligned} \delta_{xx} u_0 &= \kappa^2 \delta_{xx} \delta_{x-} u_0 - \gamma^2 \delta_{x-} u_0 - 2b_2 \delta_{t-} \delta_{x-} u_0 &&= 0 \\ \delta_{xx} u_N &= \kappa^2 \delta_{xx} \delta_{x+} u_N - \gamma^2 \delta_{x+} u_N - 2b_2 \delta_{t-} \delta_{x+} u_N &&= 0 \end{aligned}\right. \quad \text{free.} \tag{A.15}$$

This makes for a total of 9 different possible choices per string or bar. For the membrane defined over the unit rectangular domain $\mathbb{U}_\epsilon^2$ clamped and simply supported boundary conditions take the form

$$\left.\begin{aligned} u_{0,m} &= \delta_{x-} u_{0,m} &&= 0 \qquad m \in \mathbb{U}_\epsilon^2 \\ u_{N_x,m} &= \delta_{x+} u_{N_x,m} &&= 0 \qquad m \in \mathbb{U}_\epsilon^2 \\ u_{l,0} &= \delta_{y-} u_{l,0} &&= 0 \qquad l \in \mathbb{U}_\epsilon^2 \\ u_{l,N_y} &= \delta_{y+} u_{l,N_y} &&= 0 \qquad l \in \mathbb{U}_\epsilon^2 \end{aligned}\right. \quad \text{clamped} \tag{A.16}$$

$$\left.\begin{aligned} u_{0,m} &= \delta_{xx} u_{0,m} &&= 0 \qquad m \in \mathbb{U}_\epsilon^2 \\ u_{N_x,m} &= \delta_{xx} u_{N_x,m} &&= 0 \qquad m \in \mathbb{U}_\epsilon^2 \\ u_{l,0} &= \delta_{yy} u_{l,0} &&= 0 \qquad l \in \mathbb{U}_\epsilon^2 \\ u_{l,N_y} &= \delta_{yy} u_{l,N_y} &&= 0 \qquad l \in \mathbb{U}_\epsilon^2 \end{aligned}\right. \quad \text{simply supported} \tag{A.17}$$

Free boundary conditions are more complicated and depend on an extra parameter ν. After a rather lengthy analysis these may be found to be

$$\begin{aligned} \left(\delta_{xx} + \nu\delta_{yy}\right) u_{0,m} &= \kappa^2 \delta_{x-} \left(\delta_{xx} + (2-\nu)\,\delta_{yy}\right) u_{0,m} - \gamma^2 \delta_{x-} u_{0,m} - 2b_2 \delta_{x-} \delta_{t-} u_{0,m} &&= 0 \\ \left(\delta_{xx} + \nu\delta_{yy}\right) u_{N_x,m} &= \kappa^2 \delta_{x+} \left(\delta_{xx} + (2-\nu)\,\delta_{yy}\right) u_{N_x,m} - \gamma^2 \delta_{x+} u_{N_x,m} - 2b_2 \delta_{x+} \delta_{t-} u_{N_x,m} &&= 0 \\ \left(\nu\delta_{xx} + \delta_{yy}\right) u_{l,0} &= \kappa^2 \delta_{y-} \left((2-\nu)\,\delta_{xx} + \delta_{yy}\right) u_{l,0} - \gamma^2 \delta_{y-} u_{l,0} - 2b_2 \delta_{y-} \delta_{t-} u_{l,0} &&= 0 \\ \left(\nu\delta_{xx} + \delta_{yy}\right) u_{l,N_y} &= \kappa^2 \delta_{y+} \left((2-\nu)\,\delta_{xx} + \delta_{yy}\right) u_{l,N_y} - \gamma^2 \delta_{y+} u_{l,N_y} - 2b_2 \delta_{y+} \delta_{t-} u_{l,N_y} &&= 0 \end{aligned} \tag{A.18}$$

for $m, l \in \mathbb{U}_\epsilon^2$ and the four additional corner conditions

$$\left.\begin{aligned} \delta_{x-} \delta_{y-} u_{0,0} &= 0 \\ \delta_{x-} \delta_{y+} u_{0,N_y} &= 0 \\ \delta_{x+} \delta_{y+} u_{N_x,N_y} &= 0 \\ \delta_{x+} \delta_{y-} u_{N_x,0} &= 0 \end{aligned}\right. \quad \text{corner,} \tag{A.19}$$

which makes for a total of 81 possible boundary conditions.

A.2.2 Difference Operators in Matrix Form Revisited

Now that valid numerical boundary conditions have been formulated we may revisit the matrix form of finite difference operators introduced in Section A.1.2 and define them more explicitly by specifying the effect the boundary conditions have on their internal structure. For 1D problems the difference matrices $\mathbf{D}_{xx}^{(1)}$ and $\mathbf{D}_{xxxx}^{(1)}$ under clamped and simply supported boundary conditions at both ends take the form

$$\mathbf{D}_{xx}^{(1)}\mathbf{u}^n = \underbrace{\begin{bmatrix} -2 & 1 & & & 0 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & 1 & -2 \end{bmatrix}}_{\text{clamped or simply supported}} \begin{bmatrix} u_1^n \\ u_2^n \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \end{bmatrix}, \tag{A.20}$$

$$\mathbf{D}_{xxxx}^{(1)}\mathbf{u}^n = \underbrace{\begin{bmatrix} 6 & -4 & 1 & & & & 0 \\ -4 & 6 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 6 & -4 \\ 0 & & & & 1 & -4 & 6 \end{bmatrix}}_{\text{clamped}} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ u_{N-3}^n \\ u_{N-2}^n \\ u_{N-1}^n \end{bmatrix} \tag{A.21a}$$

$$\mathbf{D}_{xxxx}^{(1)}\mathbf{u}^n = \underbrace{\begin{bmatrix} 5 & -4 & 1 & & & & 0 \\ -4 & 6 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 6 & -4 \\ 0 & & & & 1 & -4 & 5 \end{bmatrix}}_{\text{simply supported}} \begin{bmatrix} u_1^n \\ u_2^n \\ u_3^n \\ \vdots \\ u_{N-3}^n \\ u_{N-2}^n \\ u_{N-1}^n \end{bmatrix} \tag{A.21b}$$

Arriving at a matrix form corresponding to free boundary conditions involves a little more work however, as now it is necessary to substitute for values of the grid function lying outside of the unit domain in terms of known values. Assuming a free boundary condition at grid point $l = 0$ we may simply use the first condition to find that

$$u_{-1}^n = 2u_0^n - u_1^n$$

Using this together with the second condition allows us to express the value of the grid function at the point $l = -2$ at time instant n as

$$u_{-2}^n = \left(3 + \frac{\gamma^2}{\kappa^2} + \frac{2b_2h^2}{\kappa^2 k}\right) u_0^n - \left(2 + \frac{\gamma^2 h^2}{\kappa^2} + \frac{2b_2h^2}{\kappa^2 k}\right) u_1^n + \frac{2b_2h^2}{\kappa^2 k}\left(u_1^{n-1} - u_0^{n-1}\right).$$

Using this knowledge we may explicitly define the operator acting on the grid function u_l^n at grid points $l = 0$ and $l = 1$ as

$$\begin{aligned}
h^4\delta_{xxxx}u_0^n &= u_2^n - \left(2 + \frac{\gamma^2 h^2}{\kappa^2} + \frac{2b_2h^2}{\kappa^2 k}\right) u_1^n + \left(1 + \frac{\gamma^2 h^2}{\kappa^2} + \frac{2b_2h^2}{\kappa^2 k}\right) u_0^n \\
&\quad + \frac{2b_2h^2}{\kappa^2 k}\left(u_1^{n-1} - u_0^{n-1}\right) \\
h^4\delta_{xxxx}u_1^n &= u_3^n - 2u_2^n + 5u_1^n - 2u_0^n.
\end{aligned}$$

Hence, the difference matrices $\mathbf{D}_{xx}^{(1)}$ and $\mathbf{D}_{xxxx}^{(1)}$ under free boundary conditions at both ends are of the form

$$\mathbf{D}_{xx}^{(1)}\mathbf{u}^n = \underbrace{\begin{bmatrix} 0 & 0 & & & 0 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & 0 & 0 \end{bmatrix}}_{\text{free}} \begin{bmatrix} u_0^n \\ u_1^n \\ \vdots \\ u_{N-1}^n \\ u_N^n \end{bmatrix} \tag{A.22}$$

$$
\mathbf{D}^{(1)}_{xxxx}\mathbf{u}^n = \underbrace{\begin{bmatrix} a_0 & a_1 & 1 & & & & 0 \\ -2 & 5 & -4 & 1 & & & \\ 1 & -4 & 6 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 6 & -4 & 1 \\ & & & 1 & -4 & 5 & -2 \\ 0 & & & & 1 & a_1 & a_0 \end{bmatrix}}_{\text{free}} \begin{bmatrix} u_0^n \\ u_1^n \\ u_2^n \\ \vdots \\ u_{N-2}^n \\ u_{N-1}^n \\ u_N^n \end{bmatrix} + \underbrace{\begin{bmatrix} a_2 & -a_2 & & & & 0 \\ & 0 & 0 & & & \\ & & \ddots & \ddots & & \\ & & & 0 & 0 & \\ 0 & & & & -a_2 & a_2 \end{bmatrix}}_{\text{free}} \begin{bmatrix} u_0^{n-1} \\ u_1^{n-1} \\ \vdots \\ u_{N-1}^{n-1} \\ u_N^{n-1} \end{bmatrix}, \tag{A.23}
$$

where the coefficients a_0, a_1 and a_2 are given by

$$
a_0 = 1 + \frac{\gamma^2 h^2}{\kappa^2} + \frac{2b_2 h^2}{\kappa^2 k} \qquad a_1 = -\left(2 + \frac{\gamma^2 h^2}{\kappa^2} + \frac{2b_2 h^2}{\kappa^2 k}\right) \qquad a_2 = -\frac{2b_2 h^2}{\kappa^2 k}.
$$

