UNIVERSITY OF EDINBURGH

DOCTORAL THESIS

The New Pulsar Generator (nuPG): Compositional Practice, Digital Sound Synthesis Model and Their Temporalities

---

Author:
Marcin PIETRUSZEWSKI

Supervisor:
Florian HECKER
Martin PARKER

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

The Reid School of Music
Edinburgh College of Art

September 12, 2023

Declaration of Authorship

I, Marcin PIETRUSZEWSKI, declare that this thesis titled, "The New Pulsar Generator (nuPG): Compositional Practice, Digital Sound Synthesis Model and Their Temporalities" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSITY OF EDINBURGH

Abstract

The Reid School of Music
Edinburgh College of Art

Doctor of Philosophy

The New Pulsar Generator (nuPG): Compositional Practice, Digital Sound Synthesis Model and Their Temporalities

by Marcin PIETRUSZEWSKI

The thesis examines compositional methods emerging from within design and creative engagement with the New Pulsar Generator (nuPG) program. Following a hybrid methodology joining sound composition and media archaeology, it provides a systematic look into the creative practice with historically inherited computational and artistic materials. To do so, it charts a trajectory of the compositional process, sound theoretical model, programming paradigm, and associated formalisms in the computer program's operative realm. The thesis has three broad goals: to outline an integrative approach to compositional practice which incorporates engagement with digital materials, their design, genealogy, and creative application; to examine and contextualise a portfolio of new compositions; and to introduce a new computer program for sound synthesis, the New Pulsar Generator (nuPG). The project thus combines and interweaves a compositional practice, media archaeology and computer program design. The relations and overlaps across these areas are emphasised throughout the thesis text, and connections among them are foregrounded. The project's overarching aim is to mobilise such exchanges and to establish them as an operative concern of contemporary computer music practice.

Acknowledgements

I want to thank my supervisors, Florian Hecker and Martin Parker, for their consistent support and guidance during this project. Furthermore, this work would not have been accomplished without the support and insights I have received from various patient advisors throughout this research. My gratitude to Curtis Roads, Marcus Schmickler, Peter Nelson, Tom Mudd, Raymond MacDonald and Michael Edwards. The work on this research project would be a lonely affair without the community which encouraged and supported it. In addition to the above, my thanks and love to Lauren Sarah Hayes, Jules Rawlinson, Alberto de Campo, Ian M Fraser, Ben Kudler and other users of the New Pulsar Generator program. Finally, thanks and big love to my supportive family for all the love and encouragement: Bozena Kwiatkowska and Marek Kwiatkowski.

Contents

List of Figures

List of Audio Examples

List of Code Examples

List of Abbreviations

nuPG	**N**ew/**N**eu/ **Pulsar G**enerator
PG (2001)	**P**ulsar **G**enerator program by Curtis Roads and Alberto de Campo
PG (2004)	**P**ulsar **G**enerator program by Tommi Keränen

For/Dedicated to/To my…

Introduction

0.1 Preamble

I first encountered pulsar synthesis as a young composer attending a concert by the German composer, Marcus Schmickler. Schmickler played an 8-channel set from the 'Altars of Science' (Schmickler, 2007) album at the Atrium of the Reid School of Music (Edinburgh University) on the 3rd of March 2009. The setting of the concert, a black box venue, the multichannel surround system, and the inhuman synthetic timbres flowing from loudspeakers all contributed to an unforgettable aesthetic experience. Material generated with pulsar synthesis featured along with other techniques, including stochastic synthesis and different forms of granular synthesis. The experience was formative for me as a composer and future researcher. I implemented pulsar synthesis in SuperCollider 3.5 in 2013 as part of the final project research and composition within the MSc in Digital Composition and Performance at Edinburgh University. The composition 'Sheaf Navigation' incorporated pulsar synthesis material along with processed violin and double bass[1]. Processes of working on the implementation of the program and composing with it prompted me to explore the richness of contexts activated through the technique of pulsar synthesis. I was drawn to its compositional possibilities as well as technological and historical discussions that enabled me to place the technique at the intersection of modern and emerging computer music. My research project originated from the impulse to deepen and renew interest in pulsar synthesis, on the one hand as a historically inherited material technology, and on the other hand as a potential new voice in the development of computer music techniques. Presented here: a portfolio of compositions, a computer program and the thesis, constitute a result of an extended period of research and practice. The result of the project, the New Pulsar Generator (nuPG) program, is a point of emergence for a series of compositional works, sound installations and performances. A critical analysis of the program's design was a source of discussion presented in this thesis and a generating principle within presented sound works.

0.2 Introduction

The thesis traces the compositional process through an interrogation of a singular digital sound synthesis technique called pulsar synthesis. Within such an engagement, computer programs, programming languages, the interface and the sound theoretical model to which it connects become anchors of the compositional practice. The literature available on the topic of pulsar synthesis and its implementations treats either the interface (program manual, tutorial) or the underlying sound-theoretical model exclusively - these are cases of Curtis Roads *Microsound* and "Sound

---

[1] Composition premiered as a live multichannel performance at Inspace, Informatics Forum, Edinburgh University on 26th of August, 2013 `https://vimeo.com/76063356?embedded=true&source=video_title&owner=16586819` (Accessed 11.09.2023)

composition with pulsars". An example of extended discussion on the topic includes (Haworth, 2015), which discusses the technique in the context of the non-standard synthesis models (such as Iannis Xenakis' Dynamic Stochastic Synthesis and UPIC) and practices of contemporary computer music composers such as Russell Haswell and Florian Hecker. While the first strand of texts focuses solely on the technical aspect of the subject, the second opens it up to include the social and musicological-comparative discussion. These bodies of texts constitute an exciting proposal for compositional practice and research projects drawing simultaneously from media theoretic methodologies and contemporary computer music practice. A position developed throughout the project does not consider media archaeology and composition as converging fields. What seems like an exciting prospect is positioning these as asymmetrical and often conflicting discourses and practices.

The thesis combines what often remains contained in separate fields of practice and study; sound and media theories, musicological research, and compositional method. A proposed compositional analysis is rooted on the one hand in the development of computer program with all its elements, including interface design and sound programming, and on the other hand, in the contextual and comparative methods brought by contemporary media archaeology studies. A composer Claudia Molitor in a conversation with Thor Magnusson highlights a deep entanglement of technology—understood in broad terms including digital tools such as Digital Audio Workstation (DAW) but also a paper and pencil—and compositional methods (Molitor and Magnusson, 2021). This convergence between the tools, methods and discursive contexts has been central to my compositional work. The computer programs and their building blocks, theoretical, technological and aesthetic texts encountered through the process of research and design can all be elements of creative engagement. Within the work with the pulsar synthesis, I assume them as indispensable to the compositional praxis itself and the final sounding outputs as recordings, sound installations and performances. A vital trait of the work presented here comes from a view of a computer program as a dynamic and operational medium. Computer programs for sound composition are not only functional devices but also influential within the creative process. They are not simple tools, receivers of human input and generators of audio output; they are also, in an expanded sense of the notion of the artefact that I argue for, active agents in the process of composition, a mediator between the user, sound-theoretical model, modes of computational, textual and graphic articulations of sound, techniques, and aesthetics. The thesis traces this convergence through an investigation of the nuPG computer program as a generator of sensual and informational presence.

Historical sound synthesis techniques have gained increasing attention in recent years. Stefan Goldman looked at FM synthesis and its instrumental implementation Yamaha DX7 as a determinant in the emergence of techno music (Goldmann, 2015). Aziz Ege Gonul has developed an exact clone of the synthesiser in SuperCollider programming language[2]. The cloned version included porting all original patches - all 16384. Documentation of the project also consists of a comparative analysis of the audio outputs between clone and original hardware versions (Gonul, 2016). In my approach to working with pulsar synthesis, I have been particularly inspired by the research on the reimplementation of Iannis Xenakis' Dynamic Stochastic Synthesis (Gendy) technique. Particularly helpful was the writing and programming of Peter Hoffmann who worked on a new implementation of Xenakis Gendy (Hoffmann,

---

[2] (`https://github.com/everythingwillbetakenaway/DX7-Supercollider`

2009) and Sergio Luque who developed an extended model including stochastic concatenation of dynamic stochastic synthesis (Luque, 2009; Luque, 2009). The design of the original Pulsar Generator (2001) program by Curtis Road and Alberto de Campo is linked to a shift in the accessibility of non-standard sound synthesis techniques. Written in SuperCollider 2 programming language and running on MacOS 9 computers the program enabled individual users of personal computers to access the technique. The program became obsolete shortly after its premiere due to Apple shifting to a new MacOS X operating system architecture in March 2001. However, the work developed by me is not simply an updated version of the nuPG. The model of the original program was a starting point. Through systematic experimentation, extension and augmentation of the model, I developed not only a new implementation of the technique of pulsar synthesis— the nuPG program — but also a set of new compositional methods and hybrid approaches mixing pulsar synthesis with other digital sound techniques. Tracing the link between the program's source code, synthesis parameters, and the graphic interface helped me to understand the complex relationship between the program's structure and what is audible. Computer programs for composition and sound synthesis are often thought of as "black boxes" - functional tools fulfilling a particular role. Within this view, the user is limited to an interface designed in advance which highlights only selected parameters of the synthesis process. Such a simplification is necessary as a need for a concise product. However, it can be an obstacle hiding deeper coding architecture where the synthetic process is formalised. Within my work with pulsar synthesis, the fixed nature of such solidification became the first site of modification. An important aspect of the project was to expose coding architecture, the working of the code, and specific programming choices, upon which the code functions, outputs, processes, and represents. This process of opening up the program became a central theme of this research project. The thesis traces the consequences of this "opening up" for the emergence of new conceptions of musical form, representation of compositional data and procedures. The key to my work with the program was creative engagement with both computational and aesthetic ramifications of the technique of pulsar synthesis. The creative activity in this context extends beyond the production of compositions and includes designing the computer program its sound synthesis model and an interface.

I propose a media archaeological discourse doubly mediated through composition and computer programming practices as a view synthesising complex relationships emerging between creative processes, technology and their objects. The work with pulsar synthesis cannot be reduced to the 'historic' moment when the technique became explicit and applied. What is at stake here is the media-epistemic temporal field of the technique opened up through its operationalisation here and now. The thesis does not present an all-encompassing and chronological view of the history of digital synthesis. Instead, it attempts to recapture productive dynamics, filiations and conjectures of the creative process with the technique of pulsar synthesis, its new instrumental encapsulation, nuPG and script-base approaches. The first source of reference is media archaeology as developed by Wolfgang Ernst[3]. Media archaeology, as presented in *Chronopoetics*, fits perfectly into the dual aspect of the computer program as an operable and analysable object. The operational approach is inextricably linked with a hands-on approach that seeks to try, unbox, and feel the object. The hands-on approach allows the user to experiment with tactile contact, touch,

---

[3] It is worth pointing out that media archaeology consists of a broad spectrum of scholarly practices. For other examples see: (Huhtamo and Parikka, 2011; Parikka, 2013; Zielinski, 1996)

and open media to get a more refined understanding of its processes. Mark Fell calls this approach a ′"reverse practice", the "analytic deconstruction of equipment in order to extract knowledge". The technology in Fell's account "takes an active role in the construction of musical formations that may otherwise be unimaginable" (Fell, 2022, p.106).

Media in digital form operate on a sub-phenomenal level we cannot perceive their principles of operation. The functioning of software enables the formation of a circuit between the user and theoretical work. Media archaeology always starts from the "close reading" of the technologies involved. In this respect, its method is close to critical media philology and platform and code studies. The thesis approaches this issue as a subject of archaeological reading of the program's interface and source code and focuses on sound theory and its computational realisation. The method of archaeology traces the formal and material aspects of the compositional practice by disassembling the complexities of a computer program. The computer program is seen as the dispositive of formal and material knowledge.

A key characteristic of this research is the simultaneous constitution of a computer program, its objects and design, and the compositional practice manifested in its outputs - recordings, sound installations and performances. Computer program design and compositional work appear as an iterative back-and-forth process. This characteristic turns questions regarding the roots of particular technologies and their archaeology into more than just a historical review - contextualisation becomes operational by incorporating fragments of context as materials in the process of computer programming or composition. Such contextualisation acts upon extracts from excavated contexts and mobilises their elements as active materials. In this way of thinking of composition, the focus shifts from self-contained work to engagement with materials - computer scripts, programming paradigms, sound theories and computational methods. Secondly, it provides a view of the emergence of compositional work from within the program's structure or as Claudia Molitor proposes the computer program becomes a carrier of "a compositional proposition" (Molitor and Magnusson, 2021). In line with the proposed above view of "opening up" of the program, this research involves procedures of recovering and uncovering contexts of pulsar synthesis as productive engagement with material and an integral component of the new compositional approach. Such a position turns contextualisation beyond simple commentary and approaches it as a source of constructive engagement.

An important effect of the pulsar synthesis technique on the compositional process is the incorporation of multiple time regimes from micro through meso- to macro-temporal. In the discussion on the workings of the pulsar synthesis, a collection of concepts such as real-time, iterations, recursions and loops might be more informative and meaningful than established music theory vocabulary. The nuPG program in line with its predecessor Pulsar Generator (2001) solves its main parameters in the time domain. A view of a horizontal line of values from left to right is a fundamental data representation in the program's design. It can also be considered as a primitive formal unit of the compositional process emerging from within the program. This finding prompted me to look at temporality in relation to the work with pulsar synthesis in a broader sense. Using pulsar synthesis as a creative tool for time understanding, this thesis and associated sound work explore how the method relates to time in a specific way. Central in this context is an object of the loop as conceived within music and computational contexts.

The domain of this thesis is the practice of computer music composition. The work presented here originates from an underlying assumption that composition with digital materials inevitably involves engagement with computer program design, issues of sound representation as data, and theoretical contexts from computation through sound theory to aesthetics. Such an engagement can be seen as a limitation—where a user accepts or does not consider it relevant to question these contexts— or a potentiality—where the very fabric and inner materials of a computer program become operative and available for experimental exploration. This research's tenet is treating the computer program as a potentiality. Makis Solomos observes in the context of granular synthesis that what is at stake is not the technology itself: "That is why its logic, whilst being linked to a technology (a type of sound synthesis), is more than a technology: it relies on a way of thinking music; one might speak of a granular paradigm or sensitivity" (Solomos, 2019, p.190). I aim to explore the sensitivity proposed by pulsar synthesis.

0.3 Overview

Key chapters of the thesis are preceded by a discussion on the objectives of the research and its methodology.

The central part of the thesis is divided into two parts. Part I discusses the work with pulsar synthesis from broadly construed but overlapping perspectives: audio signal representation, programming language design, graphic interface and computational formalism.

The chapter 1 begins by redefining the technique of pulsar synthesis from three interconnected perspectives: diagrammatic, sound theoretical, and computational. Established texts focus primarily on the sound theoretical basis of definition. To include diagrammatic (representation of sound as data) and computational considerations proposes an extended perspective on the emergence of the technique. To identify a connection between diagrammatic representation, the sound theoretical model and computational processes play an essential role in developing later in the thesis notion of the form. Having established the fundamental definition, the analysis turns into a historical example of pulsar synthesis, the Pulsar Generator program. The analysis developed in the thesis starts with an overview of the original program's functions. The basic concept of the interface is introduced.

The next chapter expands the field of analysis by looking at the SuperCollider programming language and object-oriented paradigm as functional for the emergence of the instrumental incarnations of the pulsar synthesis technique. A short introduction to the conversational model of programming proposes an alternative view on the definition of algorithms and computational processes. The final chapter of Part I synthesises prior discussion under a notion of the form.

Part II expands the findings of the archaeological stage into a set of compositional methods. Three compositional studies are presented: 1) Sieves, 2) Wavelets, and 3) Speculative Sonification. These studies, although separate from each other, are joined through a shared method: a conversational approach to programming. The compositional study 1) is preceded by an aesthetic and technical introduction to sieves. A set of methods applying sieves within the new pulsar generator program is presented. Where necessary, the method consists of a definition, a code, and an audio example. The second compositional study focuses on wavelets. The study is preceded by an aesthetic and technical introduction. The set of examples, code and audio is provided as a demonstration of the application of wavelets within various aspects of the new pulsar generator program. The last compositional study (Speculative Sonification) builds upon the extra-musical context of pulsar synthesis, i.e.

radio astronomy data sonification. Similarly to previous studies, this study includes aesthetic and technical background before introducing the compositional work.

Closing the thesis, Conclusions summarise the key findings of the research project and point at possible further work.

The first Appendix to the thesis, 'The New Pulsar Generator User Manual,' provides step-by-step information on installing and running the program. The manual explains all functions of the program embedded in the graphic interface, and these are provided through a text-based extension. The second Appendix consists of a printout of all 2048 values of the table. The printout is a fundamental representation of a compositional object within the New Pulsar Generator program. Appendix three provides additional information to an analysis presented on page 63. Appendix Four provides information on how to run the nuPG program from the script. Appendix Five lists all performances, sound installations and releases conceived using the nuPG program.

0.4 Portfolio of Works and Sound Examples

The portfolio of works with the nuPG program is an integral part of the submission. The works are located in the folder 'Portfolio of Compositions'. The set of works is a selection from a large output connected to my practice with the nuPG program but comprises a wide range of approaches engaging material and formal aspects of the synthesis technique. A list of all outputs (sound installations, performances and releases) conceived with the nuPG program can be found in Appendix Five. A large fraction of the portfolio works have been released as part of two publications:

1. *The New Pulsar Generator Recordings Volume 1* (2020) released by fancyyyyy records as a CD[4]. The release is accompanied by a rendition of an article *Sound Composition with Pulsars* by Curtis Roads (Roads, 2020). The new layout of the article with converted graphics is designed by Joe Gilmore[5].

2. *Auditory Sieve* (2020) released by ETAT[6]. The release is accompanied by text *Auditory Sieve: A protocol for pendular transitions between temporal resolutions*

Two additional works 'Synthetic Pulsar' (2021) and 'nuPG live@sonicActs (08.10.2022)' [7] are included in the portfolio.

There is no pre-defined order for the collection to be played back. However, the text of the thesis gently guides the reader in the listening process, pointing to particular works or excerpts to be listened to in the context of specific discussions. The works should be heard on a stereo loudspeaker setup at a moderate loudness level.

Apart from the portfolio of works, the text is accompanied by short sound examples illustrating the compositional methods discussed. The examples are located in the folder 'Sound Example Files'.

---

[4] `https://fancyyyyy.bandcamp.com/album/the-new-pulsar-generator-recordings-volume-1` (Accessed 11.09.2023)

[5] `https://qubik.com/marcin_pietruszewski/` (Accessed 11.09.2023)

[6] `https://etat.xyz/release/AuditorySieve` (Accessed 11.09.2023)

[7] The work is an outtake of a live performance presented initially in October 2022 at Sonic Acts Festival in Amsterdam: `https://www.sonicacts.com/biennial/biennial-2022/programme/inner-earth/marcin-pietruszewski` (Accessed 11.09.2023)

Located in subdirectories ’nuPG gendyApproximations’, ’nuPG voxApproximations’ and ’nuPG sieves’ are short sound studies focused on pulsar synthesis, aural approximations of GENDY algorithm, synthetic speech and application of sieves.

0.5 Computer Program: Standalone and Scripts

The submission is accompanied by the nuPG program. Standalone versions and source code is located in the folder ’nuPG program’. Three versions of the standalone are available, each located in the folder named to the corresponding system requirements. To run the standalone:

1. choose the right version for your OS and hardware build (M2, M1 or Intel)
2. copy and paste the ’newPulsarGenerator’ application to your Desktop. **This is important, as the program often doesn’t function when placed in the other directory**
3. double click the ”newPulsarGenerator’ application icon. You should see the loading window and, in the end, the GUI as seen on the software manual (A.1)
4. follow the user manual (Appendix A) for details on how to use the program

The folder contains the source code necessary to run the program as a script from within SuperCollider. All necessary files and the installation guide can be found in the folder ’Run as Script’. The installation guide for the scripted version can also be found in Appendix D. Extended ’text-based’ functions of the nuPG program are only available through the scripted version. While running from the script test ’nupg_fx.scd’ Additionally, a screen recording of the program in action is provided as a reference: see ’nuPG_Screen Recording.mov’.

Objectives

Proposed studies aim to interrogate the artistic practice that has formed throughout this research from the perspective of the computer program to reveal the dispositions towards particular aesthetics, the conceptual contexts and the compositional methods that underlie the compositional work. As such, the study proposes an integrative approach to reconsider how the computer program mobilises a compositional process and, through its mediation between discursive and technological contexts, deepens the knowledge of and extends computer music techniques.

A central issue for the composer of computer music, the designer of digital media or the scholar of digital media is how to engage with the computational processes. The project aims to trace the operational commitments in the structure of the program, its source code, modes of data representation, interface, and other elements that determine its functionality. This aim is not simply an analytic objective but a prerequisite for a compositional practice that seeks to account for a complex set of sources contributing to its method.

A kernel of this research is an orientation that attributes significance to the computer program as a conversational voice and a window into aesthetic and discursive contexts. In broad terms, it is a sonic, technological and theoretical project of attention to the singular synthetic sound technique, its computer program encapsulation, and its function in the sound compositional process. The research concentrates on a set of questions probing the method of pulsar synthesis as a specific engagement with questions regarding temporality in its aesthetic, sound theoretical, technological and discursive contexts:

- How far does the technique of pulsar synthesis and its instrumental implementation — the nuPG — involve temporality in a specific form, and how can compositional practice with the technique offer a distinctive way of understanding musical time?

- To what extent can engagement with a computer program functional today participate in an archaeological process dealing with multiple temporalities understood in their technical, sound theoretical and historical contexts?

- Can an engagement with the computer program in its technical, aesthetic and sensory dimensions renew the discussion on the relationship between musical material and form?

- What relationship does a digital sound synthesis technique afford with its past and present aesthetic and theoretical contexts?

- Is there a creative opportunity, both sonic and structural, presented by the computer program, its design, and its practice?

An engagement with these questions takes the form of two interrelated sets of aims and objectives:

1. Objective 1: A development of a standalone computer program for pulsar synthesis - the nuPG - which incorporates visual, textual and formalised modes of control over synthesis parameters. The objective includes practising computer programming - using SuperCollider programming language - and studying specific programming contexts for sound. The study aims at:
    (a) analysis of the sound theoretical model embedded in the technique of pulsar synthesis;
    (b) comparative analysis of the nuPG program and selected historical instrumental incarnations of the pulsar synthesis technique;
    (c) broadening the definitional scope of what pulsar synthesis might be. This is by considering the relationship between the interface of the program, its source code, and its underlying sound theoretical model;
    (d) interrogating a relationship between programming language and modes of representation of data, objects and processes with particular interest to the design of computer programs for sound synthesis and composition;
    (e) mobilising a deeper understanding and an operative engagement with a computer program as a system of interconnected objects participating in the process of compositional material forming;
2. Objective 2: A development of compositional approaches that embrace an orientation that attributes significance to the computer program as a conversational voice, a source of compositional ideas and an access point into discursive, aesthetic and technological contexts. This objective functions as an extension to Objective 1 and develop as a case study focused on particular aspects of the New Pulsar Generator program and its compositional contexts. The set of compositional methods aims at the following:
    (a) testing applicability of pulsar synthesis specific objects (e.g., the loop) in conceptualising compositional practice;
    (b) operationalisation of notions of musical material and form from a specific point of a digital medium;
    (c) revealing a relationship between compositional methods and the design of the computer program;
    (d) extending the formal and material potential of pulsar synthesis through the incorporation of techniques of sieves and wavelet analysis-resynthesis;

The respective scope of these objectives can be summarised under the following headings **Computer Program Design** (Objective 1) and **Composition** (Objective 2). These objectives, however, should not be considered as separate units of work in a successive order of execution; but more like nodes of a continuous and iterative process. By transforming and translating computational and compositional objects, this process involves a vicarious relationship. As Graham Harman writes, such a relationship entails "that forms do not touch one another directly, but somehow melt, fuse, and decompress in a shared common space from which all are partly absent" (Harman, 2013, p. 142). Similarly positioned perspective has been presented by composer Mark Fell in his book *Structure and Synthesis: The Anatomy of Practice*. Fell proposes a notion of. "material attunement" to designate a close relationship between tool-making and its use. In Fell's view, these modalities are equal and come into a mediatory relation within compositional practice (Fell, 2022). The creative process as renewed and dynamic *attunement* to the digital material of the nuPG program is one of this research's key motivations.

Functional to Objectives (1) and (2) is a practice viewed as folding together musical tools for artistic creation with tools and computational techniques. An approach realised by me aims at merging these practices, intensifying their exchange and bifurcations. Consequently, the role of contextualisation within this project is to create not only discursive links between software-based and compositional processes - a commentary - but also to establish operational strategies mobilising practical outcomes - a context objectified. Such a view on the research objectives was elemental for the methodological framework described in the next chapter (0.5).

Methodology

The section introduces the methodology developed and applied within this research project. The methodology combines composition practice with media archaeological approaches engaging material, technological and discursive contexts of compositional and computational processes.

An imperative of this project from the outset was to develop a methodological approach capable of bringing together practices of composing and programming. Thus, the method of media archaeologically informed composition designed within this project functions in multiple forms. First, as an analytical tool employed in writing on compositional practice, it recognises diverse conditions of the emergence of musical work and accounts for the operative role of technology and broader discursive contexts. Second, as a creative strategy, actively participating in the process of composing new works, guiding an engagement with diverse materials (i.e., computer programs, programming paradigms, interfaces, representations of musical objects, temporalities and processes). Such an interlaced nature of writing and artistic practice alters the role of the text. As proposed by Ross Gibson, within artistic research, "the text is an explicit, word-specific representation of processes that occur during the iterative art-making routine" (Gibson, 2010, p. 16). The text is not only a commentary but an active participant in gradual, cyclical speculation and the realisation of compositional works. This view echoes the role of writing in the experimental process as described by Hans-Jorg Rheinberger: "Writing up, tracing, sketching, is part and parcel of the experimental process" (Rheinberger, 2013, p.203). In the context of this research, the writing process accompanies and guides the processes of computer program design and creation. Writing, as engaging with the contextual extent of the work, feeds back into that work as productive proposals, conjectures and diversions projecting the work into the future. In return, the text bears traces of the experimental "goings-on" of the compositional process. The act of writing plays an essential role in Rheinberger's notion of the experimental system. Within current research, the model of the experimental system enables exchange between practices of composition, computer program design and media archaeological analysis. Consequently, the experimental system complicates a linear progression between categories of analysis and synthesis. Rather than one simply preceding the other, the strategy proposes to view their relationship as a pendular back and forth - potentially endless - oscillation. This circularity is a crucial feature of the overall proposed methodology. From this perspective, the proposed methodological couplet of media archaeology and composition is positioned similarly to an iterative feedback process described in Smith (2009).

The chapter starts with an introduction to the archaeological media method applied to the research. The process draws upon the writing on the temporality of digital media as proposed by Wolfgang Ernst. Explicated across a series of texts (Ernst, 2009; Ernst, 2013; Ernst, 2016a; Ernst, 2016b; Ernst, 2019; Ernst, 2020a) the method allows to grasp the digital object first of all in its operative state, as a generator of unique temporalities. In a concrete sense, archaeology can be a process of extracting singularities from a complex field. Extraction as an activity signifies drawing out, withdrawing, taking or getting out, pulling out or removing from a fixed position, literally or figuratively. The image of pulling out and mobilising things, whether discursive or practical, was pertinent in this work's context. Within current research, such extraction fields include programming language, digital instrument design and the sound synthesis model. Concretely, the method of archaeology takes the form of an engagement with selected developments in the field of digital instrument design (various incarnations of pulsar synthesis, but also related developments such as UPIC, graphic synthesis or non-standard synthesis approaches), compositions with digital instruments, and fundamental compositional and aesthetic discourses activated within such delineated practices. This method also manifests itself in the expansion of the field. It is characterised by finding connections between particularities of compositional qualities of the material and the computational realm of the digital medium (design choices, forms of representation of musical data, encapsulation – instrumentalisation – of the sound theory etc.).

The method of archaeology can be summarised as analytic in a broad sense. The next part of the section introduces its synthetic extension: the experimental system. Hans-Jörg Rheinberger has developed the theory of the experimental system in the context of empirical sciences, particularly molecular biology. However, the scientific and artistic domains are not homologous. Rheinberger himself has opened a possibility for the application of his model outside of the original field[8]. Recently, Paulo de Assis (Assis, 2018) and Michael Schwab (Schwab, 2013; Schwab, 2014) developed an in-depth analysis of Rheinberger's project and its applicability in the context of artistic research. I propose extending the experimental system's use to include composition practice with historically inherited computational materials.