Note that the matrix form corresponding to $h^4\delta_{xxxx}$ now actually consists of two matrices. The first one is acting on $\mathbf{u}$ at time instant n and the second one on $\mathbf{u}$ at time instant $n-1$. As at the moment only clamped and simply supported boundary conditions are implemented for 2D schemes corresponding to a membrane or plate, we will only discuss the 2D difference matrices which include these type of boundary conditions. As may be deduced from the 1D case, implementing free boundary conditions for the 2D case – although not particularly difficult – involves quite a bit of work[2] and therefore it was decided to not prioritise this. Assuming the same boundary conditions at all four edges, the difference block matrices $\mathbf{D}^{(2)}_{xx}$ and $\mathbf{D}^{(2)}_{yy}$ under clamped or simply supported boundary conditions look like

$$
\mathbf{D}^{(2)}_{xx}\mathbf{u}^n = \underbrace{\begin{bmatrix} -2\mathbf{I} & \mathbf{I} & & & \mathbf{0} \\ \mathbf{I} & -2\mathbf{I} & \mathbf{I} & & \\ & \ddots & \ddots & \ddots & \\ & & \mathbf{I} & -2\mathbf{I} & \mathbf{I} \\ \mathbf{0} & & & \mathbf{I} & -2\mathbf{I} \end{bmatrix}}_{\text{clamped or simply supported}} \begin{bmatrix} \mathbf{u}_1^n \\ \mathbf{u}_2^n \\ \vdots \\ \mathbf{u}_{N_x-2}^n \\ \mathbf{u}_{N_x-1}^n \end{bmatrix} \tag{A.24a}
$$

$$
\mathbf{D}^{(2)}_{yy}\mathbf{u}^n = \underbrace{\begin{bmatrix} \mathbf{D}^{(1)}_{yy} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{D}^{(1)}_{yy} \end{bmatrix}}_{\text{clamped or simply supported}} \begin{bmatrix} \mathbf{u}_1^n \\ \vdots \\ \mathbf{u}_{N_x-1}^n \end{bmatrix}. \tag{A.24b}
$$

Here the matrix $\mathbf{D}^{(1)}_{yy}$ is of exactly the same form as the one appearing in (A.20). Hence the explicit form of the difference matrix $\mathbf{D}_\Delta = \mathbf{D}^{(2)}_{xx} + \mathbf{D}^{(2)}_{yy}$ under clamped or simply

[2]See p. 356, Problem 12.6 of [24] for more on this

supported boundary conditions corresponding to the Laplacian operator $h^2\delta_\Delta$ is given by

$$\mathbf{D}_\Delta \mathbf{u}^n = \underbrace{\left[\begin{array}{c|c} \begin{matrix} -4 & 1 & & & 0 \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ 0 & & & 1 & -4 \end{matrix} & \begin{matrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{matrix} \\ \hline \begin{matrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{matrix} & \begin{matrix} -4 & 1 & & & 0 \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ 0 & & & 1 & -4 \end{matrix} \end{array}\right]}_{\text{clamped or simply supported}} \left[\begin{array}{c} u^n_{1,1} \\ u^n_{1,2} \\ \vdots \\ u_{1,Ny-2} \\ u_{1,Ny-1} \\ \hline u_{2,1} \\ \vdots \\ u_{2,Ny-1} \\ \hline u_{Nx-2,1} \\ \vdots \\ u_{Nx-2,Ny-1} \\ \hline u_{Nx-1,1} \\ u_{Nx-1,2} \\ \vdots \\ u_{Nx-1,Ny-2} \\ u_{Nx-1,Ny-1} \end{array}\right]. \tag{A.25}$$

The 2D difference matrix $\mathbf{D}_{\Delta,\Delta} = \mathbf{D}^{(2)}_{xxxx} + \mathbf{D}^{(2)}_{yyyy} + 2\mathbf{D}^{(2)}_{xx}\mathbf{D}^{(2)}_{yy}$ corresponding to the biharmonic operator $h^4\delta_{\Delta,\Delta}$ will have a different form for clamped and simply supported conditions. Under clamped conditions it looks like

$$\mathbf{D}_{\Delta,\Delta}\mathbf{u}^n = \underbrace{\left[\begin{array}{c|c|c} \begin{matrix} 20 & -8 & 1 & & & & 0 \\ -8 & 20 & -8 & 1 & & & \\ 1 & -8 & 20 & -8 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 20 & -8 & 1 \\ & & & 1 & -8 & 20 & -8 \\ 0 & & & & 1 & -8 & 20 \end{matrix} & \begin{matrix} -8 & 2 & & & 0 \\ 2 & -8 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & -8 & 2 \\ 0 & & & 2 & -8 \end{matrix} & \begin{matrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{matrix} \\ \hline \begin{matrix} -8 & 2 & & & 0 \\ 2 & -8 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & -8 & 2 \\ 0 & & & 2 & -8 \end{matrix} & \begin{matrix} 20 & -8 & 1 & & & & 0 \\ -8 & 20 & -8 & 1 & & & \\ 1 & -8 & 20 & -8 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 20 & -8 & 1 \\ & & & 1 & -8 & 20 & -8 \\ 0 & & & & 1 & -8 & 20 \end{matrix} & \begin{matrix} -8 & 2 & & & 0 \\ 2 & -8 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & -8 & 2 \\ 0 & & & 2 & -8 \end{matrix} \\ \hline \begin{matrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{matrix} & \begin{matrix} -8 & 2 & & & 0 \\ 2 & -8 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & -8 & 2 \\ 0 & & & 2 & -8 \end{matrix} & \begin{matrix} 20 & -8 & 1 & & & & 0 \\ -8 & 20 & -8 & 1 & & & \\ 1 & -8 & 20 & -8 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 20 & -8 & 1 \\ & & & 1 & -8 & 20 & -8 \\ 0 & & & & 1 & -8 & 20 \end{matrix} \end{array}\right]}_{\text{clamped}} \tag{A.26}$$

$$\times \left[\mathbf{u}^n_1, \mathbf{u}^n_2, \mathbf{u}^n_3, \ldots, \mathbf{u}^n_{Nx-3}, \mathbf{u}^n_{Nx-2}, \mathbf{u}^n_{Nx-1}\right]^\top,$$

whereas under simply supported conditions it is of the form

$$
\mathbf{D}_{\Delta,\Delta}\mathbf{u}^n = \underbrace{\left[\begin{array}{c|c|c}
\begin{smallmatrix} 18 & -8 & 1 & & & & 0 \\ -8 & 19 & -8 & 1 & & & \\ 1 & -8 & 19 & -8 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 19 & -8 & 1 \\ & & & 1 & -8 & 19 & -8 \\ 0 & & & & 1 & -8 & 18 \end{smallmatrix} &
\begin{smallmatrix} -8 & 2 & & & 0 \\ 2 & -8 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & -8 & 2 \\ 0 & & & 2 & -8 \end{smallmatrix} &
\begin{smallmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{smallmatrix} \\ \hline
\begin{smallmatrix} -8 & 2 & & & 0 \\ 2 & -8 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & -8 & 2 \\ 0 & & & 2 & -8 \end{smallmatrix} &
\begin{smallmatrix} 19 & -8 & 1 & & & & 0 \\ -8 & 20 & -8 & 1 & & & \\ 1 & -8 & 20 & -8 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 20 & -8 & 1 \\ & & & 1 & -8 & 20 & -8 \\ 0 & & & & 1 & -8 & 19 \end{smallmatrix} &
\begin{smallmatrix} -8 & 2 & & & 0 \\ 2 & -8 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & -8 & 2 \\ 0 & & & 2 & -8 \end{smallmatrix} \\ \hline
\begin{smallmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{smallmatrix} &
\begin{smallmatrix} -8 & 2 & & & 0 \\ 2 & -8 & 2 & & \\ & \ddots & \ddots & \ddots & \\ & & 2 & -8 & 2 \\ 0 & & & 2 & -8 \end{smallmatrix} &
\begin{smallmatrix} 18 & -8 & 1 & & & & 0 \\ -8 & 19 & -8 & 1 & & & \\ 1 & -8 & 19 & -8 & 1 & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 1 & -8 & 19 & -8 & 1 \\ & & & 1 & -8 & 19 & -8 \\ 0 & & & & 1 & -8 & 18 \end{smallmatrix}
\end{array}\right]}_{\text{simply supported}} \tag{A.27}
$$

$$
\times \left[\mathbf{u}_1^n, \mathbf{u}_2^n, \mathbf{u}_3^n, \ldots, \mathbf{u}_{Nx-3}^n, \mathbf{u}_{Nx-2}^n, \mathbf{u}_{Nx-1}^n\right]^\top,
$$

A.2.3 Rigid Connection between Objects

From this point on we will use the same notation for 1D and 2D matrices and vectors in order to save space and simplify the discussion on object inter-connection as much as possible. To this end we will drop the superscript notation used previously which indicated when we were dealing with 1D or 2D matrices and leave it up to the reader to recognise the explicit shape of any matrices and vectors appearing in the following. A simple means of coupling two distributed objects is through a inter-connection at a single point of rigid type which is defined by

$$
F_c^{(1)} = -\mathcal{M}^{(2,1)} F_c^{(2)} \qquad\qquad (\mathbf{e}^{(1)})^\top u^{(1)} = (\mathbf{e}^{(2)})^\top u^{(2)}, \tag{A.28}
$$

where $\mathcal{M}^{(2,1)} = M^{(2)}/M^{(1)}$ is the mass ratio of the second object to the first and $F_c^{(i)}$ are the forces both objects experiences due to their inter-connection. It is clear that (A.28) satisfies Newton's third law: forces are equal and opposite and the displacement of both objects is the same at the inter-connection point. The force on the first object at time instant n is then given explicitly by

$$F_c^{(1),n} = \frac{-\zeta^{(1)} + \zeta^{(2)}}{\|\mathbf{e}^{(1)}\|^2 + \mathcal{M}^{(1,2)}\|\mathbf{e}^{(2)}\|^2}, \tag{A.29}$$

where the scalar quantity $\zeta^{(i)}$ is given by

$$\begin{aligned}\zeta^{(i)} = \frac{1}{(h^{(i)})^2}(\mathbf{e}^{(i)})^\top \left[\left((\gamma^{(i)})^2 + \frac{2b_2^{(i)}}{k}\right)\mathbf{D}_{xx}^{(i)} - \frac{(\kappa^{(i)})^2}{h^2}\mathbf{D}_{xxxx}^{(i)}\right]\mathbf{u}^{(i),n} \\ - \frac{2b_2^{(i)}}{k(h^{(i)})^2}(\mathbf{e}^{(i)})^\top \mathbf{D}_{xx}^{(i)}\mathbf{u}^{(i),n-1},\end{aligned} \tag{A.30}$$

for $i = 1, 2$. Note that the superscripts used for the difference matrices is used to differentiate between the two objects rather than to indicate the dimensionality of the object as was the case before. Furthermore, it is assumed that in the case of inter-connection of multiple objects to a single object all connection points on the single object are distinct.