In the context of computer and electroacoustic music, the media archaeological approaches can be seen in the works of Florian Hecker, Marcus Schmickler and Yasunao Tone. On the CD release 'Triadex Muse Treks' (2012) Florian Hecker worked with the sequencer-based digital synthesiser 'Triadex Muse' by Edward Fredkin and Marvin Minsky. The synthesiser produced a 4-voice sound and provided control of volume, tempo (eight steps from 54 to 1,662 bpm), coarse pitch (32Hz to 4.5kHz) and fine pitch (±10 per cent of the fundamental frequency), intervals, and musical themes. The output was produced through a 4" built-in speaker[9]. Marcus Schmickler explored the potential of historical sound equipment from the SWR studio in Freiburg, Germany, In the work 'Sky Dice. Mapping the Studio' (2021), Schmickler utilised ARP 2500, Publison DHM89B, Publison Infernal Machine, and computer[10]. In the work, Schmickler utilises the studio as a model for the sonic depiction of a historical signal path. It is a work created not only with the hardware of the studio but also about the hardware and the studio itself. This approach resonates closely with the way I developed my work with the nuPG program. The media-specific approach is also characteristic of the work of Yasunao Tone. The work 'Solo for Wounded CD'

---

[8] One such field, mentioned by Rheinberger, is writing. This has to do with his conception of scientific objects as complex "bundles of inscriptions" (Rheinberger, 1997) and the concept of "inscription" as developed by Jacques Derrida in his work "On Grammatology" (*De la grammatologie*), which Rheinberger translated into German together with Hanns Zischler in 1983

[9] see the Triadex Muse programming guide: `https://web.archive.org/web/20110613194435/http://trovar.com/muse/book2.html` and the manual: `https://web.archive.org/web/20110613193225/http://trovar.com/muse/book.html`. Both websites are accessible via WaybackMachine, last accessed: 11.09.2023

[10] The work has been premiered as 11.2-channel performance at Donaueschinger Tage fur Neue Musik on the 20th of October 2018 and later released as an album by Editions Mego: `https://www.discogs.com/release/19675777-Marcus-Schmickler-Sky-Dice-Mapping-The-Studio`. (Accessed 11.09.2023)

utilises CD players (Sharp DX-100) and prepared CDs. By placing carefully sized and arranged pieces of matte cellophane tape on the playing surface of CDs. Tone creates "short frozen loops of sound, elisions and ellipses as the playing mechanism skips over entire tracks of corrupted data" (DeMarinis, 2011, p.229). Tone insisted on making audible mechanisms of data correction built into the CD players; making them operational. My work with the nuPG program mobilises primitive objects of the technique of pulsar synthesis as productive strategies for the formal development of sound.

Media Archaeology and Computer Music Composition

Important for the methodology of current research is the work of Wolfgang Ernst and its renewed perspective on the materiality of media technology. Unfolding across a rich body of texts (Ernst, 2011; Ernst, 2013; Ernst, 2016a; Ernst, 2016b; Ernst, 2018), Ernst's media archaeological perspective questions the temporal presence of media and their relationship with the history[11]. One of the significant implications of this perspective is a shift from media analysis focused on "textual interpretation" toward "diagrammatic reading of circuit plans and material hermeneutics (media-archaeological philology)" essentially operational and concerned with "dis- and re-assembling" of media (Emerson, 2013). As observed by Ernst, "media archaeology understands the arche of computing not only in its sense of the calculus as algorithmic operation but as its actual implementation" (Ernst, 2020a, p.4). The operative aspect of Ernst's media archaeological method fits into the creative process integrative to the work presented within this research. The computer program reveals its temporal essence in operation; it needs to produce an effect. Ernst elaborates on the notion of operativity in comparison to objects of classical archaeology:

> [W]hat drastically separates an archaeological object from a technical artefact is that the latter discloses its essence only when operating. While a Greek vase can be interpreted as simply being looked at, radio or computer does not reveal its essence by monumentally being there but only when processed by electromagnetic waves or calculating processes. If a radio from a museum collection is reactivated to play broadcast channels of the present, it changes its status: it is not a historical object anymore but actively generates sensual and informational presence (Ernst, 2011, p.58)

It can be argued that for example, a bone flute needs to at least imply operation for it to impart meaning. The above statement nonetheless holds relevance in terms of distinguishing the method of media archaeology as dealing with its objects in a double way. First, the method is an analytic model for distilling critical principles of objects' constitutions. Second, it is a synthetic process mobilising the operable capabilities of those objects. Existing technologies are understood here not as merely

---

[11] In 2003 Wolfgang Ernst created Media-Archeological Fundus (`https://www.musikundmedien.hu-berlin.de/de/medienwissenschaft/medientheorien/fundus/media-archaeological-fundus`), a collection of obsolete historical media technologies, and the Signal Laboratory (`https://whatisamedialab.com/tag/wolfgang-ernst/`), which centred around study of computational hardware and software. Both projects emphasised a need to open media technologies to understand their technical operations better. Rather than a museum display, objects collected within the Fundus and Laboratory are fully functional and meant to be operationalised. Both projects are informed not by nostalgia for historical media but rather by a desire to actualise them and project their operation into the future

functional applications but as operative knowledge theory. The application of the method within current research unfolds the concept of operativity in engagement with the computer program for sound synthesis. The analytic aspect of the method aids the extraction of singularities from a wider field of practice and discourse. It probes its use value in an engagement with notions of form, material, techniques of sound representation and objects functioning between realms of computation and aesthetics. During the 1960s, Michel Foucault employed the notion of archaeology to write about history, as described by Clare O'Farrell:

> Archaeology is about examining the discursive traces left by the past to write a 'history of the present. In other words *archaelogy* is about looking at history as a way of understanding the processes that have led to what we are today (O'Farrell, 2005)

In this sense, archaeology looks at the past from the present and aims to situate better and understand the present through the lens of the past. The archaeological process entails a three-part mediation going from the present to the past and back to the present. Luke Fowler quoted in Fell sums this approach up as "a practical and philosophical mediation on the past, through our contemporary selves" (Fell, 2022, p.106). This characteristic of the archaeological activity was highlighted by Foucault, who wrote:

> Archaeology does not try to restore what has been thought, wished, aimed at, experienced, or desired by men in the very moment they expressed it in discourse... it does not try to repeat what has been said by reaching it in its very identity. It does not claim to efface itself in the ambiguous modesty of reading that would bring back, in all its purity, the distant, precarious, almost effaced light of the origin. It is nothing more than a rewriting: in the preserved form of exteriority, a regulated transformation of what has already been written. It is not a return to the innermost secret of the origin: it is the systematic description of a discourse-object (Foucault, 2013a, p. 139-40)

Archaeology fits within Foucault's general perspective on discourse as a distinctly productive act constructing what he calls "practices that systematically form the objects of which they speak" (Foucault, 2013a, p.49). This productive aspect of the archaeological activity is highlighted by Hans-Jörg Rheinberger, who describes an activity of the archaeologist as digging out "the material sediments, the dispositions and depositions in which all theoretical knowledge is embodied and embedded" (Rheinberger, 1997, p. 253). In this conceptualisation, Rheinberger draws from the work of Foucault, especially his *Archaeology of Knowledge* (Foucault, 2013a). However, while Foucault was mainly interested in the production of discourses and their conditioning in apparatuses of power, Rheinberger orients his archaeology towards sources, channels and receivers of discourses - their material substrates. In this sense, Rheinberger's approach is close to Friedrich Kittler's orientation towards "inscription systems" (*Aufschreibesysteme*) and Wolfgang Ernst's media theoretical perspective. In *Media Archaeography: Method and Machine versus the History and Narrative of Media* Ernst writes that media archaeology "is interested in procedures and events that are not *historical* (i.e., narratable) but rather consist of *autochthonic transformations* (Foucault) within the realm of machines and their symbols" (Ernst, 2009, p.185). A fundamental awareness of Ernst is that media do not process cultural signs but rather technical signals, which are invisible to humans. This highlights the complicated place of technical media concerning culture. According to Ernst, media technology breaks out of the binary nature-culture, as a technical medium is "based on cultural knowledge-but, it is still of physical nature because there are electro- or even quantum-physical laws at work that are not solely dependent on the respective cultural discourses" (Ernst, 2009, p.185). In other words, technical media transcend human culture and emerge as autonomous entities.

A consequence of such a position for my research is fundamental. Primarily, in looking at the compositional work with pulsar synthesis, I am interested less in the role it plays as an agent in the history and aesthetics of computer music. This has been sufficiently examined in (Roads, 1985; Roads, 2015). Instead, the work I present focuses on a singular algorithmic technique (pulsar synthesis), its programmatic encapsulation (the nuPG program) and operative links between diverse forms of temporality emerging from their co-composition. These include focusing on the slightest temporal events enabling sound generation and computing processes, their relationship to the temporal scales of music composition, perception of time, and historical temporality. The aim is not to see pulsar synthesis in history from the macro-time perspective but to elaborate on how alternative temporality can be unpacked from its operation. Particularly important within this project is a close examination of the way the nuPG program enables alternative ways of relating to the history of contemporary music. Rather than placing the program in the preformed lineage, I propose to construct the lineage through the program as a kind of lens. The notion of microtemporality bridges the media archaeological perspective with a core aspect of the technique of pulsar synthesis, i.e., its focus on a temporal scale between milliseconds and samples. To paraphrase Julian Rohrhuber's view on algorithms, a composition practice with the nuPG program can serve as "a possible method to constitute and convey the peculiar existence of time" (Rohrhuber, Dean, and McLean, 2018, p.2).

Ernst's take on the temporal aspect of technology resonates strongly with recent developments in critical software studies (Marino, 2006; Fuller, 2003; Malina and Cubitt, 2008; Wardrip-Fruin, 2011) which propose to look through and beyond an image of a computer program as packaged functionality and expand upon its complexity as an object existing in time and producing its temporalities. The writing process moves backwards and forwards in time, connecting media instances from different periods; it creates its time axis, also challenging the concept of technological progress. Appropriating terms from the phenomenology of temporality as conceived by Edmund Husserl, Ernst focuses on an augmented temporal presence of digital objects "retaining" earlier constitutions and simultaneously "pertaining" subsequent ones. As Christopher Haworth shows in the context of analysis of the notion of 'microsound' in XXth century music, the process of protention modifies not only the future but may be subsequently modified by later occurring retentions (Haworth, 2018).

Ernst's awareness of media as amplifiers of human senses fit within the context of "sensorium", as proposed by Matthew Fuller. Fuller suggests a notion of a computer program as a constructor of "sensoriums". Within this notion: "each piece of software constructs ways of seeing, knowing, and doing in the world that at once contain a model of that part of the world it ostensibly pertains to and that also shape it every time it is used" (Fuller, 2003, p.19). In line with the above, an engagement with compositional work integral to current research begins as an analysis of the New Pulsar Generator (nuPG) program and a meeting with its diagrammatic circuit (1.1), underlying sound theoretical model, notions of an interface and programming paradigm. The idea of "sensorium" highlights a feedback relationship the computer program for sound synthesis establishes with the broader outside, i.e. sound technology, history and aesthetics. The design of the nuPG program can be viewed at once as an encapsulation of a particular sound theory and a generator of new conjectures. For example, the program not only interfaces an idea of 'Acoustical Quanta' (1.2) but also allows the theory to be heard and extended as an instrument through its operationalisation. For Ernst, such a perspective results in media becoming active "archaeologists' of knowledge" (Ernst, 2011, p.67) not just a passive representation but a probe to engage with and transform the knowledge space.

A consequence of media archaeology positioned as an operative method proposes a renewed understanding of temporality - extending an arrow of engagement from the present to the past and to the future and back. In the media archaeological method, Ernst postulates a rethinking of 'narrating' technological progress. Playing on the proximity between erzählen (narrating) and zählen (count, numbers) in the German language, Ernst shifts focus from historical narration toward mathematics and its potential for the understanding of technology and aesthetics. Ernst outlines that media archaeology does not tell a comprehensive and chronological history or counter-history. Instead, it focuses on procedures and artefacts within which history is recorded. The archaeology of the artefact engages with the past as fact - presence - not just as history. As an alternative to historical narrative, Ernst proposes a calculation-based ontology of technical media that articulates cultural memory. In this sense, the media archaeology method is "artefact-oriented"; it mirrors material culture research and is interested primarily in the physicality of technical media.

Within discussion on the nuPG program, the archaeological moment denotes the selection and isolation of singularities - material elements of the program - its interface, source code and sound synthesis model (1). Archaeology is also a site of analysis and comparative and trans-textual research. It deals with appropriate design solutions, innovations and transmissions of ideas over time. The archaeological investigation is like a hermeneutic approach, where the program's surface (interface) and depth (source code) are interpreted and compared. The process also engages with lines of reference and descent between the computer program and other fields of thought and practice, exploring its potential connection with the outside world.

Experimental System

The operative aspect of the archaeological method is manifested in an ultimately practical domain within which the computer program becomes a site of compositional experimentation. Objects of the archaeological process are not merely described but must be productively engaged with and re-situated. Functional for mobilising objects of archaeology within this research is a notion of the experimental system as conceived by Hans-Jörg Rheinberger in the context of empirical sciences (Rheinberger, 1992; Rheinberger and Fruton, 1997).

In the opening paragraph of the essay *Toward a History of Epistemic Things* Rheinberger points to a significant shift in approaching science as a field of study: "In a post-Kuhnian move away from the hegemony of theory, historians and philosophers of science have given experimentation more attention in recent years" (Rheinberger and Fruton, 1997, p. 1). The notion of experimentation is positioned against theory. Their relationship is, however, more nuanced, as clarified by Rheinberger in a conversation with Michael Schwab: "While I don't want to get rid of theory in empirical sciences, I nevertheless propose a reversal of poles: science is first and foremost a practical activity, although a theoretically laden one" (Rheinberger, 2013, p.199).

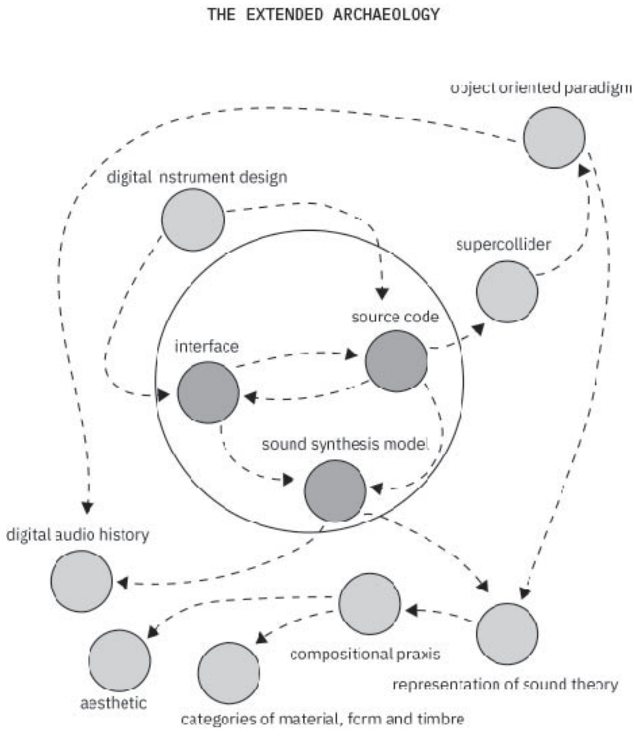


FIGURE 1: The Extended Archaeology of the nuPG spans a complex web of relationships cross-linking elements of the program identified within the archaeological stage (**??**). An extended web of connections is established, incorporating historical analogue techniques, other models of sound synthesis, problems related to the representation of musical data, a digital instrument design, and an interface. Dashed lines of arrows signify the temporality of connections; rather than being fixed and stable, they exercise a possibility of exchange. In this way, archaeology becomes constructive

.

Such remediation of theory and experimentation aids a view of science as "a process deeply inserted in the materiality of our world" (Rheinberger, 2013, p.199). In *History, Philosophy and the Central Metaphor*, Peter Galison postulates that: "We need a history of experimentation that accords that activity the same depth of structure, quirks, breaks, continuities, and traditions we have come to expect from theory" (Galison, 1988). In Rheinberger's view, we need to move beyond conceptualisations of theory and experimentation as a simple opposition and incorporate a notion of scientific work as a play with things - concepts, ideas, histories, and instruments. All of these in Rheinberger's project are not external to the "scientific reality" but material instances of theories - or, to use a term proposed by Gaston Bachelard - "theories materialised" (Bachelard, 1984, p.13). The notion of an experimental system is central to this particular framework of science-as-process. It concerns the place and extent of the research activities conducted by the researcher.

The experimental system is a notion of the practitioner, not the observer. A central aspect of the experimental system model is the experimenter's interaction with the material. This interaction is a prerequisite of creative experimentation, as Hans-Jörg Rheinberger postulates in a conversation with Michael Schwab: "If one is not immersed in, even overwhelmed by, the material, there is no creative experimentation" (Rheinberger, 2013, p.198). The relationship between the experimenter and the material is "a veritable two-way exchange" of the process of formation and of being informed. Such an exchange is visible in work with the nuPG program. The process of design and re-design informs the formal and material qualities of new compositional works. These, in exchange, propose new extensions to the program which generate new-sounding outputs. Additionally, participating in this feedback process are not only the computer program and compositional works but also other textual, computational and sonic sources encountered throughout the experimental process.

The experimental system is embedded in its instruments, apparatuses, procedures, and materials. Within the methodological model of this research, the experimental system incorporates *things* of the research protocol identified, sorted, classified and constructed through archaeology (0.5) and subjects them to a process concerned with the generation of new objects, i.e., formal, algorithmic methods of composition, computer programs, sound works and textual artefacts. The role of experimentation is, as Stephen A. R. Scrivener puts it: "the generation of unprecedented events and epistemic things, things embodying concepts, in which testing functions more as a means of affirmation than confirmation" (Scrivener et al., 2013, p.143). This process fits within a field of "study by design", which generates "know-how and understanding by studying the effects of actively and systematically varying both the design objects and their context" (Jong and Van Der Voordt, 2002, p.7). The experimental system does not simply deal with a representation of already known structures - sonic, technical, aesthetic or conceptual - but with their critical problematisation.

The kernel of an experimental system is the process of problematisation. Problematisation happens by constructing new and practical arrangements based on sources and models identified and explored within archaeological research methods (0.5). Michel Foucault describes *problematisation* as "the totality of discursive or non-discursive practices that introduces something into the play of the true and false and constitutes it as an object for thought" (Foucault, 2013b, p. 257). While the labour of archaeology mediates what things are - what is their immediate surrounding - problematisation proposes new ways of productively engaging with them. As such, the method of problematisation incorporates archaeology through its retrospective region of operation. However, its ultimate goal is a future-oriented one - a creative projection of new objects. Hans-Jörg Rheinberger calls the experimental system a 'machine for making the future' (Rheinberger, 1992, p. 309), which resonates with a mode operative within the creative process. The creative process happens in a designated time - the extended, continuous present - but also incorporates the past and projects into the future. My compositional work with the nuPG program displays these temporal axes as deeply intertwined. For example, a future-oriented composition with the program prompts questions related to the past, the history of the technique, and its use by other composers, which in turn projects new programming and sound-related extensions for subsequent explorations and possibly new works. One such example is the application of the sieve method to the pulsar masking parameter which rendered sonic results akin to Florian Hecker's scattering phrasing from the opening of 'Pulsar Wg'lett' track from the album 'Recordings for Rephlex' (2006).

Problematisation aids a program of engagement beyond a simple recreation or a reproduction of historical objects. While identification and analysis of its "things" (in the sense of archaeology) served as an essential stage - an inventory check, one can say - it is the act of extracting these "things" out of their context and exposing them to new situations which constitutes a critical stage in this project. A central aim of the experimental system is the generation of difference and surprise. This highlights the specific role of peripheral regions of the practice space. In a conversation with Hans-Jörg Rheinberger, Michael Schwab talks about "a sharp sense for secondary sounds" in a practical situation (Rheinberger, 2013, p.198). This is to underline the importance of the unexpected - the experimental system does not provide a prediction about the future. Still, it aids an engagement with it as a space of unknown and potential. Thus, the role of the "uncharted" peripheries of the experimental area is functional to the experimental system as a generator of difference. At the intersection of what is known and unknown new things emerge.

Integral to problematisation operating within this research are notions of *epistemic things* and *technical objects*. *Epistemic things* are those entities "whose unknown characteristics are the target of an experimental inquiry" (Rheinberger and Fruton, 1997, p. 3). As such, *epistemic things* address what one doesn't know yet. The *technical object* consists of a scientific object that "embodies the knowledge of a given research field at a given time"; "they might be instruments apparatuses and devices which bound and confine the assessment of the epistemic things" (Rheinberger, 2004, p.4). Underdetermination is inherent in epistemic things, whereas determination is inherent in technical objects. Both coexist within the experimental system and "whether an object functions as an epistemic or technical entity depends on the place or 'node' it occupies in the experimental context" (Rheinberger, 1997, p. 30). As Rheinberger points out, "within a particular research process, epistemic things can eventually be turned into technical things and become incorporated into the technical conditions of the system" (Rheinberger, 2004, p.4). Between these two polarities, there is room for gradations, and various degrees of complex things and objects whose function in the experimental system changes. Technicity and epistemicity are not irrelevant to knowledge production; they are not added to some pre-existing, ready-made core; their mutual relation operates simultaneously, forming the research object and outcome. Rheinberger describes their relationship as "a dialectic (...) and a constant oscillation between looking at something as being technically defined and looking at something as being epistemically open" (Rheinberger, 2013, p.209). The relationship between technicity and epistemicity echoes the dynamic between the computer program defined within the graphic user interface and the program as an operative code. Additionally, the code can be considered a hybrid entity. Depending on the perspective, it can be considered as the material inscription fixing the program's definition or as a tool of interaction (an interface) rearranging the program's purpose in a run time.

An adaptation of the experimental system to work with the nuPG program and as a generator of new compositions can be described through a set of steps:

1. Units of research—objects of computational, compositional or aesthetic nature—are identified in the process of archaeological study. These units may consist of hybrid objects such as a loop (3) which functions within multiple contexts such as programming language structure, music composition and aesthetics. The complexity of these objects' conditions becomes a site for experimental practice.

2. Amplified contexts of these objects, intensified relationships with other objects, aim at generating surprises. As Rheinberger writes: "Experimental systems must be capable of differential reproduction (...) to behave as generators of surprises"(Rheinberger, 2004, p. 3).

3. Processes of listening and composing with these new findings is a stage of their application and verification
4. The outcome of such a process is primarily a set of new musical works. However, these should not be separated from the formal and computational conditions of their emergence.

Experimental music is conducive to the notion of the experimental system (for example see descriptions of various works of David Tudor in (Nakai, 2021)). In contemporary art practice, the term "experimentation" denotes a set of methods and practices that attempt to break away from established formal languages and traditions. Experiments in art deal with novelty, trial and moments of surprise. It is closer to Rheinberger's view than the classic scientific conception of experimentation associated with notions of reproducibility, control and measurability.

Part I

The Archaeology of The New Pulsar Generator

The first part of the thesis functions as a prototype for archaeology of a specific digital sound synthesis technique and its software implementation. Such an investigation does not limit itself to published materials (scientific articles, software manuals, programme notes, source code etc.) or the outputs (musical compositions, audio demonstrations etc.). A key argument of this work is that understanding how the digital artefact and the systems that support it function require an operative approach. Following Wolfgang Ernst's media archaeological perspective, the practice of computer music composition is viewed foremost as an engagement with the time-critical aspect of the computer program and its processes. Time-criticality signifies technologies in which time plays a crucial role. Time-critical technologies do not simply happen in time - for example, time-based media - but generate their temporality and - crucially - are capable of time-axis manipulation. One of the significant implications of the concept of time-critical technologies is that the focus of analysis shifts from the discursive (signs) domain to the non-discursive (signals) domain. A change from "textual interpretation" towards "diagrammatic reading of circuit plans and material hermeneutics (media-archaeological philology)" essentially operational and concerned with "dis- and re-assembling" of the artefact (Emerson, 2013). The operative archaeology of pulsar synthesis develops across four distinct chapters.

The opening chapter *Fundamental Model of Pulsar Synthesis* sketches out essential elements of the technique and looks for its sound theoretical lineage in the work of Dennis Gabor and the concept of "acoustical quanta". A view of the fundamental theoretical objects the model deals with is crucial not only for understanding the technique's working but also serves as a scheme for compositional interventions discussed later. The technical description of the model is contextualised by a discussion on the roots of a particular sound formalism - the acoustical quanta - proposed as a conceptual point of departure of the technique of pulsar synthesis. The discussion focuses on a complicated relationship between domains of sound analysis and synthesis and problems of representation of sound as a signal. It attempts to map the complex origins of the technique of pulsar synthesis. The section also proposes an overview of basic models categorising sound analysis and synthesis techniques. Such categorisations vary depending on the author and criteria. While in-depth discussion on the rationale behind different classification systems is beyond the scope of this work[12], however, such an elementary perspective will prove indispensable for offered within this work perspective of pulsar synthesis as a non-standard synthesis model. The relationship pulsar synthesis affords with historical analogue techniques, and the practice of electronic studios is treated only peripherally. An emphasis is on the particularities of the digital medium and precise programmability, and possibilities opened up for methods of design for sound synthesis and composition. Consequently, the proposed discussion on notions of material and form - deeply rooted in the music-theoretical discourse - deals with their limited sense of representation in a digital format.

The material point of departure for the following chapter *The Model of Instrumental Encapsulation* consists of a close reading of the graphic interface and programming language of the original Pulsar Generator (2001) program. The Pulsar Generator (2001) program designed by Curtis Roads and Alberto de Campo exemplifies the first instrumental encapsulation of the technique of pulsar synthesis. The reading proposed in the chapter focuses on an overview of the program's functionality and

---

[12] Widely used systems of classification have been proposed by (De Poli, Piccialli, and Roads, 1991) and (Smith III, 1991).

analysis of mapping strategies between the synthesis method, its graphic interface and the design possibilities enabled through SuperCollider, an object-oriented programming (OOP) language. After a brief historical introduction, the analysis deals with specific aspects of the paradigm and its relationship with the design of the computer program for sound composition. An integral component of the study consists of a re-implementation of the program where the program becomes a probe to test methods at the intersection of computation and composition. Processes of computer program interpretation and re-implementation processes are inseparable within this thesis's context. The basis of such coupling is an argument that full reading of historical computer programs requires seeing them first as dynamic and operational here and now. The analysis of the program by Roads and de Campo is supplemented by reading of two other historical implementations of the technique: Pulsar Generator (2004) by Tommi Keränen and Particularity (2011) by Chris Jeffs. Two factors dictated the choice of these two particular developments. First, both implementations, like the original one, use SuperCollider as the main design environment[13] and function as extensions and redefinitions of design solutions proposed by Roads and de Campo. Second, each of the three implementations comes with a body of compositional works. An analysis of a selection of these works appears throughout the thesis in the context of particular program design problems.

The chapter *Text-Script-Code-Language* expands upon the analysis developed within the preceding study and takes a closer look at the programming language as instrumental for implementing particular composition and sound editing methods. Drawing from a conceptual repository of information sciences and conversational approaches to programming, the chapter mobilises a renewed perspective on object orientation focused on the dynamic redefinition of objects and their relationships. The chapter also includes a short overview of conversational methods developed as part of the project.

The final chapter —*Material and Form*— develops notions of form and formalisms from the perspective of a computer program as a temporal and operational device. The chapter integrates the results of preceding analyses in search of formal categories capable of accounting for computational and aesthetic aspects of the program's functioning. Opening the chapter, a short overview of notions of form and material conceived in early electroacoustic and computer music links the preceding discussion with aesthetic and compositional contexts. The central part of the chapter focuses on the notion of the loop as an integral computational and compositional object of the nuPG program.

Throughout Part I, key themes are: operational archaeology, composition with historically inherited materials, sound formalism and materialism.

---

[13] Exception being PulsarGrain UGen designed in C++ by Tomi Keränen which needs to be precompiled for SuperCollider

Chapter 1

The Fundamental Model of Pulsar Synthesis, Sound Theoretical Sources and Instrumental Encapsulations

The definition of the fundamental model of pulsar synthesis develops across three delineated fields: 1) identification of sound theoretical model; 2) representation of sound as a signal; and 3) analysis of a historical model of instrumental incarnation. While an extensive discussion on the topic is provided in (Roads, 2001; Roads, 2004), it seems essential for current research to recapitulate key elements of the discussion. The basic circuit (1.1) together with pictorial representations of pulsar (1.3), pulsaret waveform (1.5), an envelope (1.6) and pulsar train (**??**) constitute key sources - a template - for further discussion and development in both computer program design and compositional practice integral to this research. Existing definitions of the technique of pulsar synthesis function within diagrammatic and symbolic modes of notation. A definition developed within this chapter extends the scope to include the computational framework of the SuperCollider programming language. Identifying the link between the definition, domain of signal representation, and the audible is an important addition to existing perspectives.

This primary circuit of pulsar synthesis- the sound model- coheres to a particular analytic formalism. The formal model of the technique can be viewed in the context of a conceptual and operational framework for the sound signal representation as proposed by Denis Gabor in 'The Theory of Hearing' (Gabor, 1946). A claim proposed in the following section of the chapter is that the technique of pulsar synthesis builds upon Gabor's theory of acoustical quanta—in essence, analytic—and proposes its synthetic and operational extension. The relationship between the synthesis technique and the theoretical model is not one of a direct transfer. Pulsar synthesis does not simply demonstrate the theory; it problematises it by extending its operationalisation to include the synthesis of new - unheard - sounds. A discussion on the particularities of the method proposed by Gabor serves as a point of departure for a deepened understanding of a complicated relationship between processes of analysis and synthesis. It forms a basis for proposed later in the text compositional methods (2). The discussion of the technique of pulsar synthesis in conjunction with reflection on the work of Denis Gabor and his concept of 'acoustical quanta' allows us to mobilise a more nuanced view of the process of synthesis, to reveal its deep entanglement with methods of analysis, representation of sound as signal and signal as data. An introduction of the analytical model also provides a background for productive extension of the technique in wavelet transform (2.1).

Functional for current research is an analysis of the original Pulsar Generator (2001) program designed by Curtis Roads and Alberto de Campo in 2001 at The Center for Research in Electronic Art Technology (CREATE). The program was operating on Mac OS9 and became obsolete after the release of OSX in March 2001 - that is around nine months after the release of the PG (2001). The authors have sold approximately 50 copies of the program for $49 each. The PG (2001) program constituted the first instrumental encapsulation of the pulsar synthesis technique. It generalised and extended the basic model (Section 1) with precise control over a collection of synthesis parameters. The extension to the basic model included control over pulsar train duration, pulsar fundamental and formant frequency envelopes, pulsaret waveform and its envelope, amplitude and spatial path. Additionally, the program provided a set of pulsar masking algorithms that allow for the procedural omission of pulsars from a continuous train.

The chapter concludes with an analysis of an extended model of pulsar synthesis incorporated within the PG program: a model of the shared fundamental frequency with time-varying formant control, spatial path and amplitude (Roads, 2004, p. 147). A vital element of the graphic interface and its relationship to those described in the previous section's fundamental technique model are discussed. An essential set of sources contributing to this analysis constitute the PG program's manual and source code. An integral part of the PG analysis consisted of a design of the nuPG program (A). Such an approach fits into the general operational perspective realised within the current work. This can be summed up in a thesis that historically inherited sound synthesis techniques and their instrumental implementations require mobilisation as operational and functional objects to be fully analysed. The analysis includes a brief overview of two other implementations of the technique Pulsar Generator (2004) by Tommi Keränen and Particularity (2010) by Chriss Jeffs. These two models were chosen based on computational and compositional output criteria: computational because each of the implementations uses SuperCollider as a core development environment, and productive because each of the implementations functions together with a body of compositional works.

1.1 Models Between Diagrammatic and Formal Representations

The exposition of pulsar synthesis unfolds along two distinct but intertwined modes—formal and diagrammatic- representing objects and their relationships. A formal mode does it within natural language boundaries (with mathematical symbols), and a diagrammatic does it visually. The standard mode functions on well-established concepts, i.e., mathematics, computation and sound theory. Such a formal representation can incorporate a variety of descriptive approaches such as control-based information protocols (MIDI), score descriptors, and more complex programming paradigm-related (e.g. object-oriented, functional etc.) musical structures. As such symbolic representation can be considered multidimensional. According to Bresson and Agon, formal models provide a structural representation of sound. However, their compositional applicability is unknown. The term structural in this context refers to a recipe-like description of a sound model[1]. To operate the structural model, a user requires a diagrammatic interface. A diagram of the Pulsar Generator (**??**) by

---

[1] The term structural in this context can be likened to a notion of an instrument as described by Max Matthews in Music-n programming language. The essential element required to operate it was a score. In our context consisting a symbolic representation

Curtis Roads and Alberto de Campo represents a flow of data and logic of mathematical iteration. As such, the diagram represents “a schematic of the course of control through the sequence” (Goldstine, Von Neumann, and Von Neumann, 1947, p.87). The diagram is a vital link between formal definition and computational implementation. As observed by Morris and Gotel, diagrams are "a general purpose tool for planning an automated computation at all levels of composition or decomposition" (Morris and Gotel, 2006, p.139). Diagrams are an invaluable aid for understanding the computational process of working with obsolete computer programs.

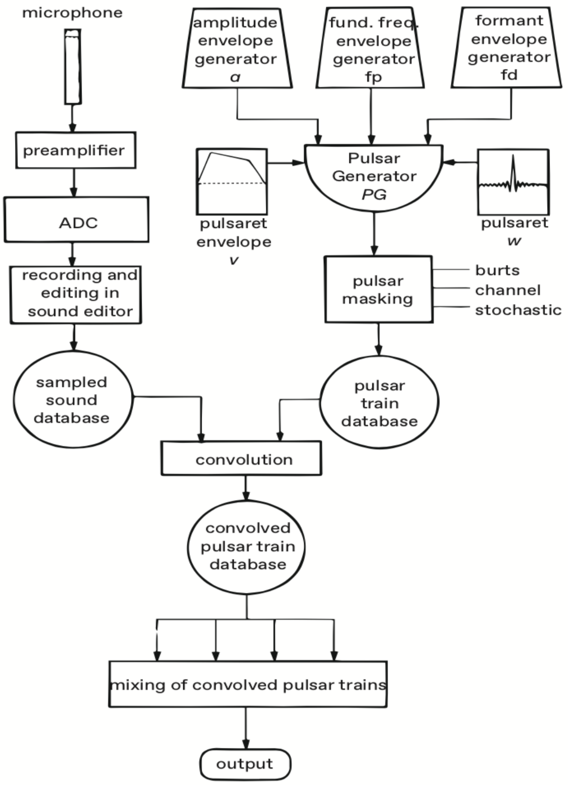


FIGURE 1.1: A diagrammatic representation of the Pulsar Generator program by Curtis Roads and Alberto de Campo. Source: (Roads, 2020), figures conversion Marcin Pietruszewski

Generally speaking, a diagrammatic representation expresses the essential features of the process succinctly. A set of visual elements substitute more complex objects and their relationships. The diagram reduces the complexity; it also structures and organises the flow of information. It also aids the readability of information, understanding and imparting knowledge and performing operations. Koray Karaca underlines the role of diagrams as models mediating between the "abstract system" and its "concrete actualisation in the world". Expanding upon the work of Margaret Morrison and Mary S. Morgan on ’Models as Mediating Instruments’, Karaca argues that diagrammatic representation proves more efficient than textual representation as a tool of design and analysis. This mediating aspect pointing towards the practical and operative modes of engagement with technology can be found in the reading of diagrams proposed by Wolfgang Ernst. Wolfgang Ernst in *Media Archeography: Method and Machine versus History and Narrative in Media* locates the diagram at the starting point of an analysis of technical media culture and as an integral element of the archaeological media method. Within such a method, diagrams function in their rudimentary sense as visualisations of information patterns, circuits, processes, objects and their relationships. Within the above scope, the diagram is understood in line with its scientific usage: i.e., as an engineering diagram, a blueprint of a machine, and an indexical mapping of circuitry for the visual representation of the flow of data in a structured way[2]. Beyond this rudimentary view and emphasising the mediatory role of diagrams, Ernst focuses on their role as "a crossing-point between epistemologically wired humanities analysis of technical media and the engineering enabled understanding of and tinkering with operationality" (Parikka, 2011, p.65). In the context of current research diagrams of pulsar synthesis (1.3, 1.4, 1.5 and 1.6) and Pulsar Generator (1.1) function as gripping points in the process of understanding the technique. The diagram defines the technique as based simultaneously on the logic of signal flow and a kind of "mathematics-in-action" (i.e. software engineering). From a perspective of computer music composition, the diagram delineates objects and the relationships between them. In such a sense, diagrams include and encapsulate vectors of transformation. As observed by Jakub Zdebik, "The diagram is an image of something to come rather than something that is already there" (Zdebik, 2012, p.16). The future-oriented aspect of diagrams—their primarily operative and temporal character–is a key point of their interpretation in the context of work with pulsar synthesis. From the diagram as an analytic starting point, the diagram in computer music composition becomes a site of unfolding practice.

The composition processes with synthetic sound can be viewed as a general procedure of complexity reduction. At a fundamental level, digital sound consists of discrete values - a continuous sound sampled at regular intervals. This is the first reduction. In a digital domain, the representation of sound consists of a sequence of binary digits that encode sampled and quantised values from the signal. While such representation provides accurate visibility, edition, and manipulation possibilities, it does not convey musical information. Bresson and Agon points out that ”bits are symbols in the computer: they correspond to a physical phenomenon interpreted as binary values. They can be combined to constitute bytes, characters, or numbers, yet higher-level symbols” (Bresson and Agon, 2007, p.3). The representation is operational and contextual. What is considered an operative object in one domain does not necessarily get transposed into a different realm of use. Bresson and Agon continues: "neither a bit nor a number will usually be considered as musically relevant symbols likely to be interpreted and handled by a composer" (Bresson and Agon, 2007, p.3). Guerino Mazola contextualises this problem as a mutual dependency between data and the navigational paradigm. In *The Topos of Music*, he writes: "Digital substance, reduced to BIT = 0,1 of two substance values, 0 and 1, of OFF and ON (...) is just a minimal substance set, but it cannot *per se* be responsible for any knowledge. A digital record is worthless without being organised in a concept form, i.e. as encoded knowledge. 'Understanding', i.e., decoding information, is needed to control and manage its knowledge potential. The digital age is not centred around 'Bits and Bytes' but around their accessibility and handling, in short: data navigation" (Mazzola, 2012, p.40). Sound as data requires a structured representation - a "navigational' paradigm" - to be applicable in a compositional context. This is a secondary reduction. The model of sound in the digital domain is operative and normative. The model is operative in that it affects, enacts and functions. The term normative establishes a relationship between the model and theoretical, historical or

---

[2] Composer David Tudor approached circuits as both the definition of an instrument and as a score for a performance to unfold. For more on Tudor's work, see (Nakai, 2021)

aesthetic contexts. This dynamic between functional reduction of operability and a template as a norm is present in work with the nuPG program.

The technique of pulsar synthesis combines and merges established principles within a new paradigm (Roads, 2001). The critical aspect of the method—a repetition of waveform segments—echoes in some way the repetitiveness of the GENDY (a portmanteau constructed from the French words *generation* and *dynamique*) algorithm as introduced by Iannis Xenakis (Xenakis, 1992). While in GENDY, "the sound is made of the repetition of an initial waveform and that at each repetition the shape of the waveform is distorted according to both time and amplitude" (Serra, 1993, p. 13), the waveform—called the pulsaret—of the pulsar synthesis circuit is by definition constant; what varies is the width of the waveform, which corresponds to formant frequency. The contextual relationship between pulsar synthesis and the GENDY algorithm became a source for the composition ’nupggendy3modulatorDistCauchy (screwdrivers)’ included in the accompanying portfolio. Additionally, the folder ’Sound Example Files’ contains a subdirectory ’nuPG GENDY Approximations’ with a series of studies approximating the timbre and temporal development of the GENDY algorithm with the nuPG program. The basic idea for these studies was to employ the algorithm[3] as a generator for table data and a modulator for various parameters of the pulsar program. Curtis Roads also points at pulsar synthesis’ similarities to historical techniques designed around the principle of the filtered pulse train (e.g. Ondioline [4], Hohner Elektronium(Williams, 2016b; Roads, 1996a)). It is worth noting that the micro-structural approach to time-modelling through the repetition of small sound segments had been around before the advent of computer music. As pointed out by Agostino Di Scipio, "early works of Elektronische Musik and musique concrète were realised according to similar views of sonic design, long before computers became available to musicians" (Di Scipio, 1995, p.40). In addition to Stockhausen’s *Kontakte* among those works, we find Xenakis’ *Concret PH* (1958) and *Analogique B* (1959), and Pousseur’s *Scambi* (1957). The emphasis on formant frequency modelling likens the technique of pulsar synthesis to FOF (*fonction d’onde formantique*) and Vosim (Kaegi and Tempelaars, 1978), techniques which evolved from attempts to simulate human speech. Pulsar synthesis also functions within diverse models of micro-temporal representation of sound. Included in the accompanying portfolio composition, ’sieveSequence(ggg)’ explores the context of pulsar synthesis as an approximation of synthetic speech. Additionally, a series of sound examples are included in the subdirectory ’nuPG vox Approximations’ of ’Sound Example Files’. Curtis Roads points to a vast dictionary of representations within contemporary digital signal processing algorithms that attempt to classify the micro-temporal aspect of auditory signals. Concepts such as "Gaussian elementary signal," "short-time segment," "windowed segment," "window function pulse," "wavelet," and "formant-wave-function" denoted attempts at categorising but also operationalising the micro-temporal domain (Roads, 1988). Generally, these approaches are unified by a singular aim to combine the time-domain information of a signal— its starting time, duration, envelope shape, and waveform shape—with its frequency-domain information— the frequency of the waveform inside the grain, the spectrum of the waveform and envelope.

The basic sound model of pulsar synthesis, as formulated by Curtis Roads in (Roads, 2001; Roads, 2004), is a central reference point in the description of a sound

---

[3] The SuperColider adaptation of the algorithm by Nick Collins has been used. See `https://doc.sccode.org/Classes/Gendy3.html`

[4] See the demonstration (`https://www.youtube.com/watch?v=hy5w7Fz0pDo`) of Ondioline by its inventor Georges Jenny

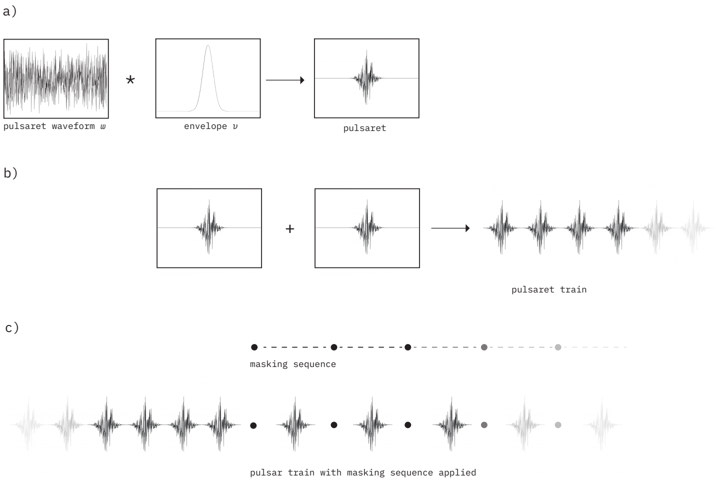


FIGURE 1.2: A visualisation of three modalities (multiplication, addition and subtraction) constructive for the technique of pulsar synthesis. A progression from the (a) pulsaret as a multiplication of a waveform w with an envelope v; (b) A composition of a pulsar train as an addition in time, iteration, of pulsarets; and finally (c) masking as subtraction.

model of the technique. It is worth noting that the form of that model was a result of a process of conceptualisation and experimentation that predates these publications. The earliest material manifestations of the method, which exist as computer program scripts (SuperCollider 1), date back to 1997. Curtis Roads also points to conversations with Professor Aldo Piccialli and his colleagues from the Physics Department of the Universita di Napoli "Federico II" and refers to three texts (A. Piccalli, 1986; Cavaliere and Piccialli, 1997; De Poli and Piccialli, 1991) as a source of inspiration. The model consists of a pulsaret waveform, envelope and pulsaret train (a repetition of pulsars). The primitive element of that model, a single pulsar (1.3), consists of an arbitrary pulsaret waveform w with a duration d followed by an interval of silence s. The total duration of a pulsar renders as follows:

$$\mathrm{p = d + s}$$

where p stands for *pulsar period*, d the *duty cycle* or *duration* and s an interval of silence. The repetition of p forms a *pulsar train* called also pulsar stream (1.4). The train can be described as a function of the rate of pulsar emission with a period of $\mathbf{f_p = 1/p}$ and the frequency of the duty cycle $\mathbf{f_d = 1/d}$.

The value of $\mathrm{f_p}$ typically ranges between 1 Hz and 5 kHz. While the value of $\mathrm{f_d}$ is from 80 Hz to 10 kHz. In pulsar synthesis a value of both $\mathrm{f_p}$ and $\mathrm{f_d}$ is independent and can vary constantly. The independence of fundamental and formant frequency paths is one of the key characteristics of the sounding of the technique.

As displayed in 1.4, the ratio of d:s varies while the value of p remains constant. The characteristic timbre of pulsar synthesis comes from independent control inputs of both the rate of pulsar emission and the duty cycle. The emission rate under 18 Hz results in discrete pulses; between 18 and 30 Hz, the resulting sound signal flutters between discrete and continuous textures; from above 30 Hz, the pulsar train fuses into a continuous tone. The technique of pulsar synthesis, with its multi-temporal

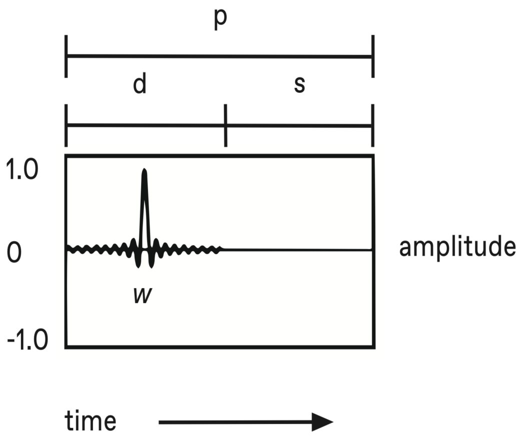


FIGURE 1.3: A single pulsar (1.3), consists of an arbitrary pulsaret waveform w with a duration d followed by an interval of silence s. Source: (Roads, 2020), figures conversion Marcin Pietruszewski

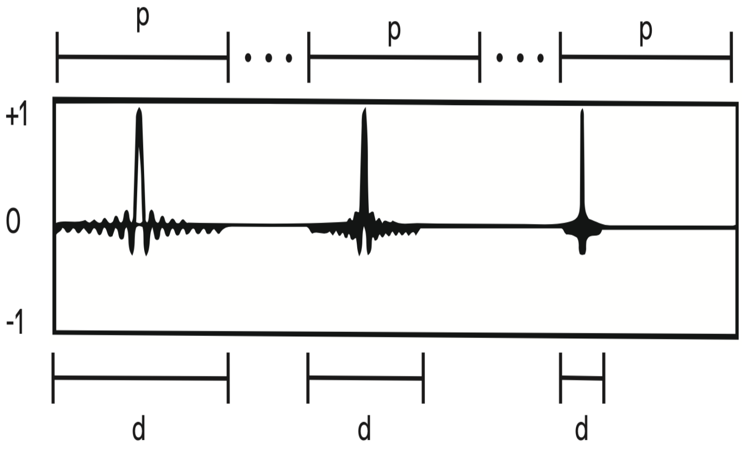


FIGURE 1.4: The succession of pulses is called the 'pulsar train'. Source: (Roads, 2020), figures conversion Marcin Pietruszewski

affordances as a system of interconnected patterns evolving on multiple timescales, exemplifies the notion of rhythm-tone continuity. An extended discussion on this and related notions are presented later in the text in a section devoted to *Form and Material* (3).

The shape of the *pulsaret* waveform w can be variable (1.5). In a basic form, the pulsaret waveform can consist of a fixed synthetic type: the sine, saw and square. Complex variants include waveforms generated by time-varying signals extracted from a sampled sound or drawn by a user.

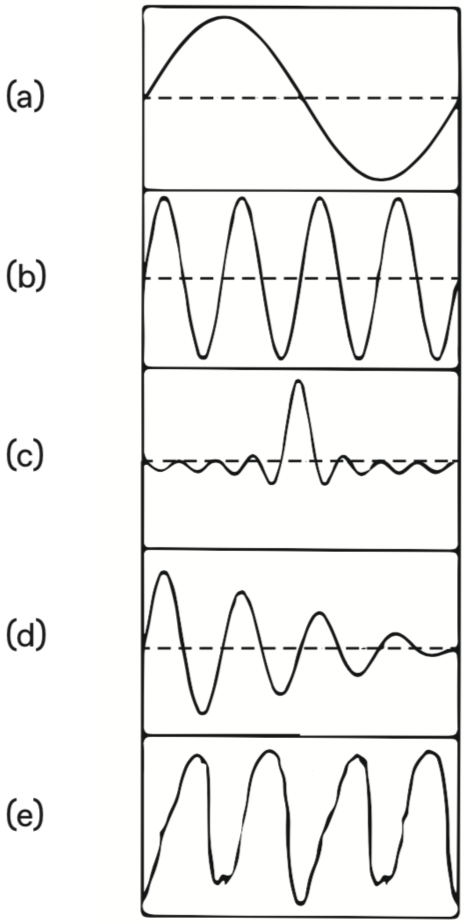


FIGURE 1.5: Examples of pulsaret waveforms. In practice, any waveform can be used. (a) Sine. (b) Multicycle Sine. (c) Gaussian Limited Sine. (d) Saw. (e) Gray Noise. Source: (Roads, 2020), figures conversion Marcin Pietruszewski

An essential element of the model of pulsar synthesis is the *pulsaret envelope*. The *pulsaret envelope* functions as a limit in time for the *pulsaret waveform*. The envelope can be shaped freely (1.6). The envelope strongly affects the spectral content of the pulsar stream (Roads, 2004, p. 146). A rectangular envelope (1.6(a)) produces a broad spectrum with strong peaks and nulls for any pulsaret. An envelope with a steep attack and decaying exponentially corresponds to a well-established formant synthesis configuration, FOF and Vosim techniques (Kaegi and Tempelaars, 1978).

An essential generalisation in the above synthesis model is that the waveform w and envelope v can be of any shape. This consists of a particular case of pulsar

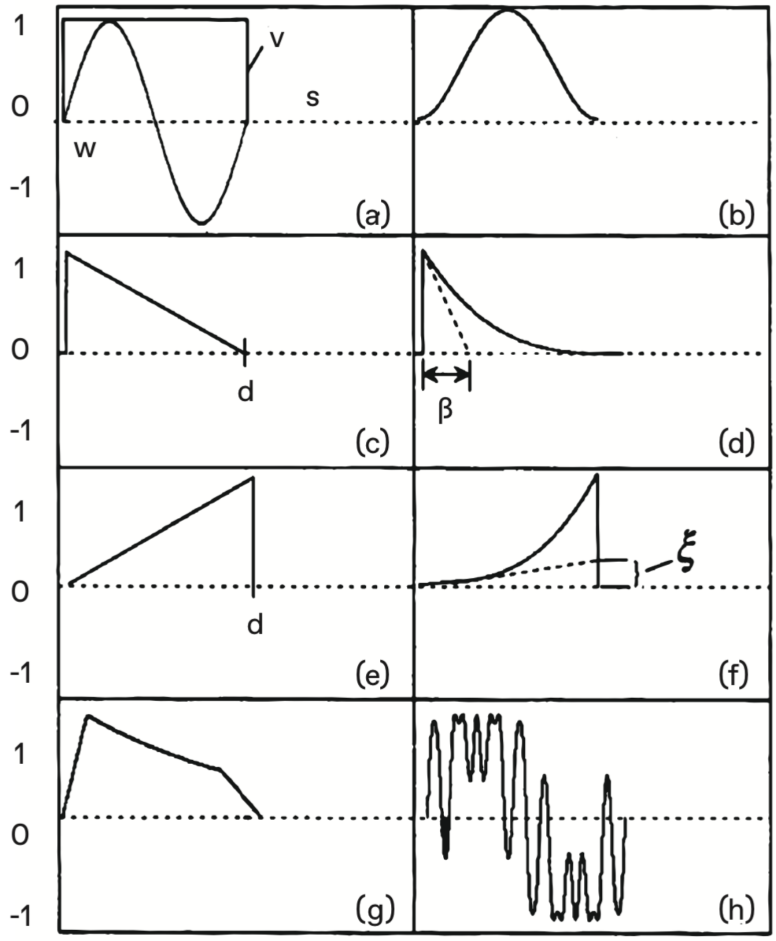


FIGURE 1.6: A standard catalog of pulsaret envelopes v. (a) Rectangular. (b) Gaussian. (c) Linear Decay. (d) Exponential Decay. β-steepness of an exponential curve. (e) Linear attach with duty cycle d. (f) Exponential attack. γ-steepness of an exponential curve. (g) FOF envelope. (h) Bipolar modulator. Source: (Roads, 2020), figures conversion Marcin Pietruszewski

synthesis.

The two modes of definition—formal and diagrammatic—function as gripping points for the process of understanding pulsar synthesis, its objects, processes and outputs. Moreover, in the context of creative practice, they can act as points of departure for the experimental process. Within current work objects such as the *pulsaret*, an *envelope*, or the *pulsar train* serve as operational elements of the compositional practice. To trace the formal and diagrammatic construction of objects of pulsar synthesis allows envisioning conjectural relationships. The design of the New Pulsar Generator (nuPG) from the outset operates in dialogue with available resources in order not to update previous now obsolete implementations but to mobilise constructive principles of pulsar synthesis technique as elements of the experimental compositional setting.

1.2 "Acoustical Quanta": Sound-as-Signal

As observed by Wolfgang Ernst, “acoustic quanta” function as "subliminal temporal elements from which sound can be calculated in the true time domain" (Ernst, 2020b, p.50). The view of acoustic phenomena as "acoustical quanta" proposed by Dennis Gabor was conceived against quantum physics and information theory[5]. The view also portended multi-temporal analysis, such as wavelets (2.1). Instead of illustrating wave mechanics with acoustical phenomena - in the style of George Ohm (1843) and Hermann von Helmholtz (1885) - Gabor takes an alternative position and proposes an application of a formalism from quantum physics. Gabor introduced "quanta of information", which unified temporal and Fourier[6] representations of signal, as a general particle-based view on acoustics. As explored in *Acoustical Quanta and The Theory of Hearing*, acoustic notions of 'what' and 'when' concerning sound are fundamentally blurred. Perhaps the simplest explanation is that sound cannot be held still to be examined and precisely indexed. The sound is at once inherently in the past - as all sound when heard is always already gone[7] - and at the same time it is continuously unfolding in the here and now - as all physical sound is heard in the present. The content of a sound wave - the 'what' of sound - exists only as a function of variation of pressure in time and location. Gabor showed that it is impossible to simultaneously observe a sound's frequency distribution and its temporal localisation with great precision. As he described it, the minimum area of this limitation is constant, while the lengths of the sides may vary. The informational diagram - a two-dimensional matrix - proposed by Gabor discretised the temporal continuum (horizontal axis) and the frequency continuum (vertical axis). The discretisation process, however, implicitly highlights another aspect of Gabor's project, namely the importance of the observation scale. In the formal framework of a new signal representation, every change of the temporal scale of observation potentially leads to highlighting different properties in the signal subjected to observation within a relatively large space of solutions whose extremes go from harmonic

---

[5] It is worth mentioning that Gabor considered sound primarily as a signal. Thus an essential aspect of its research was devoted to problems of information transfer, a minimal unit of signal relevant for the preservation of information etc., see (Gabor, 1946)

[6] With his 1822 *Théorie analytique de la chaleur* (The Analytical Theory of Heat), Joseph Fourier implicitly analysed vibrational events as “sonic” events, which made it possible to symbolically calculate the temporality of world signals, which, when calculated, could be used to manipulate electroacoustics.