A.2.4 Matrix and State Space Form for a Single Object

In absence of any excitation or coupling forces the schemes (A.9) and (A.10) may be put in the same general matrix vector form

$$\mathbf{u}^{n+1} = \mathbf{A}\mathbf{u}^n + \mathbf{B}\mathbf{u}^{n-1}, \tag{A.31}$$

where the matrices $\mathbf{A}$ and $\mathbf{B}$ are given by

$$\mathbf{A} = \frac{1}{1+b_1k}\left[2\mathbf{I} + \left(\frac{\gamma^2k^2}{h^2} + \frac{2b_2k}{h^2}\right)\mathbf{D}_{xx} - \frac{\kappa^2k^2}{h^4}\mathbf{D}_{xxxx}\right] \tag{A.32a}$$

$$\mathbf{B} = -\frac{1}{1+b_1k}\left[(1-b_1k)\,\mathbf{I} + \frac{2b_2k}{h^2}\mathbf{D}_{xx}\right]. \tag{A.32b}$$

As a first step to a modal solution rather than a finite difference one, we will convert (A.31) into state space form [36, 117]. This may be accomplished by collecting the displacement vectors into a state vector $\mathbf{w}$ and the update matrices $\mathbf{A}$ and $\mathbf{B}$ into the state transition block matrix $\mathbb{A}$ to get

$$\underbrace{\begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{u}^{n} \end{bmatrix}}_{\mathbf{w}^{n+1}} = \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}}_{\mathbb{A}} \underbrace{\begin{bmatrix} \mathbf{u}^{n} \\ \mathbf{u}^{n-1} \end{bmatrix}}_{\mathbf{w}^{n}} + \underbrace{\begin{bmatrix} \mathbf{E}_{\text{in}} \\ \mathbf{0} \end{bmatrix}}_{\mathbb{B}} \mathbf{f}_e^n \qquad \text{(A.33)}$$
$$\mathbf{v}^n = \frac{1}{k} \underbrace{\begin{bmatrix} \mathbf{E}_{\text{out}}^\top & -\mathbf{E}_{\text{out}}^\top \end{bmatrix}}_{\mathbb{S}} \underbrace{\begin{bmatrix} \mathbf{u}^{n} \\ \mathbf{u}^{n-1} \end{bmatrix}}_{\mathbf{w}^{n}},$$

where the matrix $\mathbf{I}$ is the identity matrix. Input from some external source into the system is denoted by the block matrix $\mathbb{B}$ multiplied by a column vector $\mathbf{f}_e^n$ which consist of M time varying scalar signals: $\mathbf{f}_e^n = \left[F_{e,1}^n, \ldots, F_{e,M}^n\right]^\top$ and the top block of $\mathbb{B}$ contains the distribution matrix $\mathbf{E}_{\text{in}} = [\mathbf{e}_1, \ldots, \mathbf{e}_M]$, where each column of $\mathbf{E}_{\text{in}}$ is denoting one of M different positions of excitation along the surface of the modelled object. In general the vector $\mathbf{f}_e^n$ may contain different F_e^n's, but also duplicate F_e^n's. The latter case then corresponds to an identical excitation signal which is acting upon multiple different locations along the surface of the object. The column vector $\mathbf{v}^n = [v_1, \ldots, v_N]^\top$ denotes the output (velocity in this case) of the system taken from N different locations along the surface of the modelled object denoted by the N rows of the transposed $\mathbf{E}_{\text{out}}$ matrices which make up the block matrix $\mathbb{S}$.

A.2.5 A System Containing Multiple Objects

Taking equation (A.33) as a starting point, a system consisting of Q distributed objects, assumed uncoupled for the moment, may be defined in state space form as

$$\bar{\mathbf{w}}^{n+1} = \bar{\mathbb{A}}\bar{\mathbf{w}}^n + \bar{\mathbb{B}}\mathbf{f}_e^n \qquad \text{(A.34)}$$
$$\bar{\mathbf{v}}^n = \frac{1}{k}\bar{\mathbb{S}}\bar{\mathbf{w}}^n,$$

where now the state vector $\bar{\mathbf{w}}^n$ consists of $2Q$ displacement vectors according to

$$\bar{\mathbf{w}}^n = \left[\mathbf{u}^{(1),n}, \mathbf{u}^{(1),n-1}, \mathbf{u}^{(2),n}, \mathbf{u}^{(2),n-1}, \ldots, \mathbf{u}^{(Q-1),n}, \mathbf{u}^{(Q-1),n-1}, \mathbf{u}^{(Q),n}, \mathbf{u}^{(Q),n-1}\right]^\top, \qquad \text{(A.35)}$$

and the state transition block matrix $\bar{\mathbb{A}}$ is of the form

$$\bar{\mathbb{A}} = \begin{bmatrix} \mathbf{A}^{(1)} & \mathbf{B}^{(1)} & & & & & & & & \mathbf{0} \\ \mathbf{I}^{(1)} & \mathbf{0} & \mathbf{0} & & & & & & & \\ & \mathbf{0} & \mathbf{A}^{(2)} & \mathbf{B}^{(2)} & & & & & & \\ & & \mathbf{I}^{(2)} & \mathbf{0} & \mathbf{0} & & & & & \\ & & & \ddots & \ddots & \ddots & & & & \\ & & & & \mathbf{0} & \mathbf{A}^{(Q-1)} & \mathbf{B}^{(Q-1)} & & & \\ & & & & & \mathbf{I}^{(Q-1)} & \mathbf{0} & \mathbf{0} & & \\ & & & & & & \mathbf{0} & \mathbf{A}^{(Q)} & \mathbf{B}^{(Q)} \\ \mathbf{0} & & & & & & & \mathbf{I}^{(Q)} & \mathbf{0} \end{bmatrix}. \tag{A.36}$$

The size of the velocity vector $\bar{\mathbf{v}}^n$ is dependent on the number of requested output locations per object making up the complete system and as such depends on the form of the input and output distribution block matrices $\bar{\mathbb{B}}$ and $\bar{\mathbb{S}}$ which will now be of the general form

$$\bar{\mathbb{B}} = \left[(\mathbf{E}_{\text{in}}^{(1)})^\top \quad \mathbf{0} \quad (\mathbf{E}_{\text{in}}^{(2)})^\top \quad \mathbf{0} \quad \cdots \quad (\mathbf{E}_{\text{in}}^{(Q-1)})^\top \quad \mathbf{0} \quad (\mathbf{E}_{\text{in}}^{(Q)})^\top \quad \mathbf{0} \right]^\top, \tag{A.37}$$

and

$$\bar{\mathbb{S}} = \begin{bmatrix} (\mathbf{E}_{\text{out}}^{(1)})^\top & -(\mathbf{E}_{\text{out}}^{(1)})^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & (\mathbf{E}_{\text{out}}^{(2)})^\top & -(\mathbf{E}_{\text{out}}^{(2)})^\top & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & (\mathbf{E}_{\text{out}}^{(Q-1)})^\top & -(\mathbf{E}_{\text{out}}^{(Q-1)})^\top & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} & (\mathbf{E}_{\text{out}}^{(Q)})^\top & -(\mathbf{E}_{\text{out}}^{(Q)})^\top \end{bmatrix}. \tag{A.38}$$

Note that for $\bar{\mathbb{B}}$, a distribution matrix $\mathbf{E}_{\text{in}}^{(q)}$ will only exist if the q^{th} object is being excited. Otherwise it will consist of an appropriately sized $\mathbf{0}$ matrix. Furthermore, the number of columns of each $\mathbf{E}_{\text{in}}^{(q)}$ is equal to the length[3] of the column vector $\mathbf{f}_e^n$ and the number of rows of $\bar{\mathbb{B}}$ is equal to the length of the state vector $\bar{\mathbf{w}}$. Similarly, for the output distribution block matrix $\bar{\mathbb{S}}$, a row block of $\bar{\mathbb{S}}$ will only exist if any output is required

[3]Hence, it may be the case that a column of a particular $\mathbf{E}_{\text{in}}^{(q)}$ contains zeros only in the case an excitation signal is acting upon a single point, but at the same time it is acting upon multiple points for another $\mathbf{E}_{\text{in}}^{(q)}$ (in which case $\mathbf{f}_e^n$ would contain multiple instances of the same F_e^n), or if there are multiple different excitation signals F_e^n present, but only one or a few of them are acting upon object q.

from a specific object and instead now the number of columns of $\bar{\mathbb{S}}$ will be equal to the length of the state vector $\bar{\mathbf{w}}$.