[7] To recall Roland Barthes notion of time as "punctum" (Barthes, 1981) this is not dissimilar to the way photography works, as it captures objects which belong to the past already at the moment when the shutter opens

oscillation (sinusoidal function) to the discontinuity of a brief impulse (delta function)[8]. The acoustical quanta is an analytical model that attempts to integrate the "language of time and the language of frequency" (Gabor, 1947, p.592). As Gabor observed, "time and frequency are complementary, rather than mutually exclusive" (Gabor, 1946, p. 431), and as such, should be expressed in an analytical model that accounts for both. The research objective pursued by Gabor was precisely that of demonstrating the existence of "mathematical methods suitable for this purpose" (Gabor, 1946, p.231). These methods - as Gabor observed - necessarily had provided a high degree of precision simultaneously for the determination of time and the determination of frequency. Such a problem has been addressed with the conceptual tools of quantum mechanics and the notion of the uncertainty principle proposed by Werner Heisenberg (Heisenberg, 1974) two decades before Gabor's work. Already in 1925, in a lecture given in Göttingen, Norbert Wiener pointed out the relationship between frequency and time. A waveform requires time to unfold. This time must be sufficient for the waveform to complete one full cycle. As Weiner writes: "if you take a note oscillating at a rate of sixteen times a second and continue it only for a one-twentieth of a second, what you will get is essentially a single push of air without any marked or even noticeable periodic character" (Wiener, 1976, p.545). If the fundamental frequency of the signal is cut short, it "will not sound to the ear like a note" but as a noisy click, transient, pop (Wiener, 1976, p.545). Kromhout observes in this context the uncertainty principle in action: "beyond a certain limit, tending towards (but never reaching) the infinitesimally short timeframe of a Dirac impulse, it becomes physically impossible to shorten a sound without losing its sonic identity. Beyond this threshold, the clarity of sine waves gradually gives way to the instantaneity of Dirac impulses. Clearly, definable frequencies turn into fuzzy, undefined spectra until all that remains is a transient blow, pip or noise" (Kromhout, 2020, p.34).

The model of acoustical quanta provides a formalism to describe the relationship between the temporal and the spectral - a link that is not only of practical importance but, as proven in the decades after Gabor, artistically relevant. Consequently, the difference between the 'wave' and 'grain' representation of signal should not be understood as a difference of just technical nature. Compositional consequences of both views are evident as these reflect different ways of thinking about sound and, therefore, different ways of composing it[9].

Gabor was simultaneously interested in the domain of physical-acoustic phenomena and auditory perception - as the title of his article *Acoustical Quanta and The Theory of Hearing* indicates. The reduction of sound to signal and issues related to signal representation are fundamental in the domains of analysis and synthesis. These issues are pertinent in the design and practice of nuPG too. However, a direct link between the digital sound synthesis technique and Gabor's research must be accompanied by a cautionary note. It must be highlighted that Gabor's project's goal was primarily analysis and not a synthesis of new sounds. As described by Gabor, an instrument called a “frequency converter”, is precisely a device for experimental

---

[8] For Gabor the notion of 'observation time scale' is completely implicit: making it explicit will be up to some of his followers, including Jean Morlet (1931-2007), who will come to formulate methods of wavelet analysis attuned to time-scale representations of signal

[9] This dualism reverberates an essential theme in the scientific debate of the 1940s, that of the duality between particle and wave. In the model advanced by Louis de Broglie in 1923, the particles of matter that move at identical speeds reveal, at a particular observation scale, the properties of a wave and can then be traced back to the laws of wave mechanics; the similar particles looked at another observation scale, can be described in their respective individual motions, which requires a transition to quantum mechanics. Dennis Gabor's theoretical model of the quanta of sound presents itself as the first formal representation (in a mathematical sense) in the history of corpuscular theories of sound

verification of the theoretical model. The role of such an agent was to test methods of information "compression" and "expansion", issues related to the speed of information transfer. The "frequency converter" illustrated the fact that it can change ("convert") the frequency and at the same time leave "the rhythm, that is the temporal articulation of speech or musical sound signals" Gabor, 1946, p.452. The idea of digital sound synthesis as a coherent formalism capable of different implementations for musical or other purposes only emerged a few years after the proposals by Gabor. Principal in this context was the work of Herbert Belar and Harry Olson at the RCA (Mark II Synthesiser) and Max Mathews and his group at Bell Telephone Laboratories. Matthews synthesised digital sound[10] using the IBM 704 computer at IBM World Headquarters. The computer wrote samples into a reel-to-reel storage drive, which then was converted to sound at Bell Laboratories using a custom-built 12-bit vacuum tube digital-to-analogue converter (Roads and Mathews, 1980; David, Mathews, and McDonald, 1958). The combination of the converter - which made it possible to represent different waveforms as analogue signals - and the control over the characteristics of the sound through the programming power of MUSIC (more about it in section 2.3) sets Matthews's work apart from earlier approaches to using the computer as a sound generator in Britain and Australia; those earlier computer sounds were produced by sending on/off control signals to the membrane of a speaker to generate pitched buzzes (Doornbusch, 2004; Fildes, 2008). However, the sound model Matthews had adapted in these first experiments in synthesis has no relation to the formalism proposed by Gabor. This model derived a description of a sound entity from a musical note and offered the following parameters - pitch, duration and amplitude. The question of timbre or tone colour remained a prospect as these first experiments used a distinct type of oscillator - an equilateral triangular waveform - of which a single period was repeated at pitch rate.

Holding on from within the analytic domain is the subject matter of representation of sound as signal and, ultimately, signal as data. The transformation is foundational for the synthetic function of the nuPG program. As described above, the first transformation happens between an analogue sound and a digital signal. The basis for the synthetic process of the nuPG is the second transformation: translating digital signals into operable data. The shift from signal to operable data can be called transposition. Transposition is transferring data or phenomena from one medium or system of coordinates to another. The data in the starting system and the target system differ, but both are externally observable, measurable, and quantifiable. The data are transposed, i.e. moved from one place to another. This shift highlights the already operative aspect of data forming. Johanna Drucker reconceives 'data' as 'capta'. Whereas data is assumed to be a 'given' — objective and trusted, observable and recordable; capta is to be taken actively — to engage and analyse (Drucker, 2011). The hypothesis-saturation of data already reflects the purposes and practices of those who gathered or represent them. Representation technologies and related media, from spectrogram to wave monitors, all involve exclusions (of backgrounds from foregrounds), alignments (of the subject-objects of investigation with some scale and some agency, human or otherwise, taking readings), in line with some (implicit or otherwise) purpose. In a sense, data arrive late on the scene, not at its very beginning. No datum is innocent. The distinction between 'data' and 'capta' is foundational for viewing the nuPG as an operable and changeable medium, an instrument and a filter.

---

[10] A twenty-second etude *The Silver Scale* by Newman Guttman (psychologist) was the first composition to be synthesised in MUSIC I programmed in 1957

1.3 Xenakis and Granular Model: Composed Signal

In *Markovian Stochastic Music–Theory* Iannis Xenakis articulated granular synthesis as a procedure based on generating sequences of finite base functions in the form of time-frequency screen arrangements at a fixed rate (Xenakis, 1992). Xenakis' conception of the grain has been developed through reading and conversations with Abraham Moles who popularised the applicability of information theory to the arts (Moles, 1966; Bell, 2021). Xenakis proposed a model of a screen - a two-dimensional grid delineating a plane of intensity (vertically) and frequency (horizontally) into a set of cells. A screen is characterised by the density of grains in a cell, the shape of grains distribution, and the degree of order-disorder of such distribution. Xenakis employed the mehod in a work *Analogique B* (1958-59) composed of brief sinusoidal signals with a fixed rectangular envelope and a duration of 0″04. Agostino di Scipio observed that the screen model served Xenakis as an instrument linking the micro-temporal level of composition with the long-term design (Di Scipio, 1998, p.218). From the outset, a microsound composition involved issues of the relationship between temporal levels, material and form, and emergence or imposition of macro-structure. Emblematic for Xenakis, as well as many composers of that time, was an attempt at generating complex timbral entities and the overall form of the work from a particular procedure. In *Analogique B* Xenakis combined pre-composed screens using set theory operations. A Markov chain process was proposed as a control principle for sequencing these screens in time. Xenakis offered a synthesis process control based on principles not tied to acoustical or auditory parameters such as pitch, duration, amplitude, spectrum or spatial position. The abstract generative control model was also a characteristic of Xenakis' stochastic synthesis experiments from 1967-1978 (e.g. sound material used in 'La legende d'eer') and GENDY developed in the early 1990s. The stochastic process generated a waveform operating at the sub-symbolic time level of individual samples. The connection between abstract formalism and microsound synthesis is pertinent in the context of the emergence of new compositional methods.

A grain, In a basic form, is composed of an envelope (an amplitude) and a waveform (1.7). This model serves as a universal point of reference in developing a diverse set of techniques under an expanding category of microsound synthesis[11]. The envelope shapes each grain; generally, it is a statistical limit constraining the waveform in time. Early models of real-time granular synthesis used simple line-segment envelopes due to computer memory and computation power constraints (e.g. three-part line-segments of an envelope within Barry Truax's live granulator with the DMX-1000 computer (Truax, 1988)). The duration of an envelope can be linked with the frequency of the enclosed waveform (shorter envelopes accompany, e.g. high-frequency sounds). This is a special case of the wavelet transform (see 2.1). The waveform of grain can be fixed or time-varying. A fixed waveform is often composed of sine waves added together. It can also be extracted from recorded sound or generated procedurally (e.g. frequency modulation, chaotic oscillator or stochastic synthesis). Curtis Roads describes a simple grain generator in (Roads, 2004, p.90).

The parametrisation of the micro temporal scale enabled new compositional approaches. From the outset, Xenakis defined grains with a clear implication for the

---

[11] see (Roads, 2004) for a comprehensive overview of different microsound techniques. Recently Brandtsegg, Saue, and Johansen developed a unified model of particle-based synthesis techniques as an open-source computer program called *Hadron Particle Synthesiser*. The program allows for interpolating between all known particle-based syntheses and can be seen as a tool sonifying various models of parametrising the microsound domain

compositional practice:

> All sound is an integration of grains, elementary sonic particles, and sonic quanta. Each of these elementary grains has a threefold nature: duration, frequency, and intensity. All sound, even all continuous sonic variation, is conceived as an assemblage of many elementary grains adequately disposed of in time (...). A complex sound may be imagined as a multicoloured firework in which each light point appears and instantaneously disappears against a black sky. (...) A light line would be created by a sufficiently large multitude of points appearing and disappearing instantaneously. (Xenakis, 1992, pp 43-45)

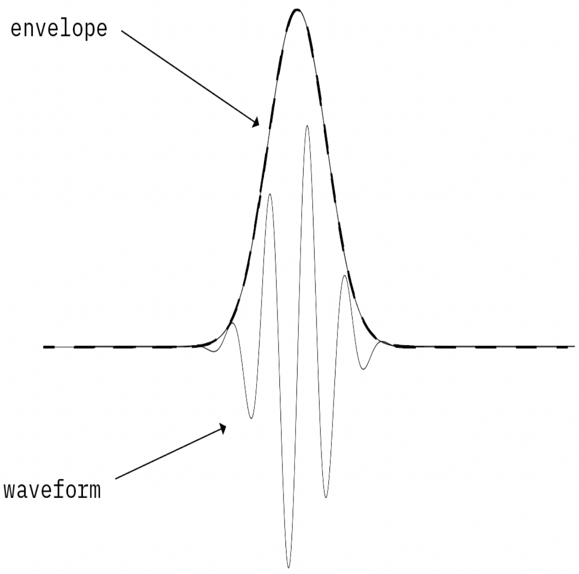


FIGURE 1.7: A composition of a grain. In its basic form, grain can be decomposed into an envelope and a waveform.

An essential consequence of the granular perspective is the conceptualisation of sound as a material object for manipulation. Sonya Hofer observes that such conceptualisation can be traced to Edgar Varese and the notion of 'organised sound' (Varèse and Wen-Chung, 1966) as well as Pierre Schaeffer's idea of 'sound object' (Schaeffer, 2017). The parametrisation of the micro-temporal level meant a possibility to compose the sound rather than compose with sound. It also meant an expansion of formal concerns down to the level of good material. As such, a discussion of microsound is intricately linked with the notion of timbre. The material approach to sound allowed to dismantle established categorisations such as pitch and harmony and form the sound as a set of elementary particles to be manipulated and sculpted. In *Atomic Music: Navigating experimental electronica and sound art through microsound* Hofer highlights "the tactile qualities of audio" and the sound as something mouldable and craft-able by reflecting on the work of composer Richard Chartier. Hofer quotes Chartier: "The sounds are treated with sculptural integrity. Each crackle, hiss, and tone is distinct, an instantiation of an auricular physicality that nonetheless proximally and often approaches silence" (Hofer, 2014, p.298). Throughout the composition with microsound, the sound alludes to and assumes a material presence.

The key features of the granular approach pertain to a non-standard sound synthesis model, as described by Stephen R. Holtzman. Holtzman coined the term *non-standard* to describe sound synthesis methods based on abstract concepts rather than acoustic, physical, or psychoacoustic principles. According to Holtzman, "the non-standard approach, given a set of instructions, relates them to another in terms of a system which does not refer to some super-ordinated model" (Holtzman, 1978, p. 1). Such a "system of relationships" is a formal description of the sound. Within the *non-standard* paradigm, "the computer acts as a sound generating instrument *sui generis*, not imitating mechanical instruments or theoretical acoustic models" (Koenig, 1978, p. 111). With the advent of computers and digital technologies, nonstandard sound synthesis systems marked a critical conceptual and aesthetic shift. The sound production became a compositional activity allowing for "the composition of timbre, instead of with timbre" (Brun and Brün, 2004, p. 189). This approach allowed composition beyond a practice centred on permutational combinatorics within a closed homogeneous system defined primarily by pitch, duration, dynamic marking, and instrumental timbre. From their respective etymologies,*composition* and *synthesis*, which are synonyms, one might think their difference has more to do with time levels than type. Agostino Di Scipio writes, “synthesis can often be thought of as micro-level composition” (Di Scipio, 1995, p.235). As suggested by Phil Thomson, nonstandard sound-synthesis approaches, in their “impulse towards the atomisation of musical material and control of that material on ever-lower levels,” can be seen as “microsound’s digital beginnings” (Thomson, 2004, p. 210).

Chapter 2

The Model of Instrumental Encapsulation: Interface <-> Script

Important for this project is an analysis of the nuPG's graphic user interface (GUI) and underlying programming infrastructure. To discuss both in conjunction problematises the notion of the interface as a surface and the code as its unaccessible deep structure. The differentiation between the surface and deep structures echoes Vilem Flusser's discussion on the notion of the interface as a "significant surface" - a two-dimensional plane that embeds or delivers meaning. Of course, not all interfaces are surfaces e.g. voice assistants. However, the model of the surface is useful as a probe testing various ways the sound data is represented, i.e., as a set of numerical values, binary code, a unit generator, a visual rendering of a waveform or an algorithm. None of these limits the applicability of the interface's notion to the graphic representation domain. In a common understanding, digital instruments have a specially designed interface that connects to the program's inner depths, a black box of their functionality; the code. In a broader sense, the interface should be considered an object of design beyond a particular representation domain. An argument developed within this chapter proposes a renewed perspective on the digital instrument interface, which places it at the intersection of the textual and visual modes of representation and control.

The chapter is divided into four sections. The opening section systematises the approach to the notion of the interface as a dynamic entity enabling control over the active system. The discussion looks back at the notion's roots in the experiments of James Clerk Maxwell with turbulent flows (Maxwell and Pesic, 2001) and James Thomson's work on solidification of cooling lava (Thomson, 1879). These early formulations contain a possibility for a perspective on the notion of the interface in the context of digital instrument design, which goes beyond the domain of graphical representation. The view presented in this section relates closely to the notion of object orientation as developed within the computer programming context. Section two of the chapter discusses object-oriented programming (OOP), its origins and features as exemplified within the design of the SuperCollider environment. The development of the technique of pulsar synthesis and its instrumental incarnations is inseparable from the object-oriented programming paradigm. The original Pulsar Generator (2001) program by Curtis Roads and Alberto de Campo, together with the first sketches of the technique of pulsar synthesis, as well as two further implementations (i.e., Keränen and Jeffs), were implemented in SuperCollider programming environment and utilised SuperCollider Server for sound synthesis. SuperCollider language is object-oriented, and its environment employs object orientation at many levels. Object orientation enabled a view of programming as a mediatory process and made it possible to include programming as a control. Section three focuses on analysing the original PG (2001) program by Roads and de Campo. The critical aspects of the analysis consist of graphic interface and modes of control over musical data. The study concludes with a brief discussion on two other historical implementations of pulsar synthesis: Pulsar Generator (2005) by Tomi Keränen and Particularity (2010) by Chris Jeffs.

2.1 The Interface as a Dynamic Site

The interface concept plays a vital role in the design and analysis of digital technologies (Andersen and Pold, 2018; Galloway, 2012). Thor Magnusson observed that digital instruments have an interface while acoustic ones are an interface (Magnusson, 2019). In both cases, an interface is an object of design. In a broad sense, the design process concerns the system's measurement and control. Barry Truax writes about the interface as a "communicational environment", inquiring this way on computer program's modalities of data input and output, data representation and program responsiveness (Truax, 1986). Branden Hookway points to the interrelatedness of control, measurement and knowledge acquisition when he writes: "The invention of the interface was the invention of an equivalence, where that which defines an interior condition or interiority is exactly and simultaneously that which allows the opening of that interiority to **knowledge** and **control**. An interface theory, then, would pertain to all situations and contexts where control is achieved through an active process that both defines and opens up an interiority" (Hookway, 2014, p. 79). The advantage of settling for such a fundamental interface definition lies in its generality and applicability to myriad contexts. Within the frame of the work with the New Pulsar Generator program, such a scope is seen as an opportunity for the interface to function as a dynamic rather than solidified way of control. An integral aspect of this objective is to exhibit strictly this fleeting perspective on the interface which operates between domains of visual, textual and auditory modes of control. Consequently, the approach pursued within this research does not treat the interface as a mere "effect on the surface" (Kittler, Von Mücke, and Similon, 1987, p. 102) but as an indispensable element in a dynamic relationship between the user and the program in its totality. These concerns can be found in the earliest conceptualisations of the interface.

The concept of an interface emerged within experiments on turbulent flows conducted by James Clerk Maxwell (Maxwell and Pesic, 2001; Thomson, 2016). Maxwell searched for a possible way of acting upon turbulence, of harnessing, directing and ultimately reversing it - in summary - the control and the interface was the site through which the turbulent system could be measured and acted upon; the invention of the interface was necessary for the management of complex systems. As pointed out by (Hookway, 2014) interface in this early formulation "denoted a dynamic boundary condition describing fluidity according to its separation of one distinct fluid body from another" (Hookway, 2014, p.63). As such, the interface plays a role of a threshold separating and differentiating between "unequal energy distribution within a fluid in motion". The interface as differentiation distinguishes and separates variation of "velocity, viscosity, the directionality of flow, kinetic form, pressure, density, temperature, or any combination of these" (Hookway, 2014, p. 59). James Thomson used the concept of the interface to describe a formation of columnar jointing out of the solidification of cooling lava. The interface in Thomson's formulation denoted the process of forming the "jointed prismatic structure" and its inscription in cooled solid rock. It is simultaneously used to describe "the internal processes by which a system is defined" and "the site where a dynamic process of forming may become visible, legible, knowable, measurable, and available for capture" (Hookway, 2014, p.63). A trait of these origins is visible in a shared understanding of interface as a doorway or a window - something that facilitates transition, a threshold between inside and outside. Francoise Dagognet sees the interface as this kind of threshold area where the input from the outside mixes with the output from the inside. This mixture, however, is not simple and transparent. It should rather be seen as a "fertile nexus" - a particular area with its autonomy, which yields various possible outcomes (Dagognet, 1982). A graphic user interface (GUI) and the text of a programming language can be considered a threshold between the inner workings of the program and its user-facing surface.

Rather than differentiating the interface based on the mode of representation, Armstrong proposes a classification of 'functional' and 'realisational' interfaces. A functional interface is orientated towards the ordered execution of a particular, well-defined task, whereas a realisational interface is orientated towards open-ended, playful encounters. As it transpires from the practice with the New Pulsar Generator, there is no necessary antagonism between these different design orientations; the development of a 'realisational' design still involves functionalising activities as particular abstractions and assumptions are made, and even the most functional interface could still afford a level of openness. A design process characteristic of current research integrates functionally orientated approaches in putting together open-ended possibilities for control. It can be said that such an integration combines "readerly" aspects of the functional with " writerly" possibilities of realisational approaches. A distinction between "readerly" and " writerly" — borrowed from Roland Barthes— has been introduced into the context of the interface by Florian Cramer (See: Barthes, 1980). A graphic user interface has readerly properties; it is "committed to the closure" (Barthes, 1980, p. 7). It establishes a fixed set of control points of synthesis parameters. The choice of these points and the way these are mapped to the synthesis model is a design choice. Cramer claims that UNIX's command line user interface is writerly in terms of its openness and encouraging the reader to become an active producer of "scriptable" code. For Cramer, these writerly aspects of coding break down the false distinction between the writing and the tool with which the writing is produced (and in terms of the computer, between code, data and processes). Cramer cites a 1998 essay by Thomas Scoville Scoville, " The Elements of Style: UNIX as Literature", to insist on the writerly aspects of programming by explaining how writers use language: " No literary writer can use language merely as a stopgap device with which to compose an artwork that is not in itself language — so, like in a recursive loop, literature writes its instrumentation." (Cramer, 2003, p. 102). The writerly aspect of the text-based approaches in digital instrument design highlights the programming language paradigm's fundamental role in the program's operation.

The nuPG program extensively uses techniques and methods of Just-in-Time (JiT) programming as described in (Rohrhuber, Campo, and Wieser, 2005; Rohrhuber and Campo, 2009; Nilson, 2007; Rohruber, 2010). JiT sound programming[1] offers new computational methods for computer music composition. As observed by Julian Rohrhuber and Alberto de Campo, "programming activity into the dynamic unfolding of sound (...) changes the relationship between formalisation and process"

---

[1] Alternative names for a similar concept, are: "live coding", "conversational programming", "on-the-fly-programming", and "interactive programming"

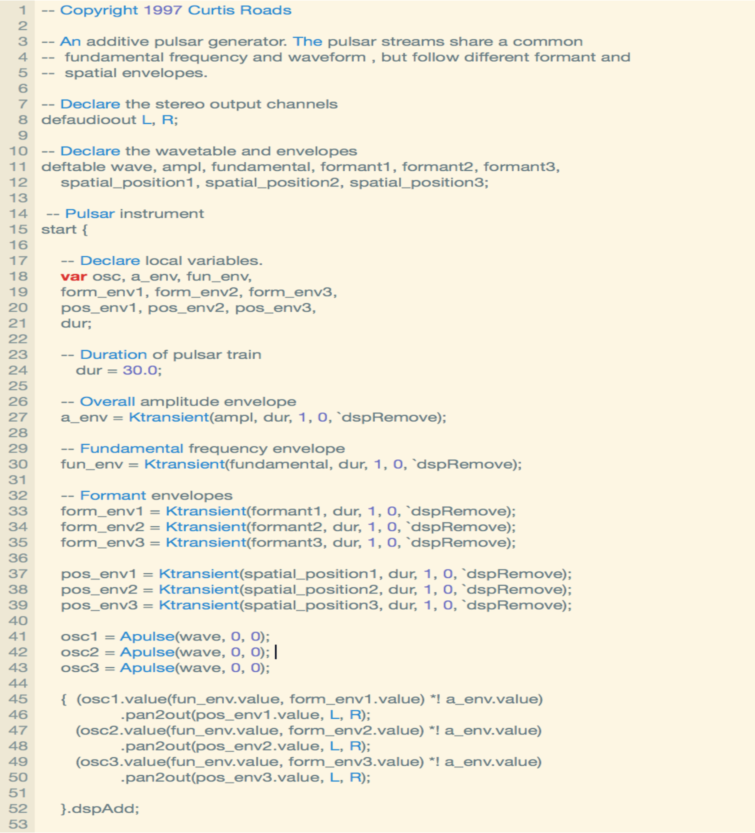
(Rohrhuber and Campo, 2009, p. 113). Rohrhuber and de Campo wrote a JiT extension for the SuperCollider language. They argue that within the JiT paradigm, the meaning of formalisation shifts from something that sets up the rules in advance to exploratory "dialectical interplay between plan and action" (Rohrhuber and Campo, 2009, p. 113). In the context of sound synthesis and composition, a consequence of such a fundamental shift is adhering to novel compositional and sound design techniques and aesthetics. The JiT perspective is constitutional for compositional methods where a text editor is used as an additional communication channel between the composer and the computer program. In the context of my work with the nuPG program, this approach can also be seen as a way of "unboxing" the computer program and enabling interrogation of some of its inner structures and processes. A set of compositional methods presented in 1 and 2 integrates the text-based approaches with graphic control modes. Crucial for the emergence of these methods is an analysis of the original PG (2001) designed by Curtis Roads and Alberto de Campo. The following section focuses on the description of the key elements of the program, its graphic interface, modes of control and modes of musical data representation.

2.2 Pulsar Generator (2001) by Curtis Roads and Alberto de Campo

Crucial for this research is an analysis of the original PG program designed by Curtis Roads and Alberto de Campo in 2001 at The Center for Research in Electronic Art Technology (CREATE). The PG (2001) program constituted the first instrumental encapsulation of the pulsar synthesis technique. It generalised and extended the basic model introduced in Section 1 with precise control over a collection of synthesis parameters. The extension to the basic model included control over pulsar train duration, pulsar train fundamental and formant frequency envelopes, pulsaret waveform and its envelope, amplitude and spatial path. Additionally, the program provided a set of pulsar masking algorithms that allow for the procedural omission of pulsars from a continuous train. The program extended the basic pulsar synthesis model by incorporating the additive model of the shared fundamental frequency with time-varying formant control, spatial path and amplitude (Roads, 2004, p. 147). The program allowed storage and recall of parameter data as a preset system. Once stored, data parameters could be interpolated using linear interpolation. The output could be monitored using a two-channel scoping mechanism.

An essential set of sources contributing to the analysis presented here constitute the PG program's manual and the program's source code.