A.2.6 Defining Object Inter-connection

The inter-connection of any two of the Q objects making up the system introduced in the previous section may be accomplished by incorporating the appropriate coupling matrices into the state transition block matrix $\bar{\mathbb{A}}$. To illustrate this we will first consider the most simple example possible: a system consisting of two objects which are inter-connected at a single point. Using equations (A.29) and (A.30) our state transition block matrix $\bar{\mathbb{A}}$ will look like

$$\mathbb{A} = \begin{bmatrix} \mathbf{A}^{(1)} + \mathbf{C}_1^{(1,2)} & \mathbf{B}^{(1)} + \mathbf{C}_2^{(1,2)} & \mathbf{C}_3^{(1,2)} & \mathbf{C}_4^{(1,2)} \\ \mathbf{I}^{(1)} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_3^{(2,1)} & \mathbf{C}_4^{(2,1)} & \mathbf{A}^{(2)} + \mathbf{C}_1^{(2,1)} & \mathbf{B}^{(2)} + \mathbf{C}_2^{(2,1)} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}^{(2)} & \mathbf{0} \end{bmatrix},$$

where the coupling matrices $\mathbf{C}_1^{(i,j)}$, $\mathbf{C}_2^{(i,j)}$, $\mathbf{C}_3^{(i,j)}$ and $\mathbf{C}_4^{(i,j)}$ are given by

$$\begin{aligned}
\mathbf{C}_1^{(i,j)} &= \frac{-\alpha^{(i)}}{1+b_1^{(i)}k} \frac{\mathbf{e}^{(i)} \otimes \mathbf{e}^{(i)}}{\|\mathbf{e}^{(i)}\|^2 + \mathcal{M}^{(i,j)}\|\mathbf{e}^{(j)}\|^2} \left[\left(\frac{(\gamma^{(i)})^2 k^2}{(h^{(i)})^2} + \frac{2b_2^{(i)}k}{(h^{(i)})^2} \right) \mathbf{D}_{xx}^{(i)} - \frac{(\kappa^{(i)})^2 k^2}{(h^{(i)})^4} \mathbf{D}_{xxxx}^{(i)} \right] \\
\mathbf{C}_2^{(i,j)} &= \frac{\alpha^{(i)}}{1+b_1^{(i)}k} \frac{2b_2^{(i)}k}{(h^{(i)})^2} \frac{\mathbf{e}^{(i)} \otimes \mathbf{e}^{(i)}}{\|\mathbf{e}^{(i)}\|^2 + \mathcal{M}^{(i,j)}\|\mathbf{e}^{(j)}\|^2} \mathbf{D}_{xx}^{(i)} \\
\mathbf{C}_3^{(i,j)} &= \frac{\alpha^{(i)}}{1+b_1^{(i)}k} \frac{\mathbf{e}^{(i)} \otimes \mathbf{e}^{(j)}}{\|\mathbf{e}^{(i)}\|^2 + \mathcal{M}^{(i,j)}\|\mathbf{e}^{(j)}\|^2} \left[\left(\frac{(\gamma^{(j)})^2 k^2}{(h^{(j)})^2} + \frac{2b_2^{(j)}k}{(h^{(j)})^2} \right) \mathbf{D}_{xx}^{(j)} - \frac{(\kappa^{(j)})^2 k^2}{(h^{(j)})^4} \mathbf{D}_{xxxx}^{(j)} \right] \\
\mathbf{C}_4^{(i,j)} &= -\frac{\alpha^{(i)}}{1+b_1^{(i)}k} \frac{2b_2^{(j)}k}{(h^{(j)})^2} \frac{\mathbf{e}^{(i)} \otimes \mathbf{e}^{(j)}}{\|\mathbf{e}^{(i)}\|^2 + \mathcal{M}^{(i,j)}\|\mathbf{e}^{(j)}\|^2} \mathbf{D}_{xx}^{(j)},
\end{aligned}$$

for $i, j = 1, 2$ and $i \neq j$. Here the constant α is dependent on the dimensionality of the problem and corresponds to the constant in front of the sum which forms the last term on the right-hand side of equations (A.9) and (A.10). Hence $\alpha = 1/h$ for 1D objects and $\alpha = 1/h^2$ for 2D objects. In the most general setting an arbitrary number of objects may be inter-connected and there may be multiple connections between any two objects. In this case the coupling matrices associated with a certain object (i.e. all coupling matrices appearing in a row of $\bar{\mathbb{A}}$) will involve a sum over all connections this object has with the other objects making up the complete system. Hence the first coupling matrix $\mathbf{C}_1^{(q)}$

corresponding to the q^{th} of a total of Q objects will appear on every uneven entree along the diagonal of $\bar{\mathbb{A}}$ and is of the form

$$\mathbf{C}^{(q)} = -\frac{\alpha^{(q)}}{1+b_1^{(q)}k}\sum_r^R\sum_s^S \frac{\mathbf{e}_s^{(q,r)}\otimes\mathbf{e}_s^{(q,r)}}{\|\mathbf{e}_s^{(q,r)}\|^2+\mathcal{M}^{(q,r)}\|\mathbf{e}_s^{(r,q)}\|^2} \times\left[\left(\frac{(\gamma^{(q)})^2k^2}{(h^{(q)})^2}+\frac{2b_2^{(q)}k}{(h^{(q)})^2}\right)\mathbf{D}_{xx}^{(q)}-\frac{(\kappa^{(q)})^2k^2}{(h^{(q)})^4}\mathbf{D}_{xxxx}^{(q)}\right]. \quad \text{(A.39)}$$

Here the index r is used to denote the r^{th} of R different objects which are connected to object q and the index s is used to indicate the s^{th} of S connections between object q and r and hence, in the case of only one connection between two objects, the innermost sum over s vanishes. Note also that we have changed our notation to signify the inter-connection of different objects slightly. Now the superscript (q,r) used for all $\mathbf{e}$ vectors corresponds to the point of connection as seen from object q, and similarly (r,q) corresponds to the point of connection as seen from object r. The second coupling matrix $\mathbf{C}_2^{(q)}$ appears as the row entree directly to the right of $\mathbf{C}_1^{(q)}$ and is given by

$$\mathbf{C}_2^{(q)} = \frac{\alpha^{(q)}}{1+b_1^{(q)}k}\frac{2b_2^{(q)}k}{(h^{(q)})^2}\sum_r^R\sum_s^S\frac{\mathbf{e}^{(q,r)}\otimes\mathbf{e}^{(q,r)}}{\|\mathbf{e}^{(q,r)}\|^2+\mathcal{M}^{(q,r)}\|\mathbf{e}^{(r,q)}\|^2}\mathbf{D}_{xx}^{(q)}. \quad \text{(A.40)}$$

The column index of the coupling matrices $\mathbf{C}_3^{(q,r)}$ and $\mathbf{C}_4^{(q,r)}$ is dependent on the index of r and is given by the $(2r-1)^{th}$ and $2r^{th}$ block in the q^{th} row respectively:

$$\mathbf{C}_3^{(q,r)} = \frac{\alpha^{(q)}}{1+b_1^{(q)}k}\sum_s^S\frac{\mathbf{e}^{(q,r)}\otimes\mathbf{e}^{(r,q)}}{\|\mathbf{e}^{(q,r)}\|^2+\mathcal{M}^{(q,r)}\|\mathbf{e}^{(r,q)}\|^2} \times\left[\left(\frac{(\gamma^{(r)})^2k^2}{(h^{(r)})^2}+\frac{2b_2^{(r)}k}{(h^{(r)})^2}\right)\mathbf{D}_{xx}^{(r)}-\frac{(\kappa^{(r)})^2k^2}{(h^{(r)})^4}\mathbf{D}_{xxxx}^{(r)}\right] \quad \text{(A.41)}$$

$$\mathbf{C}_4^{(q,r)} = -\frac{\alpha^{(q)}}{1+b_1^{(q)}k}\frac{2b_2^{(r)}k}{(h^{(r)})^2}\frac{\mathbf{e}^{(q,r)}\otimes\mathbf{e}^{(r,q)}}{\|\mathbf{e}^{(q,r)}\|^2+\mathcal{M}^{(q,r)}\|\mathbf{e}^{(r,q)}\|^2}\mathbf{D}_{xx}^{(r)}. \quad \text{(A.42)}$$

Hence, the general form of the state transition block matrix $\bar{\mathbb{A}}$ looks like

$$\bar{\mathbb{A}} = \begin{bmatrix} \tilde{\mathbf{A}}^{(1)} & \tilde{\mathbf{B}}^{(1)} & \mathbf{C}_3^{(1,2)} & \mathbf{C}_4^{(1,2)} & \cdots & & \cdots & \mathbf{C}_3^{(1,Q)} & \mathbf{C}_4^{(1,Q)} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_3^{(2,1)} & \mathbf{C}_4^{(2,1)} & \tilde{\mathbf{A}}^{(2)} & \tilde{\mathbf{B}}^{(2)} & \mathbf{C}_3^{(2,3)} & \mathbf{C}_4^{(2,3)} & \cdots & \mathbf{C}_3^{(2,Q)} & \mathbf{C}_4^{(2,Q)} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{C}_3^{(Q-1,1)} & \mathbf{C}_4^{(Q-1,1)} & \cdots & \mathbf{C}_3^{(Q-1,2)} & \mathbf{C}_4^{(Q-1,2)} & \tilde{\mathbf{A}}^{(Q-1)} & \tilde{\mathbf{B}}^{(Q-1)} & \mathbf{C}_3^{(Q-1,Q)} & \mathbf{C}_4^{(Q-1,Q)} \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{C}_3^{(Q,1)} & \mathbf{C}_4^{(Q,1)} & \cdots & & \cdots & \mathbf{C}_3^{(Q,Q-1)} & \mathbf{C}_4^{(Q,Q-1)} & \tilde{\mathbf{A}}^{(Q)} & \tilde{\mathbf{B}}^{(Q)} \\ \mathbf{0} & \mathbf{0} & \cdots & & \cdots & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{A.43}$$

where $\tilde{\mathbf{A}}^{(q)} = \mathbf{A}^{(q)} - \mathbf{C}_1^{(q)}$ and $\tilde{\mathbf{B}}^{(q)} = \mathbf{B}^{(q)} - \mathbf{C}_2^{(q)}$. Note that all $\mathbf{C}$ matrices exist only if there is an actual connection between the two objects denoted by the superscripts and therefore represent a hypothetical occurrence only. The size of the Q state vectors $\mathbf{w}$ will be given by $N_q = 2\,(N+1)$ (free boundary conditions) or $N_q = 2\,(N-1)$ (clamped or simply supported boundary conditions) for a 1D object or by $N_q = 2\,(N_x - 1)\,(N_y - 1)$ for a 2D object. Hence, the size of the state vector $\bar{\mathbf{w}}$ is $\bar{N} = \sum_{q=1}^{Q} N_q$ and $\bar{A}$ will be a $\bar{N} \times \bar{N}$ matrix.

A.2.7 Changing to Modal State Space Form

We may change to modal coordinates by employing a similarity transformation on our state transition block matrix $\bar{\mathbf{A}}$. If $\mathbf{E}$ and $\mathbf{D}$ are two arbitrary $n \times n$ matrices, $\mathbf{D}$ is called similar to $\mathbf{E}$ if $\mathbf{D} = \mathbf{G}^{-1}\mathbf{E}\mathbf{G}$. It then holds that $\mathbf{E}$ is also similar to $\mathbf{D}$ and hence will have the same characteristic equation and eigenvalues [118]. Here the matrix $\mathbf{D}$ is a diagonal matrix given by $\mathbf{D} = \mathrm{diag}\,(\lambda_1, \ldots, \lambda_n)$ containing the eigenvalues of $\mathbf{E}$ and the column vectors making up the matrix $\mathbf{G}$ will correspond to the eigenvectors associated with these eigenvalues. When applied to our state transition block matrix $\bar{\mathbb{A}}$, the eigenvalues will correspond to the normal mode frequencies and damping constants and the eigenvectors represent the modal shapes of our system. Hence, diagonalising our system described by equation (A.34) earlier will result in [119]

$$\tilde{\mathbf{w}}^{n+1} = \tilde{\mathbf{A}}\tilde{\mathbf{w}}^{n} + \tilde{\mathbf{B}}\mathbf{f}_e^{n} \tag{A.44}$$

$$\mathbf{v}^{n} = \frac{1}{k}\tilde{\mathbf{S}}\tilde{\mathbf{w}}^{n}, \tag{A.45}$$

where

$$\tilde{\mathbf{A}} = \mathbf{G}^{-1}\bar{\mathbb{A}}\mathbf{G} \qquad \tilde{\mathbf{B}} = \mathbf{G}^{-1}\bar{\mathbb{B}} \qquad \tilde{\mathbf{S}} = \bar{\mathbb{S}}\mathbf{G}. \tag{A.46}$$

The $N \times M$ transfer function matrix associated with our diagonalised state space system [117] is given by

$$\begin{aligned}
\mathbf{H}(z) &= \tilde{\mathbf{S}}\left(z\mathbf{I} - \tilde{\mathbf{A}}\right)^{-1}\tilde{\mathbf{B}} \qquad\qquad (A.47)\\
&= \begin{bmatrix} \tilde{s}_{1,1} & \tilde{s}_{1,2} & \cdots & \tilde{s}_{1,\bar{N}-1} & \tilde{s}_{1,\bar{N}} \\ \tilde{s}_{2,1} & \tilde{s}_{2,2} & \cdots & \tilde{s}_{2,\bar{N}-1} & \tilde{s}_{2,\bar{N}} \\ \vdots & \vdots & & \vdots & \vdots \\ \tilde{s}_{N-1,1} & \tilde{s}_{N-1,2} & \cdots & \tilde{s}_{N-1,\bar{N}-1} & \tilde{s}_{N-1,\bar{N}} \\ \tilde{s}_{N,1} & \tilde{s}_{N,2} & \cdots & \tilde{s}_{N,\bar{N}-1} & \tilde{s}_{N,\bar{N}} \end{bmatrix} \begin{bmatrix} \frac{z^{-1}}{1-\lambda_1 z^{-1}} & & \\ & \ddots & \\ & & \frac{z^{-1}}{1-\lambda_{\bar{N}} z^{-1}} \end{bmatrix}\\
&\times \begin{bmatrix} \tilde{b}_{1,1} & \tilde{b}_{1,2} & \cdots & \tilde{b}_{1,M-1} & \tilde{b}_{1,M} \\ \tilde{b}_{2,1} & \tilde{b}_{2,2} & \cdots & \tilde{b}_{2,M-1} & \tilde{b}_{2,M} \\ \vdots & \vdots & & \vdots & \vdots \\ \tilde{b}_{\bar{N}-1,1} & \tilde{b}_{\bar{N}-1,2} & \cdots & \tilde{b}_{\bar{N}-1,M-1} & \tilde{b}_{\bar{N}-1,M} \\ \tilde{b}_{\bar{N},1} & \tilde{b}_{\bar{N},2} & \cdots & \tilde{b}_{\bar{N},M-1} & \tilde{b}_{\bar{N},M} \end{bmatrix}
\end{aligned}$$

From this follows that the diagonal matrix $\tilde{\mathbf{A}}$, containing $\bar{N}$ complex valued eigenvalues, represents the poles of the transfer function matrix associated with our system of interconnected objects. Hence, every $H_{i,j}(z)$ in $\mathbf{H}(z)$ corresponds to a parallel bank of $\bar{N}$ complex first order sections. As the eigenvalues λ_n and the entrees in row$_i$ $\tilde{\mathbf{S}}$ and col$_j$ $\tilde{\mathbf{B}}$ appear in complex conjugate pairs, first order sections may be combined in order to create $\bar{N}/2$ real valued second order sections of the form

$$H_{i,j}(z) = \sum_{n=1}^{\bar{N}/2} \frac{-2\left(\xi_1 z^{-1} + \xi_2 z^{-2}\right)}{1 - 2\mathrm{Re}\left(\lambda_n\right) z^{-1} + |\lambda_n|^2 z^{-2}}, \tag{A.48}$$

where

$$\xi_1 = -\mathrm{Re}\left(s_{i,n}\right)\mathrm{Re}\left(b_{n,j}\right) + \mathrm{Im}\left(s_{i,n}\right)\mathrm{Im}\left(b_{n,j}\right) \tag{A.49a}$$

$$\begin{aligned}
\xi_2 = {} & \left(\mathrm{Re}\left(s_{i,n}\right)\mathrm{Re}\left(b_{n,j}\right) - \mathrm{Im}\left(s_{i,n}\right)\mathrm{Im}\left(b_{n,j}\right)\right)\mathrm{Re}\left(\lambda_n\right) \\
& + \left(\mathrm{Re}\left(s_{i,n}\right)\mathrm{Im}\left(b_{n,j}\right) + \mathrm{Im}\left(s_{i,n}\right)\mathrm{Re}\left(b_{n,j}\right)\right)\mathrm{Im}\left(\lambda_n\right).
\end{aligned} \tag{A.49b}$$

In conclusion, every mode of our model system described by equation (A.34) may be realised as a second order resonator of the form given by equation (A.48) with frequency

$$f_n = \frac{1}{2\pi T} \cos^{-1}\left(\frac{\mathrm{Re}\,(\lambda_n)}{|\lambda_n|}\right), \tag{A.50}$$

where $T = k$ is the sampling period, $|\lambda_n| \leq 1$ is the pole radius and the t_{60} decay time – i.e. the time in which the amplitude of the resonator will drop by 60 dB – is approximately given by [117]

$$t_{60,n} \approx \frac{6.91T}{1 - |\lambda_n|}. \tag{A.51}$$

Appendix B

PMLib Code Documentation

B.1 Installation Instructions

Installation instructions will be provided for Mac OSX operating systems only, as this was the platform of choice when developing the library. Instructions for Linux and Windows will be largely identical however. Installing *PMLib* is relatively straightforward. The minimum requirements are a working Python 2.7.x version with the latest NumPy and SciPy packages installed. This will allow one to define a system of inter-connected objects and calculate its modal data in Python. The computed modal data may be saved to disk in JSON format, after which it can be used in any environment which offers the possibility to read JSON files and allows for the simulation of parallel arrangements of second order filter sections (e.g. Web Audio API, Max/MSP, PD, Chuck or Csound).

As the portfolio works have been composed with the help of the SuperCollider programming language, an additional SuperCollider based interface has been developed which functions as a bridge between SuperCollider and Python. Hence, the user can accomplish all necessary steps for computing the modal data associated with the model system without having to leave the SuperCollider environment at any stage. In order to use the library with SuperCollider, one needs to place the *PMLib* directory within the SuperCollider search path. This can be done in either one of two ways. The first one is to install

PMLib as a quark[1]. The second one is to clone the *PMLib* GitHub repository[2] into a location which is within SuperCollider's search path. On OSX this is typically something like **/Users/yourusername/Library/Application Support/SuperCollider/Extensions**. One can verify that *PMLib* was installed successfully if there are no errors appearing in the SuperCollider console at startup of the application and by confirming that typing the line `ResonatorNetwork` in an empty document brings up the popup menu with suggested class names. The last thing to verify is that the `pythonPath` class variable points to the location of the right Python version (i.e. the version 2.7.x for which you have installed NumPy and SciPy). By default this is set to **/Library/Frameworks/Python.framework/Versions/Current/bin**, which points to the current version of Python on OSX. If this class variable is not set appropriately, it is likely that the library will fail silently when prompted to compute any modal data (i.e. no error message will appear in the SuperCollider console).

Optionally, one may consider to download and compile the `SOSBank` UGen plugin[3]. This UGen implements a parallel arrangement of second order resonators internally and as such offers a more convenient alternative to having to create multiple `SOS` UGens inside a synth def.

B.2 Class Reference

PMLib is based upon an object oriented approach to programming as this seemed the most logical and concise way of specifying a system of inter-connected objects. To this end, a number of classes have been defined which all serve a specific goal in the chain of computing events. These classes exist both in a SuperCollider as well as a Python specific version. Each of these classes should be interpreted as representing the same object in either language, although their properties and methods differ according to the programmatic purpose they fulfil in either SuperCollider or Python. The ones the user should interact with directly are documented below.

[1]Quarks are extensions to the standard SuperCollider code base. They may contain new classes, extension methods, documentation and UGen plugins. For more information on quarks and how to install them see the Quarks help file accessible through the SuperCollider IDE.