The PG (2001) interface (2.2) shows a collection of wavetable views, each in its window. The colour scheme and basic layout of the interface were inspired by the look of audio measurement equipment by Brüel&Kjær. The program generated three simultaneous pulsar trains which share a fundamental frequency table. This was an intentional design choice by the authors. The design of the circuit with one fundamental frequency wavetable controlling an additive set of formant, panning, and amplitude wavetables dates back to 1997 and the first pulsar synthesis sketches conceived by Curtis Roads in SuperCollider programming language version 1 (2.1). Such a design choice enabled a generation of spectrally and spatially varying streams at lower processing costs due to a unified clocking mechanism synchronising all three trains by a singular fundamental frequency cycle. Currently, such computational limitations do not apply and I have tested a variety of design choices (e.g. one fundamental to one train, one fundamental to sixteen trains, and two fundamental frequencies alternating control over a set of trains). In the end, the nuPG program maintains the original choice as a functional balance between simplicity and complexity of control.



```
-- Copyright 1997 Curtis Roads

-- An additive pulsar generator. The pulsar streams share a common
--  fundamental frequency and waveform , but follow different formant and
--  spatial envelopes.

-- Declare the stereo output channels
defaudioout L, R;

-- Declare the wavetable and envelopes
deftable wave, ampl, fundamental, formant1, formant2, formant3,
   spatial_position1, spatial_position2, spatial_position3;

 -- Pulsar instrument
start {

   -- Declare local variables.
   var osc, a_env, fun_env,
   form_env1, form_env2, form_env3,
   pos_env1, pos_env2, pos_env3,
   dur;

   -- Duration of pulsar train
     dur = 30.0;

   -- Overall amplitude envelope
   a_env = Ktransient(ampl, dur, 1, 0, `dspRemove);

   -- Fundamental frequency envelope
   fun_env = Ktransient(fundamental, dur, 1, 0, `dspRemove);

   -- Formant envelopes
   form_env1 = Ktransient(formant1, dur, 1, 0, `dspRemove);
   form_env2 = Ktransient(formant2, dur, 1, 0, `dspRemove);
   form_env3 = Ktransient(formant3, dur, 1, 0, `dspRemove);

   pos_env1 = Ktransient(spatial_position1, dur, 1, 0, `dspRemove);
   pos_env2 = Ktransient(spatial_position2, dur, 1, 0, `dspRemove);
   pos_env3 = Ktransient(spatial_position3, dur, 1, 0, `dspRemove);

   osc1 = Apulse(wave, 0, 0);
   osc2 = Apulse(wave, 0, 0);
   osc3 = Apulse(wave, 0, 0);

   {  (osc1.value(fun_env.value, form_env1.value) *! a_env.value)
            .pan2out(pos_env1.value, L, R);
     (osc2.value(fun_env.value, form_env2.value) *! a_env.value)
            .pan2out(pos_env2.value, L, R);
     (osc3.value(fun_env.value, form_env3.value) *! a_env.value)
            .pan2out(pos_env3.value, L, R);

   }.dspAdd;
```

FIGURE 2.1: A SuperCollider 1 script defines a three-streams model of pulsar synthesis - in this original version called 'additive'. The pulsar streams share a fundamental frequency and waveform but follow different formant and spatial envelopes. Image courtesy of Curtis Roads (2021)

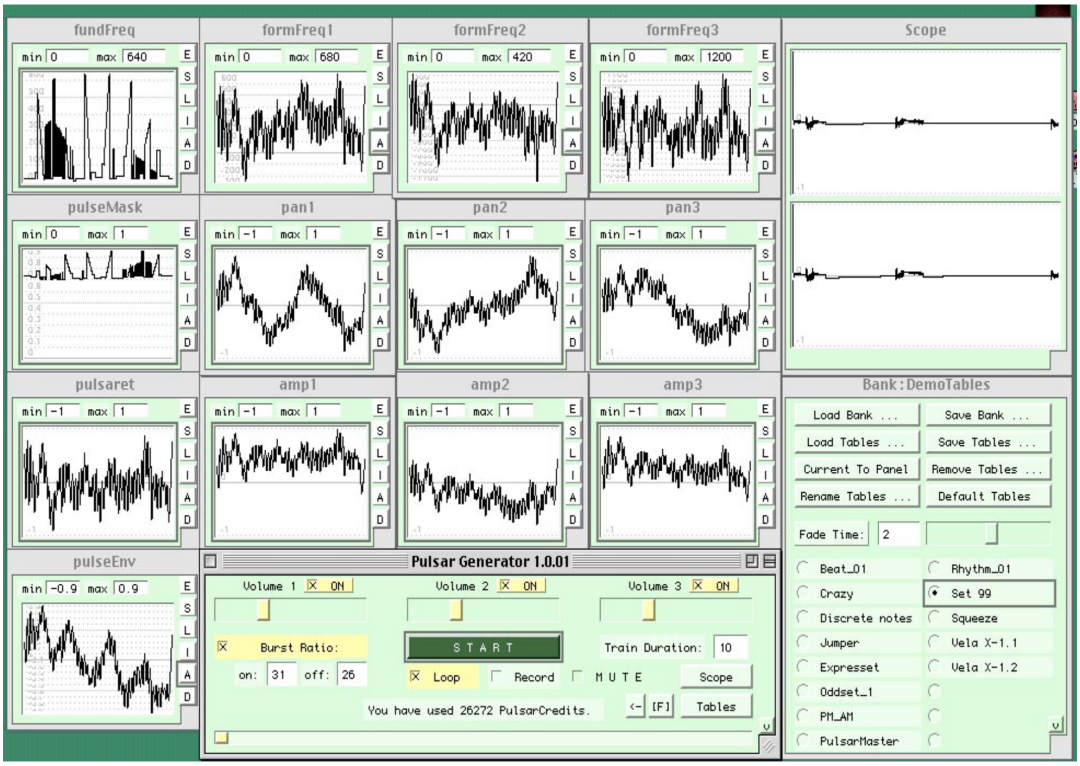
A set of graphic editors (e.g., 2.3) represented by a function of value versus time limited by minimum and maximum range. Each editor provides a set of functions for easy access and manipulation of graphs. The collection of functions includes: setting of table's range values ('E') via an additional editor window (2.4), saving the data of the table ('S')[2], loading of the data from an audio file ('L'), importing ('I') of the first 2048 samples of the first (left) channel of (almost) any audio file into the wavetable[3], rescaling ('A') of the visible range of the window so that the wavetable is centred, and resetting ('D') the range of the table to the default values.

The duration of the train can be set from the main control window (2.5). The time position on the train is displayed at the bottom of the window as a yellow bar. The user can set the playback to loop, reverse the direction ('<-') or freeze ('F') the reading in the current position. The 'TABLES' button opens TABLES BANK (2.6) menu from within which table data can be stored and recalled. The window also provided interpolation functionality allowing a linear interpolation from the current setting to the next one. The edge function, accessible through a small button

---

[2] The data by default was held as an audio file

[3] A multitude of formats are supported, including AIFF, WAV, SD2, and Sun (.au)



fundFreq min 0 max 640

formFreq1 min 0 max 680

formFreq2 min 0 max 420

formFreq3 min 0 max 1200

pulseMask min 0 max 1

pan1 min -1 max 1

pan2 min -1 max 1

pan3 min -1 max 1

pulsaret min -1 max 1

amp1 min -1 max 1

amp2 min -1 max 1

amp3 min -1 max 1

pulseEnv min -0.9 max 0.9

Fade Time: 2

● Set 99

Volume 1 ☒ ON

Volume 2 ☒ ON

Volume 3 ☒ ON

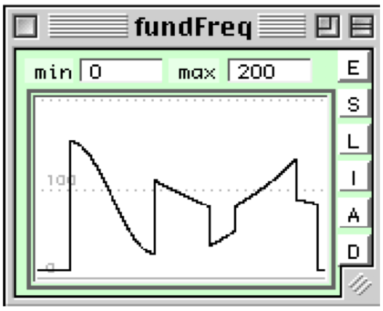
☒ Burst Ratio: on: 31 off: 26

Train Duration: 10

☒ Loop ☐ Record ☐ MUTE

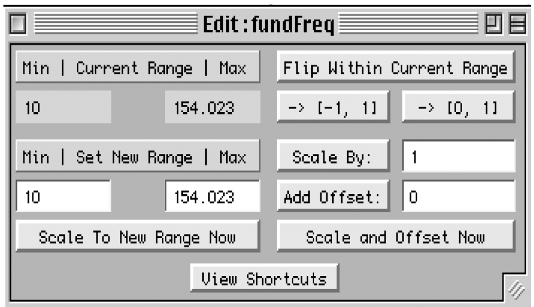
You have used 26272 PulsarCredits.

FIGURE 2.2: Pulsar Generator - a global view of the graphic interface. Notice a generalised table type control over pulsaret fundamental frequency with three separate tables for formant frequency, spatial position and amplitude. The colour scheme and basic layout of the interface were inspired by the look of audio measurement equipment by Brüel&Kjær



min 0 max 200

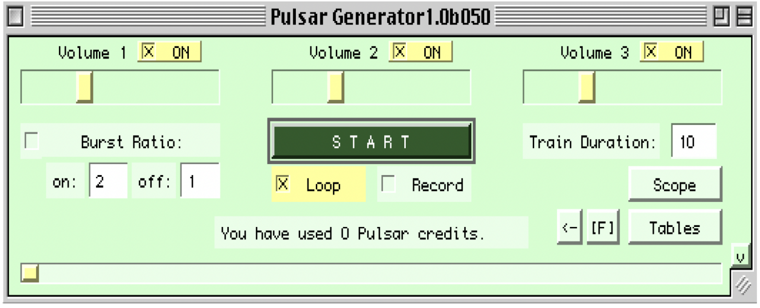
FIGURE 2.3: Fundamental Frequency interface window. The table describes a fundamental frequency function in a time scale within a range of minimum (min) and (max) values. The set of functions accessible through the editor included: setting of table's range values ('E') via an additional editor window (2.4), saving the data of the table ('S')[4], loading of the data from an audio file ('L'), importing ('I') of the first 2048 samples of the first (left) channel of (almost) any audio file into the wavetable[5], rescaling ('A') of the visible range of the window so that the wavetable is centred, and resetting ('D') the range of the table to the default values.



Min | Current Range | Max
10 154.023
Min | Set New Range | Max
10 154.023
Scale By: 1
Add Offset: 0

FIGURE 2.4: The editor of the fundamental frequency ranges. The editor allowed recalling of the default ranges, the minimum and the maximum values, adding an offset value (shifting horizontally) and flipping (vertically) the content of the table

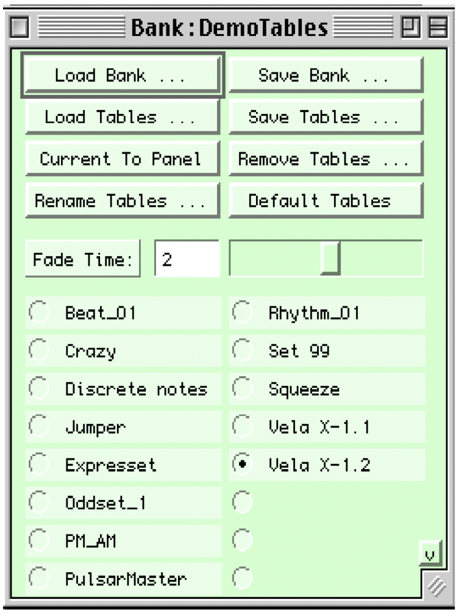
('v') in the window's bottom-right corner, allows setting an overlap value between consecutive pulses. A high edge factor means that a new pulse cuts off the previous one very quickly, creating a harsher sound; a low edge factor makes the transition more gentle. The edge factor functionality was implemented as an additional envelope encompassing pulsaret and envelope values and limiting these in time. The control over the edge parameter introduced a method of overlapped pulsar width modulation (OPulWM) into the circuit of pulsar synthesis (Roads, 2004, p.142).



Volume 1 ☒ ON
Volume 2 ☒ ON
Volume 3 ☒ ON
☐ Burst Ratio:
on: 2 off: 1
Train Duration: 10
☒ Loop
☐ Record
You have used 0 Pulsar credits.

FIGURE 2.5: The control window to activate/deactivate partial trains, control their volume, setting the duration of the train, the direction of playback ('<-'), freeze ('F') the position of playback, looping, recording and masking. From within the control window, the scope ('SCOPE') visualiser can be opened/closed. The 'TABLES' button opens the TABLES BANK menu. The yellow bar at the bottom of the window indicates the time position within the train. A small button ('v') in the bottom-right corner opens the edge factor control. The Edge controls change the way PG handles overlapping pulses. A high edge factor means that a new pulse cuts off the previous one very quickly, creating a harsher sound; a low edge factor makes the transition more gentle. The window also displayed 'PULSAR CREDIT', a counter of emitted pulsars.

A key idea behind the design of the Pulsar Generator was that each of its objects operates in the temporal domain through a set of functions, waveforms, envelope curves, and dynamic tables. This concept, also called "graphical synthesis" (Roads,



Fade Time: 2

○ Beat_01
○ Crazy
○ Discrete notes
○ Jumper
○ Expresset
○ Oddset_1
○ PM_AM
○ PulsarMaster
○ Rhythm_01
○ Set 99
○ Squeeze
○ Vela X-1.1
● Vela X-1.2
○
○
○

FIGURE 2.6: The TABLE BANK window allows storage, recall and editing of sets of table data. The window also provides control over user-defined interpolation between data sets

1996b, p. 329) links back to experiments in optical synthesis from the early twentieth century[6]. Composer Russell Haswell made the graphical and optical contexts of sound synthesis an explicit feature on the album 'ACID nO!se Synthesis' (Haswell, 2011). The album collects a set of live recordings performed with a modular synthesiser, computer, electronics and pedals. The composer was interested in the auditory aspect of the output and the visualisation with stereo/phase scope. Tracks 'Pulsar Anthem' and 'NO!se Pulsar' recorded using the Pulsar Generator program by Roads and de Campo feature characteristic pulsar sweeps repeating on a loop and rephrased iteratively.

The look of all graphical table editors of the PG (2001) program was equal with no regard for their temporal level of applicability. Additionally, all tables were stored in the memory as arrays with equal sizes of 2048 floating-point values. While the pulsaret waveform and envelope can be associated with the micro-temporal level, the rest of the editors (i.e. formant frequency, panning and amplitude) operate at a level between 1 second and 60 seconds. The size of the editor does not reflect different temporal levels of operation; instead, it offers a unified control model. This approach affects formal aspects of the compositional process as it allows a unified set of methods to be applied regardless of the time scale within which the object operates. The formal element of the table is an object of analysis developed in Part I, Chapter 3.

The Pulsar Generator program operationalised a multi-scale approach to musical data representation and control. At the level of sound microstructure, the user specifies the pulsaret waveform and the shape of the envelope, which together can be thought of as an elemental timbre of the program. At a higher level of organisation, the user operates a set of tables for fundamental and formant frequency, amplitude

---

[6] An in-depth discussion on optical synthesis is provided in Levin, 2003; Moholy-Nagy, 2004; Döbereiner, 2011 and Burns, 1995

and spatial position, drawing shapes-—lines, curves, and points-—on a value (vertical) versus time (horizontal) axis. Such uniform treatment of composition data and objects at every level mobilised a view of composition as a creative grafting across and between temporal resolutions. This profoundly affects how notions of musical material and form are conceived. We will return to these in Section (3). Through its explicit parametrisation of synthesis data and systematic approach to composition across multiple temporal levels - an attempt at fusion between micro (local) and macro (global) scales - Pulsar Generator relates to another historical piece of compositional technology - UPIC (Unité Polyagogique Informatique de CEMAMu) by Iannis Xenakis. The instrument operationalised a multi-scale approach to sound composition within a standard user interface. An incessant interpolation between temporal resolutions of the micro-, meso-, and macro-scales constituted a vital feature of the vision behind UPIC and Pulsar Generator alike. An analysis of the correlation between UPIC system, PG (2001) and the nuPG is the subject of an article I have written for a publication by the Center for Art and Media Karlsruhe (ZKM) and the Centre Iannis Xenakis (CIX) (see: (Pietruszewski, 2020b)). The creative potential of coupling sound materials generated with an original UPIC system and pulsar synthesis was explored by sound artist Russell Haswell in a composition *Full Circle (Pulsar RUNG UPIC)*[7]. As well as UPIC and the nuPG, in the work Haswell uses material generated with the Rung Divisions modular synthesis module by Fancyyyyy Synthesis[8]https://www.fancysynthesis.net/. Rung Divisions combines a universal shift register, a "divide by *n*" pulse divider, analogue noise, and several logic and binary operations. These functions synthesise an array of predictable and unpredictable digital signals at arbitrary time scales. The title of Haswell's work can be read as a reference to the circulation and transfer of concepts and synthesis models across time and practices. It also highlights material correspondence between pulsar synthesis and the historical synthesis approach in the form of UPIC. Haswell composed the work on the occasion of a compilation of pieces written with the nuPG program by various artists. To date, three collections have been published (Pulsar Scramble vol. 1 (2020), Pulsar Scramble vol. 2 (2020) and Pulsar Scramble vol. 3 (2021)) with works of over thirty sound artists from around the world (USA, Japan, Germany, Spain, Portugal, UK, Greece, Finland, Norway, Poland).

The design of the PG (2001) program emphasises micro- and meso-temporal levels of organisation over what can be called a macro domain of the whole composition. The program was conceived as a generator of sound material that would then be subjected to transformation, extension and up-mixing. This limitation is intentional, as Curtis Roads explains in a discussion on bottom-up composition strategies in electronic music: "The surface structure produced by a low-level algorithm can be quite *complicated*(in the sense of being non-redundant) while lacking a rich hierarchical structure of clearly perceivable beginnings, middles, endings, pregnant pauses, directional transitions, juxtapositions, singularities, and sectional articulations. These do not simply "emerge" out of most bottom-up strategies" (Roads, 2015, p.298). These limitations form an opportunity to treat the synthesis process as a stage in a chain of the compositional process rather than its end. A common practice is a division between stages: generation of the sound material and combination (setting it up against other material, mixing etc.). The PG (2001) program as a generator of

---

[7] see: `https://pwgen20.bandcamp.com/track/full-circle-pulsar-rung-upic` (Accessed 11.09.2023)

[8] `\unskip\protect\penalty\@M\vrulewidth\z@height\z@depth\dp`¸

sound material has been used extensively by composer Marcus Schmickler[9]. On an album, *Altars of Science* (Schmickler, 2007) material generated through the program features alongside other synthesis techniques (i.e. dynamic stochastic synthesis, frequency modulation synthesis and distortion product otoacoustic synthesis). The process of composition included transforming and editing synthetic material in a digital audio workstation (DAW). The output of pulsar synthesis is audible on *Track 6* of the album. From opening pulses, through 'scratching-like' phrases between 01:00 - 01:20, to the skipping and repeating formant sequences between 03:10 and 03:35 resembling the sound of Commodore's computer magnetic tape data storage device (Datassette), the track sounds out the staple timbres of the pulsar synthesis technique.

Another approach to working with the PG program is exemplified in the recording 'Pulsar W'glett' by Florian Hecker. Rather than using the output as a source for further processing, Hecker accepts raw and unadorned sound recordings of the program. A similar approach is featured on a number of tracks from the 'Pulsar Scramble' series where the nuPG program is often used as a "solo instrument". In this scenario, the present systems of both the original PG (2001) and the nuPG serve as points for formal sound development in the meso- and macro-temporal scales.

A powerful process provided already through the PG (2001) program is a set of methods for pulsar masking. In simple terms, the masking technique breaks up the pulsar train by introducing intermittencies (silence) into its regular sequence. In this aspect, the Pulsar Generator incorporates a subtractive synthesis paradigm. The program starts with a continuous series which then breaks up with intervals of silence articulating phrases and single pulsars. The PG (2001) program incorporates three forms of pulsar masking: burst, channel, and stochastic. The burst masking process is a classic electronic music studio technique. Pulsar burst masking interrupts pulsaret train at regular intervals, and it's controlled via burst to rest ratio: *b*:*r*. For example, a *b*:*r* ratio of 5:3 produces a sequence of five pulses followed by three periods of silence: 1111100011111000111111, and so on. At the infrasonic emission rate, the effect creates regular rhythmic patterns. At emission rate in the audio range, burst masking produces an effect akin to amplitude modulation on timbre (Roads, 2004, p. 151). Channel masking selectively omits pulsars in two or more audio channels. Stochastic masking incorporates weighted probability to choose a pulsar to skip. The likelihood is expressed as a curve over the whole duration of the pulsar train: when probability equals 1, a pulsar is emitted; if the value is less than 1, it has less possibility—a probability of 0 results in no pulsar being emitted. From a formal perspective, masking methods can be likened to the mathematical process of *sieving* as a formula for generating integer-based sets. Current research and compositional practice provide a generalised model of a sieve generator as an approach to pulsar masking. Further discussion and examples are provided in Part II, Chapter 1.

From the perspective my research, two additional implementations of pulsar synthesis were of particular interest: Pulsar Generator (2004) by Tomi Keränen and Particularity (2010) by Chris Jeffs. Both programs used SuperCollider language and the SuperCollider Server for sound synthesis. Additionally, each implementation can be linked with a body of compositional works.

The PG (2004) program by Tommi Keränen was written in SuperCollider version 3.5 and included a custom-written C++ UGen 'PulsarGenerator' designed by the author. The underlying design of the program draws upon the original solution

---

[9] The composer used the original PG (2001) by Curtis Roads and Alberto de Campo, as well as the implementation designed by Tomi Keränen. The material generated with both versions features on various releases by the composer

of three streams of formant, amplitude and spatial paths sharing one fundamental frequency, pulsaret waveform and envelope (See 2.1). Incorporating a text editor for table definition (i.e. as a mathematical function) is interesting from the point of control. The method called 'Filler' allowed the user to define all tables using the inbuilt capabilities of the SuperCollider script text editor. I view this functionality as a predecessor of text as an interface approach implemented within the nuPG program. Keränen's program was designed in a way that the colour scheme of the graphic interface changed randomly at each launch of the program. This added an element of novelty and playfulness to the experience of the program.

A composition *Sieve Bifurcation (nuPGPG(TK))* included on accompanying portfolio sets against each other material generated with the PG (2004) and the nuPG. A point of departure for the composition was unified table data loaded into both programs. The output of each program was separated spatially (the nuPG - left channel, PG (2004) - right channel). The goal was not only to sound out similarities and differences but also to perform processes of modulation and control, as these are possible within the design of both implementations.

The Particularity program consists of a new take on the pulsar synthesis paradigm. Designed in 2010 *Particularity* by Chris Jeffs is "a real-time granular synthesis system, along with a sequencer for some of the parameters, combined with a generator of animated visuals"[10]. The release notes of the program do not mention pulsar synthesis, but instead a set of techniques called granular synthesis. A pulsar synthesis can be classified as a branch of granular synthesis, as it is proposed by (Brandtsegg, Saue, and Johansen, 2011). Others, Curtis Roads, treat pulsar and granular synthesis as special cases of microsound approaches to sound synthesis. The Particularity program can generate up to 8 synthesis streams simultaneously and one visualisation window. The sound synthesis types include pulsar, frequency modulation, and granular synthesis. The frequency modulation synthesis provided within the Particularity program formed a basis of frequency and multi-parameter modulation approaches developed within the nuPG program. Both methods are further discussed in Part II, Chapter 1 in the context of the sieve as a subtraction method.

The object-oriented programming paradigm influences the design and functionality of all three instrumental implementations of pulsar synthesis. This chapter's following and concluding section focuses on analysing the object orientation within the SuperCollider environment. The section discusses the particularities of SuperCollider as an object-oriented language and looks at the specific design choices it enables. The development of object orientation and conversational approaches to programming are inseparable. From the outset, object orientation made it possible to rewrite a program “on the fly” during its runtime, which was previously only feasible in machine code. The section opens with a discussion about the relationship between computational formalisation and computational process. Introduced Just-in-Time paradigm entangles their temporal relationship mobilising the formalisation as a process operational in time rather than an outside-of-time plan preceding the action.

2.3 SuperCollider and Object Oriented Paradigm

As observed by Abelson and Sussman (1996) a programming language "is more than just a means for instructing a computer to perform tasks. The language also serves

---

[10] In June 2010, Particularity was named joint winner of the LoMus open-source music software competition organised every two years by the Association Française d'Informatique Musicale.

as a framework within which we organise our ideas about processes" (Abelson and Sussman, 1996, p.6). Within the object-oriented paradigm, concepts of computational process and the interface are implicitly embedded at the material level of the programming language. As such, object orientation echoes the above discussion on the notion of the interface that goes beyond a singular mode of representation.

On a fundamental level, the interrelatedness of notions of computational process and the interface allows their operationalisation. bject orientation is concerned with the computation opening to the operative engagement by the user. Casey Alt argues "that object orientation is not merely a way of thinking about computation (...) it is the medialisation of computation" (Alt, 2011, p.279). The process of medialisation entails a shift from computation viewed as "one continuous linear entity with a single point of control passing through each of its instructions one by one" towards a perspective on the program as a collection"of independent entities whose interactions dynamically emerge to approximate the end goals of the application" (Alt, 2011, p.292). More importantly, the view proposed by Alt signals that computation mediates between human-user and worldly effects and can act as a filter to perceive and experience the world differently. This perspective echoes a view of computation as sensorium as described in Chapter (0.5). In the context of computational methods for sound composition, Horacio Vaggione argues that an object-oriented paradigm should be seen beyond its computational realm as an embodiment of a specific compositional approach where concepts of encapsulation, inheritance and polymorphism become operational as practical tools applied within the process of composition (Vaggione, 1991). Polymorphism is the ability of different objects to evaluate the same message differently according to each object's particular context. An example of a polymorphic object within the nuPG program's design is the graph's object. As presented in section *Material and Form* (**??**). The graph can have a visual form, or it can be represented as a list of values in a numerical form. Vaggione highlights the polymorphic nature of objects: "The names of objects are handled in such a way that inside the common network, objects receiving identical messages, when activating their specific properties and methods, can on account of the latter produce uniquely different results" (Vaggione, 1991, p. 213). This feature within the nuPG program is mostly visible in how graph tables are handled. In their basic form, the graph specifies a list of values (2048, to be precise) in the range between 0 and 1. Only after passing a graph object into a context (e.g. pulsaret, envelope or sequencer) does the range of its values get updated to a new state. Also, how values from the graph are interpreted changes according to the context. In the context of the pulsaret waveform and envelope values are played once at the stipulated trigger frequency. This is the perceptual effect of playing them at a high emission rate. In the context of the sequencer, these values are interpreted as data points; each value from the graph is used as a multiplier for the corresponding parameter of the program. In the context of the compositional discussion, polymorphism is closely related to the concept of outside-of-time (see discussion in sections refGraphs and 1) objects. An object existing outside of time possesses a broad set of potential functions which come to realisation only once chained into a process with other objects. The ability to reuse and replace program components is related to the notion of polymorphism within an object-oriented paradigm. As Abadi and Cardelli (2012) points out, "A software component is reusable when it can be easily used in more than one context. For example, a module can be reused by importing it into several other modules, and a generic module can be reused by instantiating it with different parameters. In all cases, reuse entails the replacement of a component (or a placeholder) with another component in a context" (Abadi and Cardelli, 2012, p.

9). In object-oriented languages, we can specify two types of replacements: objects can be replaced by other things and methods acting on these objects can be replaced by other methods. A particular case of this reusability is inheritance. Inheritance means that objects can inherit properties and methods from other things (Krasner, 1980). The described above; the graph plays such a role in the design of the nuPG. A graph on the x-y axis can represent a pulsaret waveform, an envelope, a list sequencer values or a parameter development in time (e.g. recorded via slider record functionality); it inherits methods from SuperCollider object called ArrayedCollection, an abstract class, which is itself a subclass of SequenceableCollections whose elements are held in a vector of slots. Inheritance ignores the context of the object's placement and allows a universal set of methods to apply to it. In the case of a graph, example methods consist of methods such as .scramble, .reverse or .normalize.

The design of SuperCollider is multivalent. SuperCollider is three separate entities: the language (sclang), the synthesis server (scsynth) and the interplatform development environment (IDE). At the origins of SuperCollider development lies an idea of combining within one environment an audio-digital signal processing (synthesis and audio processing) and a high-level symbolic description of musical entities. This duality can also be seen in the fact that SuperCollider began as two languages: Synth-O-Matic, a non-real-time C-like synthesis programming language written in 1990; and MAX program-object Pyrite which contained the interpreter for the language which was extended and used in SC. SC integrated language interpreter, garbage collector and function library of Pyrite with synthesis engine and functions of Synth-O-Matic. A design of SuperCollider abandoned the duality of *instrument* and *score* present in historical programming languages for music, such as the Music-N family of languages. This two-fold implementation separating the language from the process of synthesis has major benefits. First, other applications can use the sound server for rendering audio; Second, it scales well to multiple machines/processor cores, i.e., *scsynth* can run on one or more autonomous machines; and Third, decoupling *sclang* and *scserver* makes both very stable. These two systems connect via the networking protocol OpenSoundControl (OSC). OSC is a content format developed at CNMAT by Adrian Freed and Matt Wright comparable to XML, WDDX, or JSON.[2] It was originally intended for sharing music performance data (gestures, parameters and note sequences) between musical instruments (especially electronic musical instruments such as synthesisers), computers, and other multimedia devices. OSC is sometimes used as an alternative to the 1983 MIDI standard when it desires higher resolution and richer parameter space (Wright, 2005; Wright et al., 2001).

As an object-oriented development, SuperCollider is also being built on programming units called objects. Objects are identified via a pointer to their data structures and are accessible through so-called methods that can be performed in a messaging action. All the knowledge about the object comes from its response to messages of their methods. For example, if an object describes a line on the graphical interface, the lines' coordinates are not automatically accessible. As such, they pertain to the object's privacy and must be accessed via a special method: called back, or invoked. This feature is called encapsulation. The object has an interior structure that is hidden in the program's context; you can only understand the object by its response to method messaging. This aspect of object orientation is called abstraction. The programming language represents an abstract model of computational units that allows to program and not worry about the details that are irrelevant to a given domain. The abstraction was a feature of programming languages before the emergence of object orientation. For example, abstraction provided by Music-N programming languages included a concept of a unit generator (UGen). Such abstractions made writing signal processing structures easier. An abstraction of the UGen introduced with MUSIC III was crucial for developing audio programming languages. UGen represents an atomic building block for generating or processing audio signals; it includes input user-specified arguments (parameters) and outputs a product of its operation Wang, 2007. SuperCollider provides a set of abstractions, including an extensive collection of UGens. Everything in SuperCollider is an object, including letters and numbers. The objectual model also persists in the design of the graphic user interface (GUI), as shown in the previous section (2).

Encapsulation is a concept related to abstraction and is crucial for the development of object orientation. Encapsulation is the process of completely bundling data elements and all functions that operate on them within a discretely contained coding structure or object. To encapsulate means hiding an object's details from the other parts of a program. The object can be used only through its access methods, which are carefully written to keep it consistent and secure. The most significant conceptual transformation implicit in encapsulation is the complete deconstruction of previous notions of a program. In machine, assembly, and procedural programming, a program is often viewed as one continuous linear entity with a single point of control passing through each of its instructions. Encapsulation completely reverses this notion by sundering the cohesiveness of the program into a number of independent entities whose interactions dynamically emerge to approximate the end goals of the application. Whereas data were previously reducible either to the time or space of the process and were open to any operations, data now become a primary structural element, as collections of data properties are enclosed along with all of the processes that operate on their states. Thus, instead of factoring and economy of process, there is factoring and economy of the necessary relationships between related data properties and their operation methods, even at the risk of replication and redundancy of processes or data across different objects. Encapsulation makes an object look like a black box: The box's insides are hidden from view. Controls are on the outside of the box. The user can change the operation of the box only by using the rules. However, it also signals a limit to operationalisation conceived as the opening of the technology. It can be said that rather than getting rid of the "black box" concept, operationalisation moves it to a different level. Perhaps a "black box" is a necessary mediation for operationalisation to occur.

While the hidden interiority of encapsulation can be viewed as necessary to the very notion of object orientation, it also poses an interesting paradox: What is the mechanism by which objects know what requests other objects can fulfil if they are thought of as self-contained, closed entities whose internal subjectivities are not exposed to the outside world? Through the concept of a protocol or interface, object orientation can both hide internals and present functionality. In contrast to the actual code itself, often referred to as the implementation, an object's interface is the public face of the object. It describes which methods within the object can be requested by another object, the format for making these requests, and the message (if any) the requesting object should receive as a response. Interfaces should not include all of an object's internal methods. Except for methods that must be declared public to enable outside interaction, object-oriented design dictates that all other methods should be declared private and only accessible by the object. As Gamma, Helm, Johnson, and Vlissides summarise it in Design Patterns: "An object's interface characterises the complete set of requests that can be sent to the object. Objects are known only through their interfaces. There is no way to know anything about an object or to ask it to do anything without going through its interface. An object's interface says nothing about its implementation—different objects are free to implement requests differently. That means objects having completely different implementations can have identical interfaces.” (Gamma et al., 1993, p.13). Object-oriented programming languages and environments strongly encourage programmers to develop a clear and carefully delineated interface for every object implementation as a standardised best practice. It is through these mechanisms that object orientation inserts the concept of the interface at the level of the code itself.

Within the SuperCollider environment, a decision to separate the synthesis engine (the Server) and the compiler (the Language) affected the development of the JiT programming paradigm as introduced in the previous section (2.1). If we consider a computer program as a plan (description) of a process (how something is supposed to happen), its status in the JiT approach is questioned. As noted by Rohrhuber, de Campo and Wieser, "If the program text is used as the representation of algorithmic processes with their causal relations, one encounters the problem that the process is happening in time while its description has been made in advance" (Rohrhuber, Campo, and Wieser, 2005, p.1). There is a temporal a-synchronicity between the user and the program. Changing the description of a program while it is running is problematic from a temporal perspective. On the one hand, the user imposes its temporal existence on the system. On the other hand, the program follows its history and iterative temporality. The user and the program engage in the process of concurring, simultaneous temporalities. The following chapter looks closer at selected objects of the nuPG program from the perspective of compositional practice. I chart a connection between the introduced above computational contexts and the traditions of computer and electroacoustic music.

Chapter 3

The Form and Formalisms

From a formal perspective, the basic posture of engagement with the New Pulsar Generator (nuPG) program is a model of flexible control over all parameters of the synthesis process. When working with the program, the user is confronted with a clean slate, a *tabula rasa*; the system is mute and requires input to generate sound. The initialisation of compositional labour, which requires the user to specify a set of values for objects – a pulsaret waveform and an envelope, which together form the pulsaret (see discussion in 1) — has its roots in the original Pulsar Generator by Roads and de Campo, which by default produced no sound. In essence, this can be read as an aesthetic and a conceptual stance: no audible output is possible without a composer or user's action. Such a design also focuses on actual objects the user is acting upon. In a historical context, such a stance relates to the so-called Cologne School of Electronic Music, centred around WDR and serial techniques developed by composers such as Karlheinz Stockhausen and Gottfried Michael Koenig. In its most radical guise, this approach entailed using no material that was "given" from the outside. No musical instruments, no recorded samples: Every feature, right down to the micro level, is the outcome of a choice. It was concerning this that Karlheinz Stockhausen wrote of "every sound" as "the result of a compositional act" (Stockhausen, 1963a, p.142) and Herbert Eimert coined the term "absolute composition" through which "real musical control of nature" can be asserted (Eimert, Stockhausen, et al., 1958, p.31) [1].

A practice of computational design I introduced in the previous chapter, allows us to put forward an alternative argument. The user of the program faces already-formed objects, which, while mute, already have a form and occupy a definite space within a computer's memory. Such an argument puts the idea of composition as working out of nothing. There is always some defining quality for the computational object, even if it's only its size. A composer Kaffe Matthews often starts a creative process by delimiting the duration of sound material or the RAM of the computer program to one minute (Rodgers, 2010, p.39). Time becomes the first compositional

---

[1] This comprehensive approach to composition is somehow problematic and needs to be read in a context of Stockhausen's - and also Karel Goeyvaerts's - broader spiritual program. Richard Toop has noted the relationship between a view of the sine tone as a "pure" sound element and Stockhausen's and Goeyvaerts's Catholic faith. In *Stockhausen and the Sine-Wave*, Toop writes: "For Stockhausen and Goeyvaerts, at this period, the notion of purity is not just musical, but theological. Both Goeyvaerts and Stockhausen were fanatically doctrinaire Catholic mystics: for them, music was justifiable only as 'a projection of a metaphysical datum'. More specifically, organised music (i.e. integral serialism) was intended as an image of Divine Perfection: the more rigorously organised music was in all its parameters, the more faithful its image of the harmonia Mundi and, indeed, the harmony of the universe. Suppose one accepts Fourier's theorem that any sound, however complex, can be reduced to an aggregate of sine-tones (the sine-tone functioning as the 'atom' of the acoustic universe). In that case, it becomes clear that the sine-tone is the 'purest' of all sounds, not only physically, but also theologically" (Toop, 1979, p. 383). See also in this context Melle Jan Kromhout's analysis of Goeyvaerts' attempts (and failure) at synthesising the purest sound (Kromhout, 2021)

choice and a formal constraint. While interrogating the design of the nuPG program, one becomes aware of the formal limitations the program proposes. The technique of pulsar synthesis delineates its essential elements, its objects which already pose specific formal characteristics. The pulsaret is a temporally limited entity, combining a waveform and an envelope. Within the nuPG program, this entity has a fixed size - typically 2048 floating point values. Such a choice constitutes a formal limitation composers are required to operate within. Furthermore, the pulsaret waveform is an abstraction. It too can be thought of as a construct, as a layered object containing many interacting smaller units - for example, samples in a bit resolution space. From a practical perspective, even the most fundamental compositional units can be further broken down into more minor elements that can become operational. Questioning the limit is a viable artistic position.

Consideration of form is closely related to notions of representation, continuity, discreteness and multiple temporalities. David Rosenboom points out this link, writing that "perceiving forms involves extracting differentiable entities from continuously emerging and evolving phenomena that can be stored and referred to as part of the fine structures of particulars nows, each with its synthesised past and projected future" (Rosenboom, 2021, p.35). Computer Music extended the temporal field of compositional activity and functionalised a multiple-perspective approach to musical form. All temporal levels are to be composed - at any time in the compositional process; we can intervene by synthesis and transformation at any timescale - from the macro scale of the whole work down to sections, phrases, sound objects, grains, and even individual samples. To approach musical composition from a multiscale perspective is to allow an interplay between inductive (specific and local) and deductive (general and global) thinking (Roads, 2015). To operate within a complete register of time scales is to shift the aesthetic focus away from discrete sound entities occupying well-defined time frames toward continuous and evolving objects of fuzzy boundaries and domains. These new objects do not conform to traditional musical forms. Curtis Roads highlights "cloud-like evaporative and continuously evolving morphologies" (Roads, 2015, p.145).

An analysis developed within this section aims at a focused discussion on the notion of form as conceived and mediated through the nuPG program. As observed within the circuit of the program (1.1), the formalisation process is carried out according to several modalities. The Fourier additive model synthesises harmonic cluster: a sort of outside-of-time shape waiting for its temporal actualisation. There is a formalisation of the pulsaret waveform: an envelope limiting the waveform in time and shaping its spectral content at once. The technique of pulsar masking is another example of a formal process; by subtracting pulses from a continuous pulsar train, the method synthesises phrases and patterns. A fundamental frequency table and sets of tables for formant frequency, envelope compression, spatial position and amplitude consist of a formal direction of expansion and a kind of limit, an enclosure of all possible parametrical transformations. Additionally, these modalities can operate on and in-between different temporal scales. Therefore, questions of form in the context of the nuPG program are inherently bounded, related to structural organisation and relationships between multiple temporal scales.

The section opens with a necessary, although brief, discussion on conceptualisations of the notion of form brought by early electroacoustic and electronic music (3.1). The proposed discussion focuses on two interrelated themes: continuity between tone and pulse and a renewed approach to material and form from studio and computer music techniques.

Connecting previous analyses presented in 1, 2.2, and 2.3 with the contemporary practice of computer music, the section develops a notion of form as an operative object. Within the context of work with the nuPG, questions regarding material and form unavoidably touch upon the digital realm of the technique. What is considered material and form seems to be functional to the level of operation. From the historical perspective, Karlheinz Stockhausen has noted the "subtle relationship between form and material", even going as far as stating that form and material are the same, one being a result of the other: "A given material determines its form out of its inner nature" (Stockhausen, 1989)[2]. In early electronic music, "everything was material"; the sound was referred to as material, as was the compositional method (e.g. serialism). This avoidance of a clear position regarding the definition of what constitutes the form has a root in the ambition to break with tradition, established styles, and cultural templates (Koenig, 1987)[3]. Agostino Di Scipio observes that the difference between material and form seems not ontological but functional; material and form are no longer distinct objects of different cognitive dispositions. Both materials and articulation of form are objects of design. In the context of the micro-temporal approach to composition Di Scipio states: "The conceptual dualism form/material — reflected in the clear-cut separation of composing-with-sounds vs composing-the-sound (actions bearing on formal development vs actions supplying the material for the former) — is weakened, if not withdrawn". Both material and form coalesce within "a process of timbre formation through time" (Di Scipio, 1994, p.135).

The object of the loop plays a key role within the circuit of the nuPG program and can account for the above complex contexts. Wolfgang Ernst sees the loop—as an activity of a periodic and repetitive waveform segment—together with the pulse train as a fundamental function of a mode through which the computer comprehends audio information[4]. The analysis developed throughout this section looks at the loop (3.2) as an object capable of multiple formal functions. The notion of the loop is a primitive building block of the technique. The pulsaret waveform and envelope consist of a short segment of a signal. The duration of these short segments lasts from milliseconds to several seconds. The waveform and envelope, combined into a pulsaret (see discussion in (1) on the construction principle of the pulsaret), are repeated in an iterative process according to a higher-level parameter loop: fundamental frequency table which equals pulsaret rate of emission per second. The duration of this outer loop can range from 0.5 a second to 2 minutes. A closer look at the loop reveals its variable function. On the one hand, the loop stabilises rather than imposes a form; it provides the goal of deformation and achieves it by interrupting it according to a final contour; it modulates the ensemble of already formed fragments. On the other hand, the loop is nothing but a formal container; it already contains a certain schematism that can direct the composition, a set of coherent operations included in the implicit state. From such a perspective, the loop plays a constant functional role: that of a structural germ possessing a particular directing and organising power and

---

[2] Gottfried Michael Koenig mentions Stockausen's remark that sound, once produced for a particular composition, "could never be used in any other place in any other piece" and that "sounds ought to be destroyed after a production was ended" (Koenig, 1987, p.1). This can be labelled as interdependent material and form fetishism

[3] In *Wordless Rhetoric: Musical Form and the Metaphor of Oration* Mark Evan Bonds considers form as "the unique shape of an individual work", which yields "generative" compositional approach. This tradition "considers how each work grows within and how the various work elements coordinate to make a coherent whole. In its most extreme manifestations, the generative idea of form makes no essential distinction between the form and content of given work" (Bonds, 1991, p. 127)

[4] For Ernst, such function correlates with a "sensation" of acoustic tone in the phenomenal sense, as described by Herman von Helmholtz in 1863 (Ernst, 2019, p.685)

simultaneously malleable object, not just passively deformable but actively plastic, capable of receiving the form.

Gilbert Simondon has observed this fundamental duality between two types of reality, the reality that receives the form and the one that is the form or conceals the form (Simondon and Iliadis, 2019, p.572). Within the new pulsar generator work, a gap between these realities remains ceaselessly marked and remarked. The discussion developed in the section (3.3) mediates Gilbert Simondon's reflections on form with the compositional notion of nested form developed by James Tenney.

As described previously (1), the nuPG program offers a seamless link between musical timescales of individual particle rhythms, periodic pitches, and the meso or phrase level of composition. The succession of pulsar micro-events produces rhythmic sequences or, when the density of events is sufficiently high, sustained tones, allowing the piece to pass directly from micro-structure to mesostructure. The preoccupation with these two aspects of sonic temporality opens up an interesting discussion related to the concept of form and time. Firstly, what is the functional relationship between the micro- and meso-temporal scales in the nuPG program? Secondly, what possibilities are there within the program to yield larger musical forms, beyond the meso-temporal scale?

3.1 Early Electroacoustic and Computer Music Conceptions of Form

One of the most characteristic and transversally shared views of early electronic music and modern sound technologies, in general, is the idea of being able to compose the sound instead of merely composing with sound. In the studio, sound composition became an independent dimension of the compositional method; in theory, the composer could compose at the atomic level of sound and apply various transformations impossible before.

The technology of electroacoustic studio also allowed a renewed approach to musical temporality: the sound recorded on the tape could be cut, split, dissolved into another sound, played in reverse, and repeated. As discussed in the section, (1.2) notion of *Acoustical Quanta* provided the necessary analytic and synthetic apparatus to conceive a seamless transition between tone and rhythm. The invention of Denis Gabor, the *Kinematical Frequency Convertor* (1946), allowed for slowing down and speeding up a sound without altering its pitch. The device was built around a sprocket optical recording system constructed from a 16 mm film projector. It operationalised the temporal continuity between tone and pulse[5]. Wolfgang Ernst observed that "slowing down high-frequency to low-frequency oscillations is not simply a matter of decelerating musical time perception but a radical transformation into the numerical regime of data" (Ernst, 2019, p.674). In this view, crossing into the infrasonic frequencies not only indicates a threshold between tone and rhythm but also makes audible the computable character of the sound. In the context of compositional practice, Manning makes a corresponding observation: "at sub-audio speeds, the components [of the tone] become events in their own right" (Manning, 2013, p.66). This view cuts across the fundamental sound model of pulsar synthesis - decreasing the emission rate (fundamental frequency) makes the pulsaret audible

---

[5] Beyond an interest of this thesis is an overview of similarly conceived instruments. It suffices to point at the *Phonogene* constructed by Jacques Poullin in the early 1950s and used in Pierre Schaeffer's studio. German company Springer developed a device called *Zeitregler* which was used in Herbert Eimert's *Epitaph fur Aikichi Kuboyama* (1963)

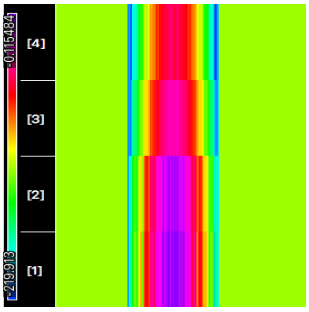


FIGURE 3.1: A singular pulsar composed of pulsaret waveform (first and twenty-fifth harmonics) and Gaussian-shaped envelope. See Appendix C for analysis parameters

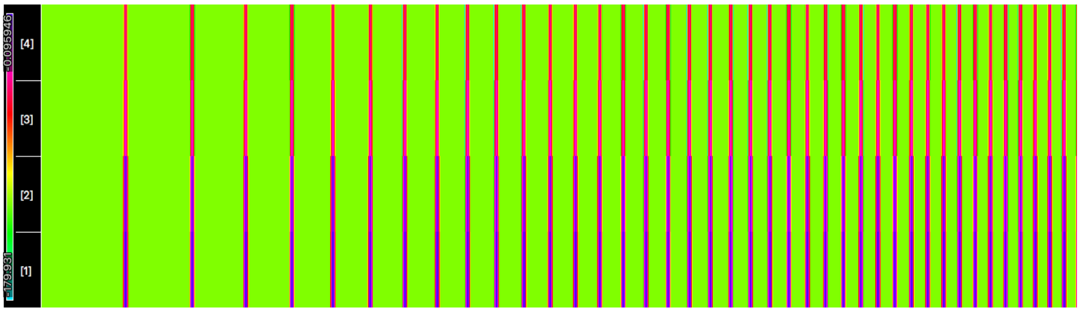


FIGURE 3.2: The analysis of accelerating progression of pulses

as a singular event. As discussed on page 35, an envelope of the pulsaret provides a spectral template of a pulsaret, the spectral property of a pulsaret train is its derivative. To hear the pulsaret in its singularity allows for building an audible relationship between discrete and continuous polarities of pulsar train emission. As a brief digression, let us focus on the following demonstration. 1 demonstrates an acceleration of pulsar emission rate:

Audio 1. nuPG gaussian envelope accelerando
Listen to the accompanying example: 'nuPG_gauss_accelerando.mp3'

A spectrogram analysis (3.1) displays three pulses with clearly delineated spectral shape (gaussian shape envelope). This envelope becomes compressed when the emission rate increases (3.2). The process can be seen as a demonstration of discussed earlier Gabor experiments in time-pitch shifting and wavelet's variable windowing (a discussion of the process in regards to compositions in the portfolio is provided on page 105). The technique of pulsar synthesis and its instrumental implementations operationalises a seamless back-and-forth form between a singular pulse and continuous tone. This form may be considered the most identifiable audible trait of the technique.

Going back to the context of early electroacoustic music, Karlheinz Stockhausen discussed the compositional potential of the pulse-tone continuity in an essay ... *how* *time passes...*[6]. For Stockhausen, the seamless transition between pitch and infrasonic pulses was not only of analytical importance but opened a possibility for systematic composition across multiple time scales[7]. Published in 1957 *... how time passes...* opens with an observation on the generality of the concept of a period, defined by Stockhausen "as an interval between two cycles". Stockhausen notices that the notion of a period can be applied to both rhythm (from 6 sec to 1/16th of a sec) and pitch (from about 1/16th sec to about 1/3200th sec). In summary, pitch and rhythm can be considered the same phenomenon, differing only in their respective time scales. Extending this argument into the micro-temporal domain, a note's tone colour or steady-state spectrum can also be seen as a manifestation of micro-rhythm over a fundamental frequency. In the macro-temporal scale, the entirety of the composition can be seen as a one-time spectrum of an entire duration. A logical consequence of Stockhausen's argument was a proposal of an absolute uniformity of temporal organisation[8]. Stockhausen elaborated this notion in (Stockhausen and Barkin, 1962) where the temporal field becomes divided into twenty-one discrete time octaves.

A direct consequence of studio technology was the ability to work with sound durations smaller than a hundredth of milliseconds. The tape editing was possible at the micro-scale as at a typical tape speed of 38 cm per second, a 1-cm fragment represented a micro-temporal interval of about 26 milliseconds[9]. An interesting example in this context consists of composition *Scambi* (1957) by Henri Pousseur. The initial material for the work was produced with the white noise generator, filtered using custom-built studio equipment. White noise can be defined as "all frequencies at random amplitudes over the full range of human audio perception" (Novati and Dack, 2012, p.129). The 'amplitude selector' designed by Alfredo Lietti "culled certain frequencies—those above a certain loudness threshold—from a more complex sound or a band of noise" (Iverson, 2017, p.374). Pousseur's approach is subtractive; rather than building a complex sound from a collection of pulses or sine tones, the composer starts with a complex (noise) sound source and subjects it to an iterative reduction process. The process produced various sound materials ranging from short pointillistic bursts to continuous noise textures. Pusseur composed the work in

---

[6] A fully-encompassing analysis of the text is beyond the scope of this work, for such see (Roads, 2004, p.72), (Koenigsberg, 1991)

[7] However the focus here is the essay of Stockhausen, it is essential to mention that the topic of continuity was also conceptualised decades earlier by Henry Cowell. In 1930 he described precisely this process: "Rhythm and tone, which been thought to be entirely different musical fundamentals, are related through overtone ratios. Assume that we have two melodies parallel to each other, the first written in whole notes and the second in half-notes. If the time for each note were to be indicated by the tapping of a stick, the taps for the second melody would recur with double the rapidity of those of the first. If the taps were to be increased greatly in rapidity without changing the relative speed, it would be seen that when taps for the first melody reach sixteen to the second, those for the second melody will be thirty-two to the second. In other words, the vibrations from the taps of one melody will give the musical tone C, while those of the other will give the tone C one octave higher. Time has translated, as it were, into musical tone" (Cowell and Nicholls, 1996, p.132)

[8] A lesson from Stockhausen's essay *... how time passes...* was to show how difficult it was to apply a proportional series developed parameter operating on one timescale (e.g., pitch periods) to another working on a different one (e.g., note durations) (Stockhausen, 1959). The boundary between rhythm and pitch is perceptual, and as such, it is tied to the affordances and limitations of the human body. As noted by Justin London, 2012, the human hearing has upper and lower limits - durations smaller than 100ms are too fast to be perceived as separate objects and are fused into a continuous stream. Between the constant tone and the metered rhythm stands a zone of ambiguity. This infrasonic frequency domain is too slow to form a constant tone but too fast for rhythmic definition (Roads, 2004, p. 19)

[9] as pointed in discussion on pulsar synthesis as a micro-sound technique (1), in the early electroacoustic studio practice, various composers addressed directly micro-temporal scale. We can list Iannis Xenakis' *Concret PH* (1958) and *Analogique B* (1959), and Henri Pousseur's *Scambi* (1957)

sixteen sequences and envisaged the listener to "build" the piece, resulting in a new rendering each time. *Scambi* in Italian means 'exchange': as the listener or realisation could exchange sixteen sequences, layer them differently and play them back in a different order (Dack, 2017; Cochrane and Gatherer, 2020; Pousseur and Behrman, 1966). Within the compositional work with the nuPG program, both additive and subtractive approaches are part of a dynamic process unfolding at timescales from micro to macro-temporal. Applied in works ['Sieve Decantations i' and 'Sieve Decantations ii'] technique of filtering in conjunction with additional transformations (e.g. modulation, spectral spatialisation, spectral muting, reversed in time) was loosely based on Lietti's "amplitude selector". The implementation within the circuit of the nuPG divides the audio pulse train into two, four or eight bands. When summed together, each of the newly created band segments results in an exact input signal[10].

The practice of electroacoustic studio introduced the montage as a compositional method. In the compositional realm, montage denotes the arrangement of predefined units in time (horizontally) and space (vertically)[11]. In the realm of microsound, the process of montage can be called micro-montage, where the term "micro" refers to the timescale of these operations (usually less than 100 ms)[12]. A micromontage as a compositional approach was an essential component of the compositional practice of Horacio Vaggione's[13]. In micromontage, the composer extracts short grains—loops—from sound files and rearranges them in time. As noted by Curtis Roads: "this detailed manner of working" can be thought of as "a musical equivalent of the pointillist painter Georges Seurat, whose canvases portray a dense sea of thousands of meticulously organised brush strokes" (Roads, 2015, p.168). Within the nuPG table editor, the user can draw a shape with the cursor of a mouse, load it from a sound or text file, and define it as a function (e.g. polynomials). Forms can be further transformed using interpolation and re-synthesis functionality (2.1). The term micromontage and the activity of composing at the microscale also delineate the sequencing of these small fragments into longer-duration phrases. This aspect of the micromonatge within the program is controlled through a set of table sequencers. The technique of montage played an important role in hip-hop and techno music. Both rely on repetition and transformation of prerecorded short samples or sound phrases. As observed by Imani Perry, the hip-hop DJ creates "new rhythms or beats between and within records". The new phrase emerges from pulling and scratching the record "back to the same place, again and again, to repeat the riff or a set of words" (Perry, 2004, p.70). The nuPG's Scanner method implements scratching-like functionality to interact with tables of data (A.5).

The notions of repetition, looping and scratching reframe the discussion on temporal continuity and rhythm as an emergent property of sound manipulation. In a limited view, any structure or pattern evolving in time can be considered a rhythm. A common understanding of rhythm focuses on its quantified character. Usually, we can distinguish the meter and count its essential elements. The scope of rhythm, however, goes beyond this simple conjunction of onset time and duration. The rhythm "subsumes the undulations internal to a sound, such as accents, swells,

---

[10] See also discussion on band splitting in the context of 'Sieve as a Method of Subtraction 1.7

[11] The montage as a technical tool is not limited to electroacoustic music only. Benoit Gibson has covered a vast set of montage techniques used by Iannis Xenakis in his instrumental music (Gibson, 2011) and also writing of Horacio Vaggione is a valuable resource in this context (Vaggione, 1994; Vaggione, 1991; Vaggione, 1993)

[12] See the corresponding discussion on beginnings of micromontage in section 1.1

[13] Vaggione's works such as *24 Variations* (2002) and *Points critiques* (2011) are examples of the micromontage

vibrato, and tremolo, which can be generalised to fluctuations in any parameter" (Roads, 2015, p. 137). The rhythm also unfolds on multiple timescales. As Cooper and Mayer observed: "As a piece of music unfolds, its rhythmic structure is perceived not as a series of discrete independent units strung together, but as an organic process in which smaller rhythmic motives, while possessing shape and structure all their own, also function as an integral part of a larger rhythmic organisation" (Cooper, Cooper, and Meyer, 1963, p.16). From the generalised perspective, rather than as a separate entity, the rhythm appears as a result of operations - oscillations, modulations and density changes - in pitch, amplitude, timbre and space. The perspective resonates directly with the mode operative within the nuPG program, particularly its fundamental object: the loop. Edgar Varese wrote about the rhythm as "the generator of the form" (Varèse, 1967, p.3). The object of the loop is a key form generator within the nuPG program.

In the context of Elektronishe Muisk, the notion of the loop was inseparably connected with a vision of multi-temporality. In *Kontakte* (1960), Stockhausen designed rhythms by splicing together pieces of tape with various pulse patterns and making tape loops of the beats. Long loops were wrapped around microphone stands to keep them taut, a typical practice that day. Several loops ran simultaneously for long periods, and a mixed result was recorded. Stockhausen noted: "So loops were running everywhere, and you could see it between the glass windows of the studios. Finally, I used fast-forward on the tape recorder to accelerate the tapes, so they were already four or five octaves up. The result went up four octaves—so then I was up eight octaves—until I finally got into an area where the rhythms are heard as pitches and timbres" (Stockhausen, 1959, p.18). The account of tape loops occupying physical space contrasts with the loop as outside of time form developed later in the thesis (3.2). The section between 16:56 and 18:26 of *Kontakte* (1960) features an iconic transition from rhythm to tone and back to the rhythm: "like an aeroplane falling out of the sky, of a rasping, twisting stream of sound. This strident intruder spirals down until its pitch characteristic becomes transformed into an ever-slowing succession of sharp clicks" (Manning, 2013, p.66). Stockhausen contextualised this phenomenon in writing: "If the rate of beats is gradually increased beyond the time constant of the filter and the limits beyond which the ear can no longer differentiate, what started as a rhythmically repeated note becomes continuous. We see a continuous transition between what might be called durational intervals characterised as pitch levels" (Stockhausen, 1958, p.14). *Kontakte* (1960) realised with assistance from Gottfried Michael Koenig (Supper, 1997) features filtered pulse trains[14]. In a narrow band, the pulses resonated at a specific pitch. If the pulse train was irregular, the infrasonic impulses generated metrical rhythms[15]. By transposing these rhythms - via tape speed manipulations - up into the audible frequency range, Stockhausen was able to build noises from aperiodic impulse trains. The technique of recirculating tape feedback loops was developed in 1951 by Werner Meyer-Eppler, Stockhausen's teacher (Ungeheuer, 1992, p.121). The Cologne studio of the Westdeutscher Rundfunk (WDR) was equipped with a device for spooling loops of up to 150 meters,

---

[14] A widely used equipment in the electroacoustic studio was the Analogue Impulse Generator. The Generator was initially built as laboratory equipment where it served, for example, to measure a system's impulse response, generating a continuous train of pulses at a given rate. As described by Roads, an impulse is "a discrete amplitude-time fluctuation, producing a sound that we hear as a click" (Roads, 2004, p.68). A circuit of a square wave generator called a multivibrator formed the basis of the *Dynaphone* (1928) instrument developed by Rene Bertrand

[15] A comprehensive description of the application of impulse generator in early electronic music can be found in Kaegi (1967)

corresponding to over six minutes at a 38 cm per second tape speed. A tape loop could articulate either an indefinitely sustained tone or a repeating pattern[16].