[2]`https://github.com/michaeldzjap/PMLib`

[3]This repository may be cloned from `https://github.com/michaeldzjap/SOSBank`.

B.2.1 Resonator1D

Represents a single linear 1D resonator like a string with stiffness and damping or a thin bar with damping.

`Resonator1D.sc` variables:

class variable	type	default	description
`validBoundaryConds`	array		An immutable array of valid boundary conditions one is free to choose from.
`sampleRate`	integer	`44100`	An integer denoting the sample rate.
`timeStep`	float	`1/44100`	A float denoting the time step.

instance variable	type	default	description
`gamma`	number	`200`	A number denoting a spatially scaled wavespeed parameter.
`kappa`	number	`1`	A number denoting a spatially scaled stiffness parameter.
`b1`	number	`0`	A number denoting a frequency independent damping constant.
`b2`	number	`0`	A number denoting a frequency dependent damping constant.
`boundaryCond`	symbol	`'bothSimplySupported'`	A symbol denoting a valid 1D boundary condition.

`Resonator1D.sc` class methods:

- `new(gamma, kappa, b1, b2, boundaryCond)`

  Create a new instance of `Resonator1D` with the given instance variable values.

`Resonator1D.sc` instance methods:

- `jsonString()`

  Generate and return a JSON representation of a `Resonator1D` object instance.

`Resonator1D.py` attributes:

class attribute	**type**	**default**	**description**
`SR`	integer	`44100`	An integer denoting the sample rate.
`k`	float	`1/44100`	A float denoting the time step.

instance attribute	**type**	**default**	**description**
`gamma`	number	`200`	A number denoting a spatially scaled wavespeed parameter.
`kappa`	number	`1`	A number denoting a spatially scaled stiffness parameter.
`b1`	number	`0`	A number denoting a frequency independent damping constant.
`b2`	number	`0`	A number denoting a frequency dependent damping constant.
`boundaryCond`	symbol	`'BothSimplySupported'`	A number denoting a frequency dependent damping constant.

`Resonator1D.py` class methods:

- `newInst = Resonator1D(gamma, kappa, b1, b2, boundaryCond)`

  Create a new instance of `Resonator1D` with the given instance variable values.

`Resonator1D.py` public instance methods:

- `constrUpdateMatrices()`

  Construct this objects finite difference update matrices.

- `constrCouplingMatrices()`

  Construct this objects finite difference coupling matrices.

B.2.2 Resonator2D

Represents a single linear 2D resonator like a membrane with stiffness and damping or a thin plate with damping.

`Resonator2D.sc` variables:

class variable	**type**	**default**	**description**
`validBoundaryConds`	array		An immutable array of valid boundary conditions one is free to choose from.
`sampleRate`	integer	`44100`	An integer denoting the sample rate.
`timeStep`	float	`1/44100`	A float denoting the time step.

instance variable	**type**	**default**	**description**
`gamma`	number	`200`	A number denoting a spatially scaled wavespeed parameter.
`kappa`	number	`1`	A number denoting a spatially scaled stiffness parameter.
`b1`	number	`0`	A number denoting a frequency independent damping constant.
`b2`	number	`0`	A number denoting a frequency dependent damping constant.
`boundaryCond`	symbol	`'allSidesSimplySupported'`	A symbol denoting a valid 2D boundary condition.
`epsilon`	number	`1`	A number denoting the aspect ratio: L_x/L_y of the rectangular domain (1 represents a square).

`Resonator2D.sc` class methods:

- `new(gamma, kappa, b1, b2, boundaryCond, epsilon)`

  Create a new instance of `Resonator2D` with the given instance variable values.

`Resonator2D.sc` instance methods:

- `jsonString()`

  Generate and return a JSON representation of a `Resonator2D` object instance.

`Resonator2D.py` attributes:

class attribute	**type**	**default**	**description**
`SR`	integer	`44100`	An integer denoting the sample rate.
`k`	float	`1/44100`	A float denoting the time step.

instance attribute	**type**	**default**	**description**
`gamma`	number	`200`	A number denoting a spatially scaled wavespeed parameter.
`kappa`	number	`1`	A number denoting a spatially scaled stiffness parameter.
`b1`	number	`0`	A number denoting a frequency independent damping constant.
`b2`	number	`0`	A number denoting a frequency dependent damping constant.
`boundaryCond`	symbol	`'AllSidesSimplySupported'`	A number denoting a frequency dependent damping constant.
`epsilon`	number	`1`	A number denoting the aspect ratio: L_x/L_y of the rectangular domain (1 represents a square).

`Resonator2D.py` class methods:

- `newInst = Resonator1D(gamma, kappa, b1, b2, boundaryCond)`

  Create a new instance of `Resonator2D` with the given instance variable values.

`Resonator2D.py` public instance methods:

- `constrUpdateMatrices()`

  Construct this objects finite difference update matrices.

- `constrCouplingMatrices()`

  Construct this objects finite difference coupling matrices.

B.2.3 ResonatorNetwork

Represents a network of inter-connected 1D and 2D dimensional resonator objects.

`ResonatorNetwork.sc` variables:

class variable	**type**	**default**	**description**
`pythonPath`	string	`"/Library/Frameworks/..."`	A path pointing to the version of Python to use for all numerical computations.

instance variable	type	default	description
`resonators`	array	`[Resonator1D, ...]`	An array of `Resonator1D` and/or `Resonator2D` objects.
`connPointMatrix`	2d array	`[[0.5], [0.5]]`	A 2d array of floats (or 2 item array of floats in case of a 2D object) denoting the relative connection point (0 - 1) on the resonator object.
`massMatrix`	2d array	`[[1], [1]]`	A 2d array of numbers denoting the masses of the resonator object.
`excPointMatrix`	2d array	`[[0.25], [0]]`	A 2d array of floats (or 2 item array of floats in case of a 2D object) denoting the relative excitation point (0 - 1) on the resonator object.
`readoutPointMatrix`	2d array	`[[0.5], [0]]`	A 2d array of floats (or 2 item array of floats in case of a 2D object) denoting the relative readout point (0 - 1) on the resonator object.
`modalData`	dictionary	`nil`	Initially nil. After calling `calcModalData()` successfully, this variable will hold a dictionary containing the modal data associated with this instance of `ResonatorNetwork`.

`ResonatorNetwork.sc` class methods:

- `new(resonators, connPointMatrix, massMatrix, excPointMatrix, readoutPointMatrix)`

  Create a new instance of `ResonatorNetwork` with the given instance variable values.

`ResonatorNetwork.sc` instance methods:

- `calcModalData(minFreq, maxFreq, minT60, gain, pathname, incl, async)` Calculates the modal data associated with this instance of `ResonatorNetwork`. This is stored in the `modalData` instance variable and is also written to disk as a JSON file. The full path of this JSON file customisable by specifying the `pathname` argument. The `minFreq` and `maxFreq` arguments determine the minimum and maximum modes to include. The `minT60` may be used to exclude any modes with a T60 decay time which falls below the specified value. The `gain` arguments allows one to boost the gain of the filter sections if necessary. The `incl` arguments is a four character string, where each character can either be a `"y"` or a `"n"`. The first character determines if the modal data should be included in digital filter coefficients form, the second if the eigenvalues should be included in polar form, the third if the eigenvalues should be included in rectangular form and the last character if the eigenvectors should be included. The default is to return the digital filter coefficients only (`incl = "ynnn"`) The `async` argument determines if the computations should be executed synchronously (default) or asynchronously.

- `loadModalData(pathname)`
  Load previously computed modal data in JSON format from disk. Note that the user is responsible for making sure that this modal data actually corresponds to the current object configuration of this instance of `ResonatorNetwork`.

`ResonatorNetwork.py` attributes:

class attribute	**type**	**default**	**description**
`SR`	integer	`44100`	An integer denoting the sample rate.
`k`	float	`1/44100`	A float denoting the time step.

instance attribute	**type**	**default**	**description**
`resonators`	array	`[Resonator1D, ...]`	An array of `Resonator1D` and/or `Resonator2D` objects.
`connPointMatrix`	2d array	`[[0.5], [0.5]]`	A 2d array of floats (or 2 item array of floats in case of a 2D object) denoting the relative connection point (0 - 1) on the resonator object.
`massMatrix`	2d array	`[[1], [1]]`	A 2d array of numbers denoting the masses of the resonator object.
`excPointMatrix`	2d array	`[[0.25], [0]]`	A 2d array of floats (or 2 item array of floats in case of a 2D object) denoting the relative excitation point (0 - 1) on the resonator object.
`readoutPointMatrix`	2d array	`[[0.5], [0]]`	A 2d array of floats (or 2 item array of floats in case of a 2D object) denoting the relative readout point (0 - 1) on the resonator object.

`ResonatorNetwork.py` class methods:

- `newInst = ResonatorNetwork(resonators, connPointMatrix, massMatrix, excPointMatrix, readoutPointMatrix)`

  Create a new instance of `ResonatorNetwork` with the given instance variable values.

`ResonatorNetwork.py` public instance methods:

- `calcModes(minFreq, maxFreq, minT60)`

  Calculates the modal data associated with this instance of `ResonatorNetwork.py`. Discard any modes that fall outside the range `minFreq`, `maxFreq` or which have a T60 decay time smaller than `minT60`.

- `calcBiquadCoefs(gain)`

  Calculate all digital filter coefficients corresponding to a particular arrangement of input and output conditions and possibly boost the gain of the resulting transfer functions by multiply by the `gain` argument.

- `saveAsJSON(path, include)` Save the calculated modal data to disk as a JSON file at a location specified by the `pathname` argument.

Bibliography

[1] Stefan Bilbao. A modular percussion synthesis environment. In *Proc. of the* 12^{th} *Int. Conference on Digital Audio Effect (DAFx-09)*, pages 1–8, Como, Italy, September 1–4 2009.

[2] Claude Cadoz, Annie Luciani, and Jean Loup Florens. CORDIS-ANIMA: A modeling and simulation system for sound and image synthesis – the general formalism. *Computer Music Journal*, 17(1):19–29, Spring 1993.

[3] Nicholas Ellis, Joël Bensoam, and René Caussé. Modalys demonstration. In *Proc. of the 2005 International Computer Music Conference*, pages 1–2, 2005.

[4] Mark Pearson. Tao: a physical modelling system and related issues. *Organised Sound*, 1(1):43–50, 1996.

[5] Hans Tutschku. *Portfolio of Compositions*. PhD Thesis, Department of Music, School of Humanities, The University of Birmingham, March 2003.

[6] Van Stiefel, Dan Trueman, and Perry Cook. *Re*-coupling: the *uBlotar* synthesis instrument and the *sHowl* speaker-feedback controller. In *Proc. of the 2004 International Computer Music Conference*, 2004.

[7] Fredrik Eckerholm. Physical modeling of musical intruments – interaction models. Master's thesis, Linköping University, Norrköping, Sweden, 2008.

[8] Steven Gelineck. *Exploratory and Creative Properties of Physical-Modeling-based Musical Intruments: Developing a framework for the development of physical modelling based digital musical instruments, which encourage exploration and creativity.* PhD Thesis, Aalborg University Copenhagen, Department of Architecture, Design and Media Technology, 2012.

[9] Huhues Vinet. Recent research and development at IRCAM. *Computer Music Journal*, 23(3):9–17, Autumn 1999.

[10] Georgina Born. *Rationalizing Culture: IRCAM, Boulez and the Institutionalisation of the Musical Avantgarde*. University of California Press, 1995. ISBN 0-520-08507-8.

[11] Pierre Boulez. Technology and the composer. In Simon Emmerson, editor, *The Language of Electroacoustic Music*, pages 5–14. Palgrave Macmillan, 3 December 1986.

[12] Sarah Orr and Maarten van Walstijn. Modal representation of the resonant body within a finite difference framework for simulation of string instruments. In *Proc. of the* 12^{th} *Int. Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, September 1–4 2009.

[13] Andy Farnell. *Designing Sound*. The MIT Press, 2010. ISBN 978-0-262-01441-0.

[14] Michel Chion. *Guide Des Objets Sonores*. Éditions Buchet/Chastel, 1983.

[15] Edward N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20:130–141.

[16] Michaël Dzjaparidze. Composing music with algorithms: Mapping the schrödinger equation to obtain musical form and function. Master's thesis, Hogeschool voor de Kunsten Utrecht, Faculty of Art, Media & Technology, Hilversum, The Netherlands, 2010.

[17] Michaël Dzjaparidze. From equation to creation: Using the schrödinger equation in an artistic context. Technical report, Centrum voor Wiskunde & Informatica (CWI), Amsterdam, The Netherlands, November 2009.

[18] Christoph Cox. Return to form. `http://soundartarchive.net/articles/Cox-2003-Neo-modernist sound art .pdf`, November 2003. Accessed 25 November, 2014.

[19] Perry R. Cook and Gary P. Scavone. The Synthesis ToolKit (STK). In *Proc. of the 1999 International Computer Music Conference*, Beijing, China, 1999.

[20] Julius O. Smith III. Viewpoints on the history of digital synthesis. In *Proc. of the 1991 International Computer Music Conference*, pages 1–10, Montreal, 1991.

[21] Lutz Trautmann and Rudolf Rabenstein. *Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method*. Kluwer Academic / Plenum Publishers, New York, 2003.

[22] Julien Bensa, Stefan Bilbao, Richard Kronland-Martinet, and Julius O. Smith III. The simulation of piano string vibration: From physical models to finite difference schemes and digital waveguides. *J. Acoust. Soc. Am.*, 114(2):1095–1107, August 2003.

[23] Neville Horner Fletcher and Thomas D. Rossing. *The Physics of Musical Intruments*. Springer Science + Business Media, Inc., second edition, 1998. ISBN 978-1-441-93120-7.

[24] Stefan Bilbao. *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. John Wiley & Sons Ltd, 2009. ISBN 978-0-470-51046-9.

[25] Gianpaolo Borin, Giovanni De Poli, and Augusto Sarti. Algorithms and structures for synthesis using physical models. *Computer Music Journal*, 16(4):30–42, Winter 1992.

[26] Nicholas Castagne and Claude Cadoz. 10 criteria for evaluating physical modelling schemes for musical creation. In *Proc. of the* 6^{th} *Int. Conference on Digital Audio Effects (DAFx-03)*, pages 1–7, London, UK, September 8–11 2003.

[27] Tero Tolonen, Vesa Välimäki, and Matti Karjalainen. Evaluation of modern sound synthesis methods. Technical Report 48, Helsinki University of Technology, Department of Electrical and Communications Engineering, Laboratory of Acoustics and Audio Signal Processing, Espoo, March 1998.

[28] Julius O. Smith III. Physical modeling using digital waveguides. *Computer Music Journal*, 16(4):74–91, Winter 1992.

[29] S. A. Van Duyne and J. O. Smith III. A simplified approach to modeling dispersion caused by stiffness in strings and plates. In *Proc. of the 1994 International Computer Music Conference*, pages 407–410, September 1993.

[30] Stefan Bilbao. *Wave and Scattering Methods for Numerical Simulation.* John Wiley & Sons Ltd, 2004. ISBN 978-0-470-87017-6.

[31] Marc-Laurent Aird. *Musical Instrument Modelling Using Digital Waveguides.* PhD Thesis, University of Bath, 2002.

[32] Maarten van Walstijn and Eoin Mullan. Time-domain simulation of rectangular membrane vibrations with 1-d digital waveguides. In *Forum Acusticum 2011*, Aalborg, Denmark, 2011.

[33] Matti Karjalainen and Cumhur Erkut. Digital waveguides versus finite difference structures: Equivalence and mixed modeling. *EURASIP Journal on Applied Signal Processing*, 2004(7):978–989, 2004.

[34] John C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations.* Wadsworth & Brooks / Cole Advanced Books & Software, siam edition, 2004. ISBN 0-89871-567-9.

[35] Cynthia Bruyns. Modal synthesis for arbitrarily shaped objects. *Computer Music Journal*, 30(3):22–37, Fall 2006.

[36] Stefan Bilbao and Julius O. Smith III. Modal synthesis of a piano string. Lecture Supplement, March 15 2010.

[37] Bernd Girod, Rudolf Rabenstein, and Alexander Stenger. *Signals and Systems.* John Wiley & Sons, LTD, 2001. ISBN 0 471 98800 6.

[38] Balázs Bank and László Sujbert. Generation of longitudinal vibrations in piano strings: From physics to sound synthesis. *J. Acoust. Soc. Am.*, 117(4):2268–2278, April 2005.

[39] Oliver Thomas, C. Touzé, and A. Chaigne. Non-linear vibrations of free-edge thin spherical shells: Modal interaction rules and 1:1:2 internal resonance. *Int. Journal of Solids and Structures*, 42:3339–3373, 2005.

[40] C. Touzé, O. Thomas, and A. Chaigne. Asymmetric non-linear forced vibrations of free-edge circular plates. part 1: Theory. *Journal of Sound and Vibration*, 258 (4):649–676, 2002.

[41] Juraj Kojs, Stefania Serafin, and Chris Chafe. Cyberinstruments via physical modeling synthesis: Compositional applications. *Leonardo Music Journal*, 17:61–66, 2007.

[42] Stefan Bilbao and John ffitch. Prepared piano synthesis. In *Proc. of the* 9^{th} *Int. Conference on Digital Audio Effects (DAFx-06)*, pages 77–82, Montreal, Canada, September 18–20 2006.

[43] Giovanni De Poli. In search of new sounds. *Computer Music Journal*, 20(2):39–43, Summer 1996.

[44] David M. Howard and Stuart M. Rimell. Cymatic: A tactile controlled physical modelling instrument. In *Proc. of the* 6^{th} *Int. Conference on Digital Audio Effects (DAFx-03)*, London, UK, September 8–11 2003.

[45] Claude Cadoz and Jean Loup Florens. The physical model, modelisation and simulation systems of the instrumental universe. In Giovanni De Poli, Aldo Piccialli, and Curtis Roads, editors, *Representations of Musical Signals*, pages 227–268. MIT Press, Cambridge, 1991.

[46] Nicolas Castagne and Claude Cadoz. GENESIS: A friendly musician-oriented environment for mass-interaction physical modeling. In *Proc. of the 2002 International Computer Music Conference*, pages 330–337, 2002.

[47] Claude Cadoz. The physical model as metaphor for musical creation "pico..TERA", a piece entirely generated by physical model. In *Proc. of the 2002 International Computer Music Conference*, pages 305–312, 2002.

[48] Jean-Marie Adrien. The missing link: Modal synthesis. In Giovanni De Poli, Aldo Piccialli, and Curtis Roads, editors, *Representations of Musical Signals*, pages 269–298. MIT Press, Cambridge, 1991.

[49] Stefan Bilbao, Alberto Torin, Paul Graham, James Perry, and Gordon Delap. Modular physical modeling synthesis environments on gpu. In *Proc. of the 2014 International Computer Music Conference*, September 14–20 2014.

[50] Stefan Bilbao. Modular percussion synthesis environment sound examples. `http://www2.ph.ed.ac.uk/~sbilbao/modperc.html`, May 2014. Accessed 11 November, 2014.

[51] Stefan Bilbao. Modular percussion synthesis environment demo piece by Gordon Delap. `http://www2.ph.ed.ac.uk/~sbilbao/roughbounce.mp3`, May 2014. Accessed 11 November, 2014.

[52] Gordon Delap. Ashes to ashes. `http://www.ness-music.eu/music/ashes-to-ashes`, 2014. Accessed 20 January, 2015.

[53] NESS. Next generation sound synthesis project home page. `http://www.ness-music.eu/`, January 1 2012. Accessed 20 January, 2015.

[54] Rudolf Rabenstein, Stefan Petrausch, Augusto Sarti, Giovanni De Sanctis, Cumhur Erkut, and Marri Karjalainen. Block-based physical modeling. In *Proc. of the* 5^{th} *Int. Symp. on Mathematical Modelling (MATHMOD)*, 2006.