The notion of the loop relates closely to discussion on feedback systems as appeared in contexts of systems theory (Bertalanffy, 1969) and cybernetics [17]. These two theories from the 1940s strongly emphasised the relevance of closed information loops in organised structures. A minimal definition of feedback takes into account the configuration of a system, provided with input and output, in which some kind of transformation occurs, where the work is connected (fed back) to the information after a delay (Sanfilippo and Valle, 2013). An in-depth analysis of feedback as the compositional principle is beyond the scope of this thesis; for such, see (Valle and Sanfilippo, 2012; Sanfilippo, 2020; Di Scipio, 2003). From the perspective of current research, the notion of feedback highlights the importance of the processing channel as a transformation zone between the input and the output. The loop in this context relates to a possibly infinite chain of signal processing[18]. In the practical domain of composition, Roland Kayn explored a variety of auto-regulating systems based on feedback loops both as formal models for instrumental arrangement and as networks of analogue signal generators for electronic music (Patteson, 2012)[19] [20]. David Tudor employed feedback loops within his instrument circuits and compositions. In work *Toneburst(1975)*, initially composed for Merce Cunningham's choreography, Tudor chained a set of commercial and custom-built electronic instruments into a feedback network ([see diagram for the work][p.397](Nakai, 2021)). In work *Pulsers (1974)* Tudor used a complex modulator designed by Gordon Mumma. The work used no primary input signal; phase-shifted feedback is the trigger source. Tudor's view on boundaries between the instrument's design, the synthesis circuit and the composition resonates closely with the compositional practice presented within this research project. The nuPG program is at once an instrument, a generator of compositional forms, a research environment and an artefact. The following section focuses on a detailed account of the loop and its function within the New Pulsar Generator program is given in the following section.

3.2 The Loop

The loop as an independent and operative unit of composition emerged within techniques of the closed grove and tape transformation of early electroacoustic studio. For a composer of electronic music, the loop often consists of the primitive unit of compositional material—a musical vocabulary of Mark Fell centres around systematic interrogation of computer-generated temporal loops. A composer describes a basic procedure in *Structure and Synthesis. The Anatomy of Practice*: "For example, a

---

[16] Jean-Claude Eloy's composition *Shanti* (1974) using the functionality of the studio extensively

[17] weiner1948cybernetics

[18] As observed by Sanfilippo and Vale 1997 From a theoretical point of view, we can think of a zero delay feedback loop as a system whose fundamental frequency is infinity,- in practical terms, any implementation and performance of a feedback system imply a delay more significant than zero (Sanfilippo and Valle, 2013, p.12)

[19] A catalogue of works by Kayn is being recently digitised and reissued `https://rolandkayn.bandcamp.com` bringing more attention to this often unrecognised body of work

[20] Nicolas Schoffer (1954) was the first in art history to advocate the use of cybernetic systems in his "spatial-dynamic" works. In 1955 he created the first physical installation implementing an automated mechanism, the CYSP 1. Installed in Paris, this cybernetic sculpture included some computational capabilities thanks to the technology offered by the Philips company. Provided with photocells and a microphone, CYSP 1 could sense the environment—including itself—and sonically react by playing back sounds composed by French composer Pierre Henry

list of values such as 3, 5, 7, 4, and 2, multiplied by a base value (let us say 150) to create intervals of 450 milliseconds (ms), 750ms, 1050ms, 600ms, and 300ms, with a total loop length of 3150ms(a little over 3 seconds). Using this procedure, if one loop is changed, the whole lop length is either extended or shortened" (Fell, 2022, p.265). Fell's music is indebted to techno and electro. An emphasis on the loop, and the relationship between a closed set of values and their multiplicator, shifts compositional concerns towards the generation and transformation of numerical values. In the *INTRA* Fell applied a pattern-generating system in the form of two nested loops. The inner loop of five units forms a repeating sequence, while the outer loop functions as a reset. For example, with the inner loop consisting of five-time units, with a value for each unit equal to half a second ($\frac{1}{2}$), one inner loop would equal two and a half of a second (5 times $\frac{1}{2}$ equals $2\frac{1}{2}$). The outer loop is independent of the inner one and acts as a reset. If the outer loop value equals eight-time units, then the inner loop would reset after four seconds (8 times $\frac{1}{2}$ equals 4). Within *INTRA* every percussion player followed an equal inner loop sequence; however, each player's outer loop (the reset) was different. This simple procedure allowed stable patterns which occasionally shift in phase with one another (Fell, 2022, p.267). Within the work with the nuPG, the method of 'group offset' allows exploring a relationship between temporally shifted sequences (1.10). Historically, a process of diverging a phase of interleaved loops was a standard compositional method for minimalist composers such as La Monte Young, Steve Reich, and Tom Johnston[21]. Recently, a British composer, Lee Fraser, developed a systematic compositional practice with trainlet synthesis (See Roads, 2004, p.129) which operates with enveloped particles of varied duration and density. The technique generates a synchronous cloud of trainlets. With over twenty parameters to control, the loop is employed in the technique's circuit as a control model and iteration over user-provided input values. Lee Fraser employs trainlet synthesis on an album *Cor Unvers* (Ge-stell, 2018) (Fraser, 2018).

Within hip-hop music, the loop also played an important role. For example, J Dilla (James Yancey) developed a systematic technique of drum loop sequencing and offsetting beats on a micro-temporal scale. His unique approach can be heard in Common's 'The Light' which features a loop that starts in the middle of the phrase from Bobby Caldwell's 'Open Your Eyes' or in Slum Village's 'Get Dis Money' which samples a seven-bar loop from Herbie Hancock's 'Come Running to Me.' Hip-hop is particularly responsible for popularizing the idea of music sampling–using fragments of existing records or other sound-sources in a short form, or for the creation of loops as the basis for new musical tracks. The ideas typical of hip-hop style in music and culture originated in the mid-1970s, when Jamaican-born DJ Kool Herc instead of playing current hits in discos and park parties, sought out more obscure records in order to play only their instrumental breaks, or the "hottest" parts of them, over and over again, until they sounded like new records (breakbeats). The loop also formed a basis of techno music, where a repeating sequence is a key characteristic of the genre. Techno emphasised a "four on the floor" rhythm with a kick drum playing precisely on all four beats, which were then repeated in a loop. Techno tracks often have no musical sections longer than a single bar, generally a concise one. The use of step sequencers (a sound organising device) is a commonly shared technological feature that could help to define Techno. Instrumental in the establishment of

---

[21] In the analysis of the temporal aspect of minimalist music Jonathan Kramer coined terms "moment time" and "vertical time" (Kramer, 1988, pp. 50-52) to denote music composed of static loops rather than a succession of events

the style were digital drum machines such as Roland TR-808 or Akai MPC60 developed together with Roger Linn. Techno can be characterised by grid-like, inhuman rhythms repeated without or a small timing variation, and an extreme focus on units of a sound placed in a measured loop(Feser, Pasdzierny, and Kaul, 2016)[22].

The loop as a formal object within a composition has an attractive property. The loop “moves” - repeats - but is also “still" - frozen. It embodies a tension between fluidity and fragmentation. Iteration is a particular type of looping where the result of one pass through the loop influences the development of succeeding passes. The simplest example of an iterative process is nothing more than counting: beginning with an initial value of 1, adding 1 to this (that is, incrementing it) to produce 2, performing the same incrementing operation again to yield 3, and so on. If the program goes to a given limit, for example, 10, the loop halts. The context of iteration and counting was embedded in the original PG (2001) program by Roads and de Campo in the form of ’pulsar credit’. ’Pulsar credit’ was a simple counter which kept a log of the number of pulsars emitted and displayed it to the user. The iteration can also consist of a point of inference and control. In the context of the new pulsar generator, at each iteration pulsar synthesis circuit updates its parameters. The iteration acts like a clock system providing synchronicity and yielding singular pulsars in time—additionally, the repeated action of the loop functions as an object of both analytic and synthetic operationalisation. The experiment of segmenting and repeating fragments of tape contributed to the invention of Denis Gabor’s Kinematical Frequency Converter, which demonstrated that the time and pitch of sound signals could be controlled independently (see discussion on Acoustical Quanta in (1.2 and Wavelets in (2.1)).

As proposed here, the loop is not just any form; it already contains certain schematisms that direct the organisation of musical material. The loop provides a set of coherent operations contained in an implicit outside-of-times state. In a raw format, the loop possesses a contour; it can be described as having a particular form or distribution (3.3). It does not indicate details regarding its temporal unfolding, localisation in time - what comes before and after it - and duration. However, it can still be described: a shape can be symmetric or a-symmetric, have one or several peaks, be skewed, etc. These are simple descriptors from data science that do not infer any information about precise values but still give information about the general statistical distribution. At an even more elementary level, when such a shape is stored as a digital signal, it also exists synchronically outside of time. Wolfgang Ernst has described the shift from outside to in-time as "un-freezing". In an article *As Slow as Possible? On the Machinic (Non-)Sense of the Sonic Present and Digital Indifference toward Time*, Ernst writes: "The elementary unit of a technological being-in-time is the time-varying signal. When not in motion, a phonographically recorded acoustic signal is not in its signal state but frozen in a state of graphic storage. It becomes an operative media diagram only when turned back into a time object once again, un-frozen and transduced by the movement of the apparatus and the sonic pickup. The phonographic record waits for the mechanical player to unfreeze its signals in a technological act of representing." (Ernst, 2019, p.684)[23]. Iannis Xenakis developed

---

[22] A research group ’Techno Studies’ at Berlin University of the Arts (UdK) provides a scholarly reflection on aesthetic, cultural and social aspects of techno. For more information see: `https://www.udk-berlin.de/universitaet/fakultaet-musik/institute/institut-fuer-musikwissenschaft-musiktheorie-komposition-und-musikuebertragung/musikwissenschaft/veranstaltungen/archiv/techno-studies/`

[23] Ernst metaphor of a frozen state of a signal is borrowed from François Rabelais’s *Gargantua et Pantagruel* (1532), a boatsman tells of a frozen lake where the noise and cries of a battle have crystallised

the notion of outside-of-time to its limits and stated that "music is outside of time and time serves only for it to manifest itself" (Xenakis and Varga, 1996, p.83).

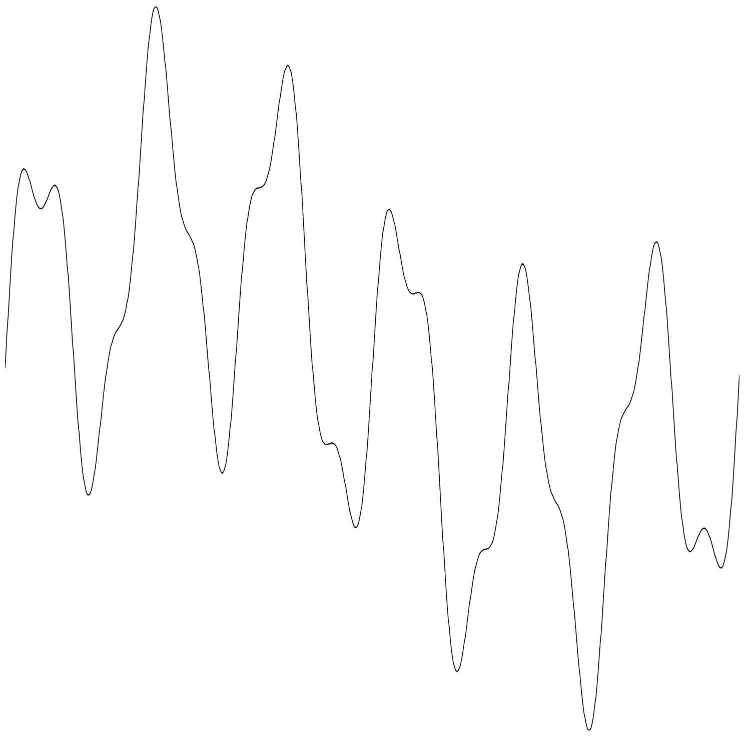


FIGURE 3.3: An example of a shape without coordinate values (XY) does not allow precise temporal and spatial indexing. However, the shape's general contour can still be described: a shape can be symmetric or asymmetric, it can have one or several peaks, it can be skewed, etc. These are simple descriptors from data science that do not infer any information about precise values but still give information about the general statistical distribution.

An engagement with sound synthesis complicates such a view in a way that composition not only happens in time but is also capable of producing its temporality. The loop-form within nuPG functions at first as a proposal, or to use Michael Koenig's term "variant-form" (Koenig, 1987). Whether as a graphic shape or as a description of an algorithmic procedure - its existence is potential. It exists "outside of time", waiting to be hooked up to the actual synthesis model and sound out.

---

in the icy air, waiting to be released in the warmth of springtime. This is fictitious, however plausible, the anticipation of phonographic sound recording and replay

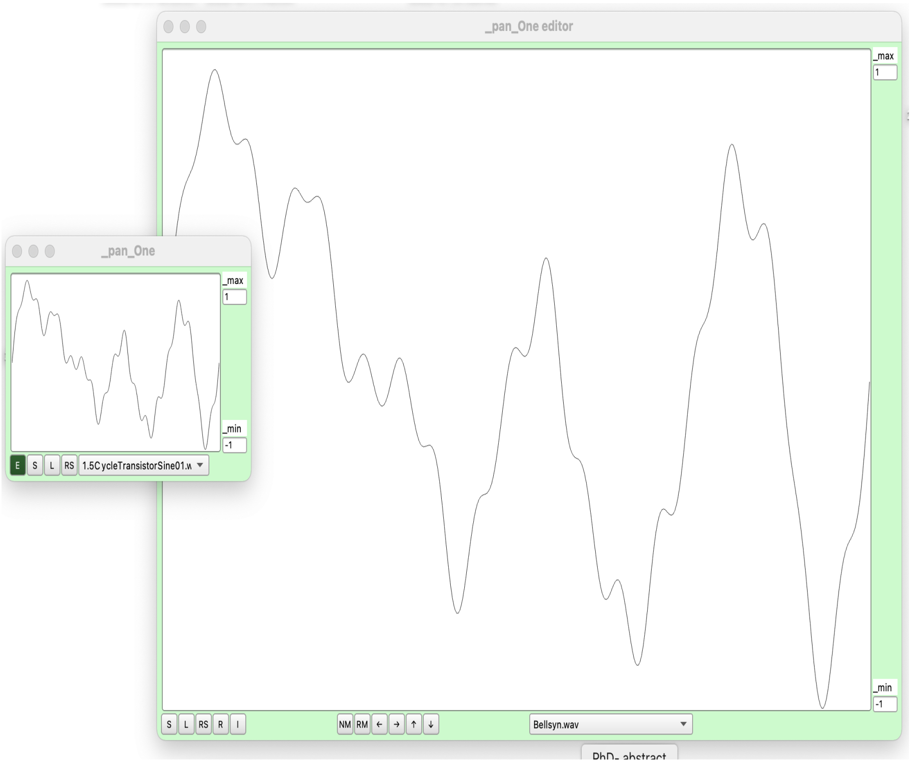


FIGURE 3.4: A shape in context as a parameter value

The transition from outside-of-time shape to its temporalisation as an in-time object is mediated through a category of logical time (3.4). This category, introduced by Agon et al., indicates "partial orders represented by lattices showing temporal logic relations between musical units" (Agon et al., 2004, p. 148). Iannis Xenakis recognised the importance of exhibiting this mediating category[24] as a diagram of relationships between sets of materials in the composition. In the context of the nuPG program, this progression can be observed as a dichotomy of a schema of synthesis logic - e.g. synthesis circuit as a structural representation of the technique - and its temporal realisation; iteration. First, the outside-of-time shape acquires a contextual function by being placed in one of the table views. While there is no visual difference between, for example, the 'pulsaret table' and 'fundamental frequency table', their role in the circuit differs significantly. Iterations over shapes as data structures project structural into temporal order. Iteration counts and yields outside-of-time structures in time. The nuPG temporalises its outside-of-time shapes through an iterative process. The iteration has a variable speed and equals the pulsar emission rate (i.e. how many pulsars are emitted per second). The emission range varies but usually is 1 to 3000, with values between 35-60 functioning at the threshold of discrete pulses and a continuous tone texture. The iteration is also a temporal clocking mechanism updating values of all synthesis parameters - the program checks at each iteration, and only then all values of the synthesis process. The iteration as an updater allows additional processing to be integrated into the circuit of pulsar emissions.

The formal aspects of the loop and processes of modulation were instrumental in the composition of works *sieveSequence_01*, *sieveSequence_02* and *sieveSequence_04* included in the accompanying portfolio. Each work employs a fixed-length loop

---

[24] see a logical time chart (Fig 3) for composition Herma (1962) for piano in (Agon et al., 2004, p.148)

enclosing a given number of pulses in a sequence. The modulation of the sequence utilises a sample-and-hold technique to create repetitive phrases. More about the technique and these works can be read later in the thesis in the context of the sieve method (1).

The loop as a formal unit within the nuPG program is intrinsically bounded with a notion of multi-temporality - it functions simultaneously at micro and meso-temporal levels of compositional operationalisation. At the lowest level, the loop takes a form of a pulsaret waveform and an envelope, which together form the pulsaret. The meso-temporal level is represented by control loops of parameters: sets of tables for formant frequency, envelope compression, spatial position and amplitude. The question of scale highlights a relationship between the fragment and the whole in the formal development of the compositional work. The loop emphasises fragmentary perspective, an instant over the total. However, the fragment and the whole should not be considered separately but as objects of an overlapping and interrelated form. A particular whole can be composed of more minor elements and consist of a fragment concerning other fragments of a higher-level whole. It can be said then that the notion of the loop institutes a mediation between an inter-elementary, macro-temporal whole and an intra-elementary micro-temporal fragment. The operationalisation of form enabled by the nuPG is articulated precisely through such "reciprocal exchange", where the focus is not on form as a fixed entity but on form as a transforming modulator or a cascade of nested loops.

Despite their variable function within different temporal levels, each of these loops represents the same basic unit, a sequence of floating point values distributed horizontally along the x-axis. The loop within the design of the new pulsar generator program manifests an object-oriented notion of polymorphism. Within the design of the nuPG program, the same shape can represent a pulsaret waveform (**??** or a parameter graph (**??**). I have discussed this above in the context of polymorphism and object-oriented programming approaches. The numerical format opens a set of possible transformations, such as resampling to a new range, reversing, scaling, morphing with other values, and erasing. These transformations function outside of time; what temporalises them is the iterative counting. Iterations over data structures project structural and potential information from the loop into temporal order. The iteration counts and yields outside-of-time structures in time[25].

The transfer between scales can also be grasped through a lens of variable synthetic modalities, i.e., multiplicative, additive, and subtractive. As discussed earlier (1.1), pulsar synthesis melds established approaches to sound synthesis into a new paradigm. From a perspective of temporal organisation, it can be identified that different operators function at different time scales (see Figure **??**) within the circuit of the nuPG program. A collection of these operators can be considered under a generalised notion of form as a modulator.

The relationship between the loop and what is being looped takes the shape of a limit. It can be said that the loop stabilises material. Gilbert Simondon writes about the boundary between form and matter as one of "an amplifier relay" (Simondon and Iliadis, 2019, p.573). The form amplifies implicit structures within the material. The formalism of a modulator described by Simondon resonates closely with a notion of "nested form" described in the context of composition by James Tenney. The proceeding section is devoted to developing a synthesis of these two formalisms.

---

[25] see also discussion on the notion of "outside-of-time" in the section on Sieves 1

3.3 Form as a Modulator: Shape, Structure and Statistics

The notion of form plays a functional role: that of "a structural germ" consisting of specific organising and guiding powers. Gilbert Simondon highlights the qualitative, active, and hierarchical asymmetry of form and matter that takes form. This does not mean a total subordination of matter to form taking process. Matter possesses potential, which can be actualised through form taking approach. Simondon incorporated the notion of modulation from the context of information theory[26] to account for the mediatory relationship between the form and the matter (Simondon and Iliadis, 2019, p.572).

The form as a modulator allows the material's structural potential to advance and be realised. Rather than the subordination of one to the other, the relationship between form and material resembles transplantation, incorporation, absorption, translation and reworking of one through the other. Matthew Fuller observes these processes in the technique of frequency modulation, where "the features of one source or kind of sound are extracted to act as control parameters for another" (Fuller, 2019, p.11). An analogous relationship can be observed within the technique of phase modulation (Anderson, Aulin, and Sundberg, 2013). The modulator—often an oscillator—modulates the phase parameter of another oscillator, the carrier. As an index of function in time, the phase then follows the form of the modulator's amplitude. The carrier possesses a phase parameter; however, in its raw state, its value is only potential, and the modulator is the source of its actualisation. The modulator, as the form, plays an informing role by exerting a force which limits the realisation of the potential of the phase parameter of the carrier. In the compositional context, the French composer Eliane Radigue focused on long-form modulations expanding over minutes and hours. Radigue's exploration of partials modulating over the fundamental, subtly undulating noises and hisses emphasises temporalities at a polar opposition to these, which are an object of the work presented as part of this project. However, the attention to work with "sound within sound" is what Radigue's work shares with microsound synthesis. In a practical context of the work with the nuPG, the modulation process can be observed at various levels. For example, formal qualities of fundamental frequency loop together with additive sets of the formant, envelope compression, spatial position, and amplitude modulate a pulsaret waveform and its envelope. This process of modulation brings forth a continuous train of pulses.

The notion of the modulator, general in scope, allows the formalism of the nuPG program in its variable manifestations - especially as a graphic interface and as codified within the programming language. The boundary between parameters of the A.6 and the **??** is a modulation. A value from the main parameter set serves as a carrier of the modulation values from table-sequencers iterated according to the time index (see Appendix A (A) for the user manual with a detailed description of parameters and their relationships). An essential generalisation within the nuPG program is the ability to formalise a new loop at a run time of the program. An example of such a loop is the sample and hold modulator, which "freezes" and repeats a selected measure of pulses (see also figure (1.11) and the corresponding discussion). The resulting loop sounds as follows:

Audio 2. nuPG loop: 'Risset-like'

---

[26] Within theoretical information discourse, the term modulation has come to represent in the most general sense any process whereby a message is translated into information-bearing signals for purposes of transmission and by which the signs reproduce the message at the receiving end (Black, 1953).

Listen to the accompanying example: 'nuPG_risset_loop.mp3'

Incorporating this particular functionality resulted in a systematic listening and programming process. A material point of departure was the album '18 Studies' (2007/2008) by Martin Neukom[27]. The album consists of thirteen short studies, each lasting 3 minutes and 19 seconds, exploring a form of the golden ratio as compositional material. The synthesis technique used by Neukom is an adaptation of frequency modulation (FM) as developed by John Chowning (1967); an application of functions based on the golden ratio generated a series of patterns which seem to rise or descend endlessly. An ability to create an approximation of the effect with the New Pulsar Generator is a testament to the pulsar technique's versatile modes of operation.

In *Track 6 (Untitled)* from the album *Altars of Science (2007)*, Marcus Schmickler applies multi-parameter modulation and scattering (procedural repetition of sets of pulsars) at a level of a singular pulsar. At 1:00, the composition transforms into a stream of fractious sonorities, imitating the dragging of rubber objects against each other. At 3:11, the chaotic oscillator as modulator simulates a sound of ripping apart, iterative staggering and repetition of short phrases. These particular sections of the track inspired a lengthy trial-and-error process. An approximation of the effect has been achieved by modulating multiple parameters with a dynamic cubic noise generator. The technique has been used extensively on tracks *tnpgr_(modulator_DistHyperbcos)* and *tnpgr_(nupggendynmodulator)* included in a portfolio of compositions accompanying the thesis. Additionally, a set of studies is included in the folder 'nuPG_gendyApproximations' of 'Sound Example Files'.

Audio 3. nuPG per-pulsar multiparameter modulator
Listen to the accompanying example: 'nuPG_perPulsar_modulator.mp3'

James Tenney proposed a complementary description of form as a nested hierarchy of correlations between three factors: shape, structure and statistic. The shape relates to a morphological and continuous "contour, the variation of some attribute of a thing in space or time". At the same time, the structure denotes a discontinuous aspect of "the disposition of parts, relations of part to part, and of the part to whole" (Tenney, 2014, p. 150). Defined as such, form implies a hierarchical level of organisation and perception[28], with the relation between "whole", "parts", and "parts of parts" as its constituent feature. Such a model is not bound to any particular timescale; it's functional to all timeframes.

An explanation of the shape and structure of a standard unit at one of these hierarchical levels involves an analysis of the statistical characteristics of the next lower-level unit. The analytical model nests at each level triad of shape, structure and statistics within the trio of shape, form and statistics at a higher level, and so on, up to the level of a whole composition. At a given hierarchical level, the "content" of each component unit is determined by its structural, morphological, and statistical

---

[27] The album can be downloaded from the website of a label domzil: `http://www.domizil.ch/releases/d25/d25.html`

[28] "The musical work - as Boris de Schloezer writes - appears to us as a hierarchy of [organic and composed] systems nested within the other, each one forming concerning the ones it embraces and matter for those that are embraced by it" (Schloezer, 1979, p. 35). Fundamental in de Schloezer's conceptualisation of musical form is the distinction between its horizontal and vertical structures, as well as the established relationship between the structural and perceptual qualities of form.

properties. In other words, the "content" is the result of "forms" at a lower level than the one we have decided to formalize[29].

Tenney approached the question of form and content along these temporal scales, where conditions at one level become content at another. The relationship between form and content becomes problematised. In this context, both can be considered as materials understood not as a hypothesis - as an inert matter manipulated arbitrarily - but as an active force imposing constraints and feeding back into the creation process. The material is not pre-configured; it must be understood as information is gathered. During the manipulation process, it is also made up, becoming isomorphic to its understanding. In this approach to musical form, practical intervention is emphasised rather than pure speculation or imposing abstract ideas on passive, malleable material. Lau, 2017 searches for the underpinnings of such problematisation in the philosophical reading of set-theoretical and categorical concepts by Alain Badiou (Badiou, 2014; Badiou, 2019). The context of set-theoretical concepts is audible in the series of works composed with the use of sieves (see section (1) for more). Let's look at Tenney's model from a set-theoretical perspective. The elements of a level could be said to be composed of the powerset of parts at its next lower domain - "*urelements* become the (quantitative-multiple) forms that coalesce into (qualitative-unitary) matter at a higher scale" (Lau, 2017, p.134). It is a protocol of pendular transition from many to one. The audible's nuances, qualities, and thresholds are determined by an operative synthetic logic at a particular temporal level. Each level has different qualities corresponding to micro, meso and macro timescales. The rates at the micro level result from fusions and contractions of vibrations into single qualitative states - timbres, tone colours. At the level of meso timescale, qualities in operation relate to sequences, patterns, phrases or sections, and at the level of macro scale to the architecture of the whole work[31]. This kind of nested architecture is evident in the nuPG program's design. The microscopic level represented by a pulsaret waveform can further decompose into various elementary objects. If we consider the waveform as a periodic function, it can be defined as a set of FFT magnitudes; as a function in time, it can be conceived as a set of 2048 data points distributed on an x-y axis or a list of such values (B).

The nuPG program's underlying design paradigm favours an iterative process between two strands of conceptualisation: the inductive, which binds the elements into the whole, and the deductive, which breaks the whole down into smaller parts. Therefore, the nuPG may be described as a system of "transparent stratification", leaving all levels of temporal organisation open to pendular differentiation and reintegration of sound materials. It further complicates the duality between form and material by allowing the same object to be conceived as either material or form (substance or container), depending on the level of investigation.