[55] Edgar Berdahl and Julius O. Smith III. An introduction to the Synth-A-Modeler compiler: Modular and open-source sound synthesis using physical models. In *Proc. of the Linux Audio Conference 2012 (LAC2012)*, CCRMA, Stanford University, CA, USA, April 12–15 2012.

[56] MATLAB. The language of technical computing. `http://www.mathworks.co.uk/products/matlab/?refresh=true`, 2014. Accessed 11 November, 2014.

[57] Scott Wilson, David Cottle, and Nick Collins, editors. *The SuperCollider Book*. The MIT Press, Cambridge, Massachusetts, 2011. ISBN 978-0-262-23269-2.

[58] Python. The Python language reference. `https://docs.python.org/2/reference/`, September 2014. Accessed 11 November, 2014.

[59] NumPy and SciPy. Numpy and Scipy documentation. `http://docs.scipy.org/doc/`, September 2014. Accessed 11 November, 2014.

[60] Csound. The csound community. http://www.csounds.com/, 2015. Accessed 11 August, 2015.

[61] Pure Data. Pd website. https://puredata.info/, 2015. Accessed 11 August, 2015.

[62] Web Audio API. Information about the Web Audio API. `http://webaudioapi.com/`, July 2015. Accessed 24 July, 2015.

[63] JSON. Introducing json. `http://json.org`, 2015. Accessed 24 July, 2015.

[64] pyo. Python digital signal processing module. `https://code.google.com/p/pyo/`, July 2015. Accessed 24 July, 2015.

[65] scipy.signal. Signal processing using SciPy. `http://docs.scipy.org/doc/scipy/reference/signal.html`, July 2015. Accessed 24 July, 2015.

[66] Julia. A high-level, high-performance dynamic programming language for technical computing. `http://julialang.org/`, 2015. Accessed 27 July, 2015.

[67] Atom. A hackable text editor for the 21st century. `https://atom.io/`, 2015. Accessed 29 July, 2015.

[68] Gerard Nierhaus. *Algorithmic Composition: Paradigms of Automated Music Generation*. Springer-Verlag/Wien, 2009. ISBN 978-3-211-77539-6.

[69] Gottfried Michael Koenig. Acousmatrix 1/2. CD, 1990. BVHaast.

[70] Gottfried Michael Koenig. The use of computer programmes in creating music. Unpublished writing, `http://www.koenigproject.nl/Computer_in_Creating_Music.pdf`, Accessed 11 November, 2014, 1970.

[71] Stockhausen. Elektronische Musik 1952–1960. CD, 1991. Stockhausen-Verlag.

[72] Karlheinz Stockhausen and Elaine Barkin. The concept of unity in electronic music. *Perspectives of New Music*, 1(1):39–48, Autumn 1962.

[73] Bruce L. Jacob. Algorithmic composition as a model of creativity. *Organised Sound*, 1(3):157–165, 1996.

[74] Simon Waters. Beyond the acousmatic: Hybrid tendencies in electroacoustic music. In Simon Emmerson, editor, *Music, Electronic Media and Culture*, pages 56–83. Ashgate, 2000. ISBN 0 7546 0109 9.

[75] Denis Smalley. Spectromorphology: Explaining sound-shapes. *Organised Sound*, 2(2):107–126, August 1997.

[76] Kees van den Doel, Paul G. Kry, and Dinesh K. Pai. Foleyautomatic: Physically-based sound effects for interactive simulation and animation. In *ACM SIGGRAPH 01 Conference Proceedings*, 12–17 August 2001.

[77] Denis Smalley. Defining timbre – refining timbre. *Contemporary Music Review*, 10(2):35–48, 1994.

[78] Jean-Claude Risset. Computer music: Why? Unpublished writing, `http://www.utexas.edu/cola/insts/france-ut/_files/pdf/resources/risset_2.pdf`, Accessed November 11, 2014.

[79] Max V. Mathews. The digital computer as a musical instrument. *Science*, 142 (3592):553–557, November 1 1963.

[80] John Chowning. Fifty years of computer music: Ideas of the past speak to the future. In 4^{th} *International Symposium, CMMR 2007*, Copenhagen, Denmark, August 2008.

[81] Jean-Claude Risset. Sud / Dialogues / Inharmonique / Mutations. CD, 1987. INA-GRM.

[82] Julian Anderson. A provisional history of spectral music. *Contemporary Music Review*, 19(2):7–22, 2000.

[83] Joshua Fineberg. Spectral music. *Contemporary Music Review*, 19(2):1–5, 2000.

[84] Barry Truax. Composing with time-shifted environmental sound. *Leonardo Music Journal*, 2(1):37–40, 1992.

[85] Simon Emmerson. Aural landscape: Musical space. *Organised Sound*, 3(2):135–140, 1999.

[86] Trevor Wishart. Sound symbols and landscapes. In Simon Emmerson, editor, *The Language of Electroacoustic Music*, pages 41–60. Palgrave Macmillan, 3 December 1986.

[87] Jean-Claude Risset. Modèles physiques et perception - Modèles physiques et composition. In *Modèles Physiques Création Musicale et Ordinateur*, volume III, pages 709–720. ACROE, Grenoble, ed. de la maison des sciences de l'homme edition, 1990.

[88] Trevor Wishart. *On Sonic Art*. Harwood Academic Publishers, 1996. ISBN 3-7186-5846-1.

[89] Denis Smalley. The listening imagination: Listening in the electroacoustic era. *Contemporary Music Review*, 13(2):77–107, 1996.

[90] John Young. Imagining the source: The interplay of realism and abstraction in electroacoustic music. *Contemporary Music Review*, 15(1):73–93, 1996.

[91] Gottfried Michael Koenig. Complex sounds. Unpublished writing, `http://www.koenigproject.nl/Complex_Sounds.pdf`, Accessed 11 November, 2014, 1965.

[92] Xavier Rodet. The CHANT project: From the synthesis of the singing voice to synthesis in general. *Computer Music Journal*, 8(3):15–31, Autumn 1984.

[93] Trevor Wishart. *Audible Design: A Plain and Easy Introduction to Practical Sound Composition*. Orpheus the Pantomime Ltd, 1994. ISBN 978-0-9510313-1-5.

[94] Gary S. Kendall. The decorrelation of audio signals and its impact on spatial imagery. *Computer Music Journal*, 19(4):71–87, Winter 1995.

[95] Horace Vaggione. Composing musical spaces by means of decorrelation of audio signals. In *Add. of the COST G-6 Conference on Digital Audio Effects (DAFx-01)*, pages 1–8, Limerick, Ireland, December 6–8 2001.

[96] Denis Smalley. Space-form and the acousmatic image. *Organised Sound*, 12(1): 35–58, April 2007.

[97] Maarten van Walstijn, Michael Alcorn, and Stefan Bilbao. Spatialisation using sounding objects. In *Proceedings of the 2005 International Computer Music Conference*, Barcelona, Spain, 2005.

[98] Miguel Cerdeira Negrão. ImmLib - a new library for immersive spatial composition. In *Proc. of the 2014 International Computer Music Conference*, Athens, Greece, September 14–20 2014.

[99] Manuella Blackburn. Composing from spectromorphological vocabulary: Proposed application, pedagogy and metadata. In *EMS09 Proceedings*, pages 1–7, Buenos Aires, Argentina, June 22–25 2009.

[100] Manuella Blackburn. The visual sound-shapes of spectromorphology: An illustrative guide to composition. *Organised Sound*, 16(1):5–13, 2011.

[101] Jonty Harrison. Articles Indéfinis. CD, 1992. Empreintes Digitales.

[102] Robert Erickson. *Sound Structure in Music*. University of California Press, 1975. ISBN 0-520-02376-5.

[103] Joanna Demers. *Listening through the Noise.* Oxford University Press, 2010. ISBN 978-0-195-38766-7.

[104] Frank J. Oteri. Pauline Oliveros: Creating, performing and listening. `http://www.newmusicbox.org/articles/pauline-oliveros-creating-performing-and-listening/`, 1 December 2000. Accessed 6 January, 2015.

[105] Monty Adkins. Metaphor, abstraction and temporality in electroacoustic music. In *ArtMusFair 2010*, Chopin University, Warsaw, Poland, September 22–26 2010.

[106] Ian Nagoski. La Monte Young & Marian Zazeela: An interview by Ian Nagoski. `http://www.halana.com/lymz.html`, November 12 2014. Accessed 11 November, 2014.

[107] Eluvium. Static Nocturne. CD, 2010. Watership Sounds.

[108] Brian Eno. Ambient music. In Christopher Cox and Daniel Warner, editors, *Audio Culture: Readings in Modern Music*, pages 94–97. Continuum, 2008. ISBN 0-8264-1615-2.

[109] Gas. Königsforst. CD, 1998. Mille Plateaux.

[110] William Basinski. The disintegration loops. CD, 2012. Temporary Residence Limited.

[111] Erik Nyström. Texture and entropic processes in electroacoustic music. In *Proc. of Sound, Sight, Space and Play 2009, Postgraduate Symposium for the Creative Sonic Arts*, pages 45–58.

[112] Stephan Mathieu. The Sad Mac. CD, 2004. Headz.

[113] Stephan Mathieu. The Falling Rocket. digital download, FLAC, 2013. Schwebung.

[114] Stephan Mathieu and Caro Mikalef. Radioland (Panorámica). CD, 2012. Line.

[115] Pierre Boulez. *Orientations.* Harvard University Press, 1986. ISBN 978-0-674-64376-5.

[116] Leigh Landy. The "something to hold on to factor" in timbral composition. *Contemporary Music Review*, 10(2):49–60, 1994.

[117] Julius O. Smith III. Introduction to digital filters with audio applications. online book, September 2014. `https://ccrma.stanford.edu/~jos/filters/`.

[118] Wilfred Kaplan. *Advanced Calculus*. Addison Wesley, 5th edition, 2002. ISBN 978-0-201-79937-8.

[119] Julius O. Smith III. Physical audio signal processing. online book, September 2014. `https://ccrma.stanford.edu/~jos/pasp/`.