Used by Tenney, the term statistical is worth consideration in this context. As noted by (Iverson, 2018), the term was incorporated into the discourse on music

---

[29] This view can be traced back to Boris de Schloezer and a proto-structuralist, Gestalt psychology influenced approach to musical form analysis proposed in his *Introduction à J.-S Bach. Essai d'esthétique musicale*[30]. Schloezer writes, "The musical work appears to us as a hierarchy of [organic and composed] systems nested one within the other, each one forming concerning the ones it embraces and matter for those that are embraced by it" (Schloezer, 1979, p. 35). Schloezer expounded a theory of form that considers the whole more than the sum of its parts. The notion of the system employed by Schloezer in conceptualising form is close to what can be designated by the term 'structure'

[31] Accoridng to Snyder, 2000 each of these temporal levels can be further correlated with different aspects of auditory memory. The micro timescale is a region of the echoic and early processing memory operation. The meso timescale activates short-term memory. The makro timescale involves long-term memory.

composition in the mid-1950s by several composers in the European avant-garde[32]. The term "statistical form" according to Iverson "was the output of a series of studio-based translations, in which the composers learned the core concepts of information theory, revised them in the studio, and emerged with *statistical* understanding of their electronic and acoustic music" (Iverson, 2018, p. 105). Karlheinz Stockhausen had seen the origins of statistical form in the work of Claude Debussy. In *From Webern to Debussy: Remarks on Statistical Form*, Stockhausen focused on descending and ascending curves in pitch, loudness, and speed in Debussy's orchestral ballet score *Jeux* (Games). According to Stockhausen, the fluctuations of these shapes are key to a new formal approach replacing traditional motives, harmonies, etc. Interesting in Stockhausen's account is the notion of the textural continuum: "Between very large densities and pointillistic tone, dispersions lie continually all variations of density. I am thinking of a row of graded densities, for vertical as well as for horizontal density" (Stockhausen, 1963b, p. 78)[33]. Jonathan Cott has asked Stockhausen: 'What exactly is this statistical, aleatoric structure?'. Stockhausen's answer outlines the basic premises of the statistical approach to form: 'It's a random distribution of elements within given limits. ... If there's a tendency, then it's a directional statistical one – going upward or downward, becoming thinner, thicker, brighter, or darker. [...] A mass has a certain density, it has certain tendencies, it has shape' (Cott and Stockhausen, 1974, p.73). All of these terms - *densities*, *dispersions*, *horizonatl* and *vertical* distributions, are all drawn implicitly from the field of descriptive statistics, which in its fundamental guise, break down into measures of central tendency and measures of variability (spread). There are several measures of central tendencies, such as mean, median, and mode. In contrast, measures of variability are standard deviation, variance, minimum and maximum variables, as well as kurtosis and skewness (Bernstein and Bernstein, 1999). Recently, the statistical approach in the form of time-frequency statistics became a useful tool in analysing sounds composed of irregular dynamics and noise (McDermott, Oxenham, and Simoncelli, 2009). A set of algorithms developed by Josh McDermott calculate statistics (mean, variance and correlations between bands) of the audio signal, which then can be used as a filter on other audio signals. An adaptation of the time-frequency statistics by Axel Röbel has been used extensively in *Resynthese FAVN* (2017) by Florian Hecker.

Herbert Eimert wrote of statistical structure inherent to electronic music and replaced traditionally conceived musical form with a notion of densities (Eimert, Stockhausen, and Helms, 1956, p. 7). As Sean Williams shows, the electronic music studio equipment played a crucial role in mediating the emergence of the statistical form (Williams, 2015; Williams, 2016a; Williams, 2016a). For example, in composing *Gesang de Junglinge*, Stockhausen and Koenig improvised pitch curves in real-time using knobs on generators and filters, creating effects such as glissandi or varying speed impulses.

Gérard Grisey has incorporated statistical notions to conceptualise musical durations and their successions. In "Tempus ex Machina: A composer's reflections on musical time", Grisey writes of "a veritable white noise of durations" and statistics as breaking off from periodicity and approaching a-periodicity. The author discusses

---

[32] Iverson's book 'Electronic Inspirations. Technologies of the Cold War Musical Avant-Garde' explores lines of transmission that connect Claude Shannon's information theory and compositional practice of Euro-American composers of the mid-20th century.

[33] Stockhausen's *Gruppen* (1955-1957) for three orchestras distributed spatially, explores the notion of textural densities and possibilities of passing timbres from orchestra to orchestra in different dynamic shapes, speeds and directions. See: `http://stockhausenspace.blogspot.com/2014/12/opus-6-gruppen.html`

the possibility to organise durations in a spectrum-like manner here; densely packed durations "do away with periodicity, and thus with the 'harmonic' effect of the whole formant-spectrum", what is approached can be called 'noise' (Grisey, 1987, pp. 256-257). For Grisey, the notion of statistical functions as a polarity to the periodic distribution with a gradient continuous-dynamic and discontinuous-dynamic between them. The nuPG operationalises this range seamlessly, oscillating between periodic (tonal) and aperiodic (noise).

As James Tenney introduced, a model of nested form can be seen as a process of cascading modulations functioning simultaneously from the bottom up and from the top down. The chapter's final section summarises the discussion above through the notion of a multiscale approach to composition. The nuPG program emerges as an instrumentalisation of a multiscale approach to composition.

3.4 Mobilising the Bottom-up and Top-down duality: the Multiscale Approach

Articulating coherent structural relationships across and between macro-, meso- and micro-temporal levels is an ultimate test for composition and consists of a perennial issue in algorithmic and computer music. As Jean-Claude Risett observed: "The composer has to take a multiplicity of scales into account and provide connections between non-congruent temporal dimensions"(Risset, 2005, p. 289). This attention to the multiple scales and their relationship was also a concern of Iannis Xenakis. He emphasised the importance of thinking about music "not only from the detail to the generality but conversely from architectural to the details" (Zaplitny and Xenakis, 1975, p.1). Within the integrative scope, the formal concern stretches across all available timescales and calls for a new conceptualisation of musical form. Signalised at the beginning of the chapter notion of "composition of sound" indicates the persistence of formal concerns across all elements of the compositional process.

Curtis Roads proposes a general schema for a formal composition plan: Top-down, Bottom-up and Multiscale. This schema can serve as a valuable lens to examine formal aspects brought up through work with PS and nuPG.

In broader terms, the top-down approach to composition look from a perspective of the whole, the macro scale of the composition. It starts by definition of the macro scale templates - duration of sections, general shape and statistical distribution of parameters over time. This first step can be compared to sketching a silhouette without paying too much attention to details and the content. In a limited sense, the top-down approach can be likened to a historical notion of form conceptualised by Bond as a "normative container", an empty frame which can be filled with content. Of main interest for this conceptualisation is the study of form in a restricted sense of a fixed or standard scheme of relationships (e.g. “sonata form”, "rondo", "waltz")[34]. From the turn of the Twentieth Century onwards, composers have called into question such a limited notion of form. Herbert Eimert, 1959 saw in Claude Debussy's

---

[34] An interesting discussion on these traditional forms and their epistemic status has been proposed by Boris de Schloezer. In *Introduction à J.-S Bach. Essai d'esthétique musicale* Schloezer writes, "To call forms those formal schemes used by the artists is an abusive use of language. Sonata, rondo, fugue are not musical forms which are by definition individual, but rather different types of modes of action of unity upon multiplicity"(Schloezer, 1979, p. 118). The compositional approach in the realm of such tradition is referred as "conformational" (Bonds, 1991, p. 127). Burkholder, Grout, and Palisca, 2019 points out that such generic forms originated from religious ceremonies (pagan rites, sung prayers, carols, chants, hymns, masses, madrigals, motets) and social functions of the aristocracy and the military (dance fashions, fanfares, military exercises, gala symphony and opera concerts)

Jeux a replacement of traditional form by a fusion of ornamental figures and timbres that we hear as "rhythmicised time". This echoes the discussed above theme of the rhythm as a form builder[35]. The practice of top-down planning remains common in today's music. In the context of contemporary music and research, a systematic conception of data sonification can be thought of as a type of top-down planning (see discussion in (1)).

As noted by Curtis Roads, strict top-down planning has its limitations: "An issue with strict top-down planning is its overemphasis on high-level concept and structure and its underemphasis on sound material" (Roads, 2015, p.293). In the context of computer music with a myriad of heterogeneous sound materials, these need to be chosen carefully to conform to the beforehand established form.

The top-down approach in composition with the PS and nuPG can be seen in the definition of longer-duration development paths for crucial parameters such as fundamental frequency, grain (formant) frequency, amplitude and spatial position. These paths can describe a general shape within which other aspects of synthesis definition, such as pulsaret waveform and envelope, are specified. In the exact context of the work with the nuPG program, this can be seen as a compositional process which begins with devising data for tables—the process of data preparation, reformatting and mapping functions from the top down.

The bottom-up approach is a counterpart to the top-down one, it starts from the sound material, and the form of the whole work is its result. Curtis Roads identifies two characteristic material formation processes in action within this approach. The first comprises attraction processes such as symmetries, aggregations, synchronisations, and resolutions. The second is repulsion processes such as dissonances, scatterings, and desynchronisation. Generative algorithms are a bottom-up approach, among which Xenakis' GENDY system for stochastic sound synthesis is a good example. Xenakis describes this algorithm in the chapter *More Thorough Stochastic Music* of *Formalized Music* (Xenakis, 1992, pp. 295-322) and applies it in composition GENDY3 (1991)[36]. Agostino Di Scipio calls the bottom-up strategy "a microstructural time-modelling of sound". The assumption of this strategy is the emergent nature of higher-level forms (sound objects, phrases, sections, compounded sound structures) which can be "shaped by displacing some minimal sonic units in time" (Di Scipio, 1995, p.39). In this context, Di Scipio introduces a distinction between control of the parameters of synthesis and compositional control structure:

> Accordingly, the control structure is more one of process parameters (parameters evaluated by a rule-based, microlevel strategy) than of synthesis parameters: the model of material is no more a procedural description of an acoustic model but embodies a microlevel compositional strategy (Di Scipio, 1995, p.39)

The microstructural approach is also indeterminate. As observed by Di Scipio, the micro-level control is also indeterminate "in the sense that the sounding result is not necessarily a predictable, linear function of the control structure" (Di Scipio, 1995, p.40). Because the approach does not simulate "a pre-existing, known acoustic

---

[35] The interest in Debussy's *Jeux* among so-called Darmstadt School (Eimert, Stockhausen and Boulez) is a topic beyond the scope of this work. A doctoral thesis by Ding Hong discusses some problems with a misinterpretation and context analysis of the work. See (Hong, 2014). Also, see (Pasler, 1982) and (McGinness, 1998) for an analysis of *Jeux* as a music composed for ballet

[36] A thorough discussion of GENDY algorithm is beyond the scope of this thesis. For an in-depth analysis see (Luque, 2006) and (Hoffmann, 2009)

phenomenon, the composer cannot operate in a straightforward, goal-driven manner" (Di Scipio, 1995, p.40). In this way, the micro-level approach fits nicely within the introduced notion of "thingness" of sound - "thing" as something requiring active and engaged manipulation to reveal its function.

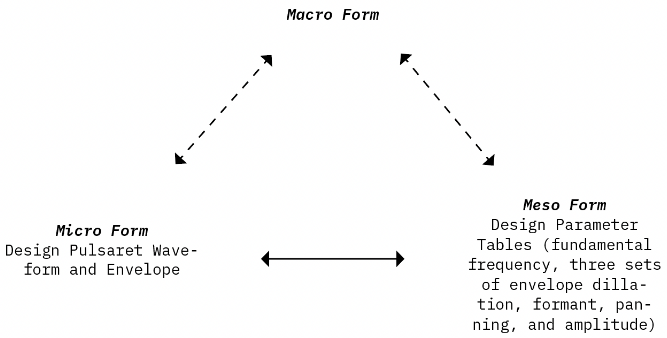


FIGURE 3.5: A schema of the multiscale approach of the New Pulsar Generator program. Notice different arrows connecting Micro and Meso Levels with Macro Levels. This indicates the non-existence of the macro-level organising method within the program's core.

Transpiring from the compositional practice and the design of the nuPG program, transitions between temporal scales can take the form of modulation. As a process, modulation operates across and in between temporal levels. As discussed previously (1.2), modulation, rather than being associated with a particular temporal scale, functions outside-of-time as an abstract and formal procedure. The design of the nuPG program, at least from the perspective of the graphic interface, favours the micro and meso-temporal levels of organisation (3.5). At the micro-temporal level, the primitive object of the pulsaret is composed of a waveform and an envelope (the relationship between them is a multiplication). An iterative process of pulsar emission in time produces the train of pulsars controllable through a set of parameters. From this perspective, the process of composition can be characterised as a bottom-up approach. However, nothing in the program's design dictates the starting point of the compositional process. All objects are accessible through the same interface disregarding their function in the circuit or temporal level of operation. Nothing stops the composer from starting with the design of parameter tables and then moving to the specification of pulsaret and envelope data. Unlike top-down and bottom-up strategies, the program is flexible and avoids a one-way strategy. The program's design, including the text as an interface, considers the entire network of complicated relationships among timescales. As observed by Curtis Roads: "Multiscale planning encourages an interplay between inductive and deductive thinking, that is, from the specific to the general, and from the general to the specific" (Roads, 2015, p.298). The practice central to this artistic research project is characterised by precisely that kind of flexible oscillation between local and global levels of composition. Figure (3.6(a)) displays a schematic view of global temporal levels available to the composer. The process of composition can begin at any level. At each level, the process can be described through a unified set of steps (3.6(b)):

1. the output of the program is listened to, viewed and analysed;
2. result can be stored as an audio file or the data as a preset;
3. it can also be transformed through additional processes;
4. if the result is not satisfactory, it can be discarded, and new settings fed in;

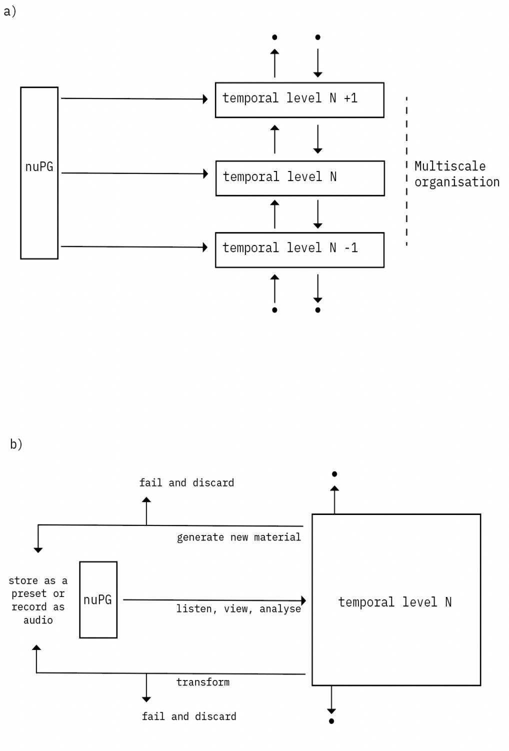


FIGURE 3.6: A global (a) and local (b) view of the multiscale assembly within the New Pulsar Generator program. The global view (a) distinguishes between temporal levels (... N-1, N, N+1 ...) as levels of the organisation. The New Pulsar Generator program approaches these shifts between scales as a generalised form of modulation. The local (b) view zooms into one of these levels and describes a generative strategy. Within a chosen temporal level, the output is listened to, viewed and analysed. It can be stored as a preset or discarded. The model of assembly adapted from Curtis Roads as described in (Roads, 2015, p. 301)

A set of compositional studies presented in the following part mobilises concepts of the multiscale assembly, modulation, outside and in-time, statistics, and nested form.

Part II

Compositional Applications

The focus of this part is to present a set of productive engagements with the technique of pulsar synthesis and its complex ecologies described in the previous chapter. This part is organised around a group of compositional methods. Each process takes as a material point of departure a singular object of the nuPG program. These objects—systematically described in a preceding chapter—consist of a closed set: envelopes, waveforms, pulsars, tables of data, iterations and loops with a group of descriptors providing information about their composition and function within the system. The purpose of the chapter is to construct a new synthetic perspective on compositional work, which puts the digital instrument with its objects and processes at its centre.

The first two Chapters build upon notions of two formal and material polarities described at the end of the preceding chapter (3). A practical engagement with the topic of bottom and top-down strategies takes the form of two converging compositional methodologies: sieves and wavelet analysis-re-synthesis. From a formal perspective, sieves and wavelets represent two polarities.

The method of sieve functions outside-of-time and is context-free, in the sense that it does not indicate a realm of its application[37]. An integer-type of sieve activity echoes the fundamental aspect of the digital material, i.e. a BIT. Integrating sieves in the context of pulsar synthesis is an attempt at incorporating this elemental materiality as an operative and audible mark. The wavelet analysis and re-synthesis, on the other hand, are sensitive to the input material and its formal structure. In the domain of audio analysis, the technique works out variable features of the input signal by applying a time-frequency grid. The model of a grid parametrising the micro-temporal level of composition will be the focus of the compositional methods presented in this chapter. These opposing perspectives dictated the choice of the two particular techniques. The purpose of the compositional study was to understand two opposite compositional prospects—confront them, multiply them, merge them— to explore and expand the scope of the audible potential of pulsar synthesis.

The exploration of compositional approaches in the conjunction of sieves and wavelets builds upon and extends the pulsar sieve implementation developed by me as a contribution to the work FAVN (2016) by Florian Hecker. From a global perspective, the FAVN was composed of three movements. It used material generated with an original implementation of sieves for the nuPG program (Movement 2). The work premiered at the Alte Oper Frankfurt, Frankfurt am Main, on October 5th, 2016, and was presented again at 'the Geometry of Now Festival' in Moscow. Recently the work was presented at REDCAT Roy and Edna Disney/CALARTS Theatre in Los Angeles. The sieve implementation for FAVN was adapted from Christopher Ariza's AthenaCL programming package (Ariza, 2001). It involved the generation of looped sequences of various complexity, which were then fed into parameters of the nuPG program. The critical function of the adaptation was a script for an automatic interpolation between presets of the nuPG program. All presets consisted of a large set of sieve-based number sequences loaded into all sequence-data-type objects of the program. The script recalled presets according to an additional sieve-based formula. The whole process was automated and required only a set of initial conditions, which were: data presets for the playback sequence, a list of intervals (duration of how long one data preset is being played), a type of instrument (a choice of an implementation model of pulsar synthesis; there were three to choose from AdC - Alberto de Campo, TK - Tomi Keränen and CJ - Chris Jeffs) and dry/wet parameter for other processes to be applied to a pulsar stream in real-time. In the final version of the

---

[37] A sieve-based process has been applied within the field of spatial organisation, architecture etc.

script, the program recorded the audio output iteratively, generating a collection of audio files.

A primary compositional material in Movement 1 of FAVN consists of 50 iterations of 47 blocks of the sound of duration equal to 21 seconds, distributed spatially over three audio channels. The playback of re-synthesised iterations is ordered by a gradient of noisiness (from noisiest to tonal). As such, the compositional-formal structure of the movement corresponds to the iterative-machine process of re-synthesis. Making audible the computational processes underlying sound synthesis has also been a key motif of work presented within this part of the thesis

The first chapter focuses on the method of sieves. A brief definition and historical background is followed by a set of applications of the method within the nuPG program. The application of sieves within the nuPG program is an original compositional approach developed within this research project. A set of introduced methods is grouped under two categories: sieve as a subtractor (vertical and horizontal) and sieve as a time parameter.

Chapter two introduces the method of wavelet analysis and re-synthesis. A practical implementation of the technique within the nuPG program is preceded by a brief historical description of the technique. The set of original methods developed as part of this research includes the application of wavelet transform at various temporal levels delineated through objects of the nuPG program, i.e., pulsaret, parametrical loop and train duration.

The last Chapter focuses on the work 'Speculative Sonification' (2021) developed as a commission for the CTM Festival and Deutschlandradio Kultur. The Chapter opens with a brief introduction to data sonification and describes a model-based sonification approach that was used in the work.

Each chapter concludes with a brief analysis of selected compositions included in the portfolio of works. The goal is to locate the correspondence between the formal structure of these compositions and the methods underlying their emergence.

Chapter 1

Sieve (Decantations)

Developed by composer Iannis Xenakis for the construction of integer-sequence generators, the sieve (in French *crible*) is the formal tool used to generate numerical patterns to represent various parameters of sound - or well-ordered sound structures - such as pitch series, time-points, loudness, density, degrees of order, or local timbres. In short, the sieve is a selection from among the points available along the axis of some dimensions. The process of sieving can be likened to the operation of decantation. Decantation is a mixture separation process; it removes a liquid layer free of a precipitate or the solids deposited from a solution. Decantation serves two purposes; it can produce a decant (liquid free of particulates) or recover precipitate. The term 'sieve' is used in mathematics to describe abstraction. By using a modulo operation, this set-theoretical filtering-out generates a series of integers that exhibit intervallic repetition. It is a common mathematical practice to select numbers by removing elements from a set(Hawkins, 1958).

Within the design of the nuPG program, sieving operates as both an outside-of-time - independent of temporal and audible manifestation - formal procedure and an in-time parametrical procedure reshaping the sound output as it unfolds. From a control point of view, encapsulated in the graphic user interface and accessible through the text editor, sieving procedures generate geometrically and aesthetically pleasing visual structures. Their sonic deployment tests the limits of perception for the formal symmetry of these structures. Christopher Ariza, the author of a versatile Python-based sieve implementation[1], approached the compositional process with sieves from a perspective of data sonification. A short introduction to sonification is provided in the Chapter **??**.

The structure of a sieve - in a raw format, a one-dimensional list of integers - can be interpreted as lines with points on them. Such a basic view was central to Iannis Xenakis's interest in sieves[2]. He said: "the image of a line with points on it, which is close to the musician and the tradition of music, is very useful" (Xenakis, 1996, p. 147). The beauty of sieves lies in their capability to generate structures of varying complexity, often periodic and displaying a variety of symmetries. Questions of

---

[1] Sieves form only a subset of several compositional tools developed by Ariza as part of his *AthenaCl* development (Ariza, 2001). Using Python to collect, interpolate, and map sieve-derived values, Ariza employed sieves for additive and subtractive synthesis, amplitude and frequency modulation, and waveform segment synthesis. The implementation by Ariza served as an essential reference point in the development and employment of sieves within the nuPG program

[2] Curtis Roads, 2015 observes a similarity between the Xenakian sieve and the modular - a visual and spatial grid system for architecture - as introduced by Le Corbusier. First published in 1948, *Le Modulor* provided a renewed scale of proportions for architecture based on the human body. The modulor was inspired by the notion of musical scales. As Le Corbusier, 2004 observed: "The modulor is a scale. Musicians have a scale; they make music that may be trite or beautiful." Xenakis worked, first as an engineer and later as an architect, under the management of Le Corbusier from 1950 to 1959. Xenakis used the modulor spatial scale to design the windows of their collaborative masterpiece, Le Couvent de la Tourette, which began in 1953

symmetry and periodicity resonate with compositional practice, too; Xenakis has observed the multileveled importance of these notions:

> In music, the question of symmetries or periodicities plays a fundamental role at all levels: from a sample in sound synthesis by computer to the architecture of a piece. It is thus necessary to formulate a theory permitting the construction of symmetries which are as complex as one might want and, inversely, to retrieve from a given series of events or objects in space or time the symmetries that constitute the series. We shall call these series "sieves." (Xenakis, 1992)

There is an exciting tension between abstract properties like mathematical symmetry or logical consistency versus perceived properties experienced through listening. As pointed out by Curtis Roads (2015): "The point is that abstract symmetry (codified visually on paper or in software) is not always perceived as such in the medium of sound" (Roads, 2015, p. 286). Composer Tom Johnson explored the relationship between visual and musical symmetries in a work *Symmetries* for piano and four hands (Johnson, 2006). The material point of departure for the work was a series of 49 drawings made in 1980 by Johnson. Musical symbols were used to compose the drawings - notes of various durations, ties, slurs, beams, trills, tremolos, etc. These drawings are symmetrical: some revolve around a central vertical axis, and others rotate or move horizontally. The constellations consist of diamonds, pyramids, columns, curves, and curves. There was no "musical" intention behind any of them. As sonic representations of these non-musical creations, these compositions are resolutely literal.

The compositional practice of the nuPG explores the tension between formal and perceived properties of sieves in their application at various levels of the program's operation. The following section opens with a brief historical introduction to sieves as both a proper tool and a compositional method. This introduction focuses mainly on the work of Iannis Xenakis. However, some alternative developments are also mentioned. The core development of this section consists of a formal description of sieves and their application within the nuPG program. Methods of sieves as a form of pulsar masking, table generator and macro-form segmenting are presented with an accompanying code and sound examples.

1.1 Xenakian Models of a Sieve

A study of Xenakis' theory of sieves can be seen in relation to his interest in number sequences and the use of logic operators with screens in his compositions[3] and groups, and his desire to develop "outside-time" musical structures. The first published explanation of sieves and their use in compositional practice was introduced in Xenakis's article "Vers un metamusique," *Le Nef 29* (1967) - which became chapter 7 of *Formalized Music*. Suitable employment of sieves, however, predates the publication - see *Herma* (1962) for piano. A second article on sieves was published in 1990 in

---

[3] Chapter 2, "Markovian Stochastic Music," of *Formalised Music*, introduces the concept of screens (see Xenakis, 1992, p.51). There are two dimensions to a screen; the frequency and intensity of this plane can be filled with events or grains, as Xenakis calls them. Xenakis employs the logic operators union, intersection, and complementation to "envisage in all its generality the manner of combining and juxtaposing screens" (Xenakis, 1992, p.57). *Analogique A* (1958) and *Analogique B* (1959) are offered by Xenakis as demonstrations of this technique (Xenakis, 1992, pp98-108). The sieve might be seen as a one-dimensional screen

Perspectives on New Music 28/1 (1990): 57-78. The article reintroduces the theoretical undertaking of sieves and includes a computer program - written in C programming language - to generate and analyse sieves. The paper and the program become chapters 11 and 12 respectively of Formalized Music [4].

An analysis of the development of the theory of sieves by (Ariza, 2005) shows that Xenakis had distinguished between two models of sieves. The first model - the complex sieve - has been laid out by Xenakis in (Xenakis, 1965; Xenakis, 1968; Xenakis, 1988). In his last treatment of the topic and its accompanying software implementation, he presents the second model, the simple sieve (Xenakis and Rahn, 1990). The second model does not incorporate aspects of the original. The main difference between these two models is the type of allowed logic operators and their levels of residual class nesting[5]

Xenakis used sieves to divide a span into multiple subdivisions by applying different moduli. Then, he manipulated each subdivision by employing logical operators like intersection and union. Xenakis' sieve formula is powerful because it is independent of time. As a result of this procedure, the material is generated outside of time, which means we do not yet know how it will unfold over time. A part of a musical work that can be formalised independently of time is called outside-of-time. Aspects dependent on time flow belong to the in-time domain. In its precompositional, theoretical state, a 12-tone row is outside-of-time, while a particular realisation of this series in a score is within it. Xenakis used sieves in many of his instrumental compositions, the best documented being *Herma* for piano solo [6]. (Squibbs, 1996) provides an overview of complex sieve formulation in Xenakis' *A.r. (Hommage à Ravel)* for piano solo. There is no documentation of sieves being used in sound synthesis. However, Xenakis points to their possible application in the conclusion of an article *Sieves*: "it is quite conceivable to apply this theory (sieves) to the synthesis of sounds by computer, imagining the amplitude and the time of the sound signal ruled by sieves" (Xenakis and Rahn, 1990, p.67). The implementation of sieves within the nuPG program stems from this unrealised proposal. The following sections focus on practical applications of sieves in sound synthesis. The examples are presented as represented in the syntax of SuperCollider language rather than using formal set-theoretical notation[7]. For reading this section, the accompanying script file 'nuPG_sieves.scd' should be open simultaneously.

---

[4] The version of the program in *Formalized Music contains numerous typographical errors which prevented it from functioning correctly. Squibbs, 1996 includes a corrected version as Appendix I of his dissertation*

[5] At this point, it is worth mentioning a set of alternative developments of sieve method for composition: Sever Tipei implementation of a sieve model in FORTRAN for use with his computer-assisted composition program MP1 (Tipei, 1975); (Malherbe, Assayag, and Castellengo, 1985) an application of sieve structures to model spectral peaks found in analysed sound files (this model was further developed by (Amiot, 1986)); Marcel Mesnage and André Riotte implemented a variety of sieve structures as part of PCR ("Partitions d'Ensembles de Classes de Résidus") system (Riotte, 1992; Mesnage and Riotte, 1993). Theoretical work on sieves by other authors include Squibbs, 2002 analysis of Xenakis' piano composition *Mists* (1980 - DOUBLE CHECK) and a use of sieves as pitch and texture generators, Squibbs, 1996 a foundational chapter on sieve theory, (Noll, Andreatta, and Agon, 2006) application of sieves as a tool for computer-aided analysis of Scriabin's Study for piano Op. 65 No. 3, Agon et al., 2004 a computer modelled analysis of *Herma* (1962) for piano and *Nomos Alpha* (1965) for solo cello, Jones, 2001 generalisation of the sieve as an analytical tool for the description of any collection as a sieve and any intervallic distance between two clusters in terms of their structure as a sieve.

[6] see a computer model and analysis of the work in Agon et al., 2004

[7] Christopher Ariza devoted a paragraph of his article "An Object-Oriented Model of the Xenakis Sieve for Algorithmic Pitch, Rhythm, and Parameter Generation." to a problem of sieve notation, highlighting a difference between formal - set-theoretical - notation of sieves and their programming language specific notation.

1.2 The New Pulsar Generator Sieve Model

The model of a sieve incorporated into the nuPG program builds upon two recent developments in updating Xenakis's method to operate within contemporary programming languages:

1. Daniel Meyer's Sieve extension to SuperCollider language (2020).
2. Christopher Ariza's new object-oriented model and Python implementation of the Xenakis sieve (Ariza, 2005; Ariza, 2004; Ariza, 2009).

1.2.1 Construction and Transformation of Sieves

A basic type of a sieve *s* is a *module* - which can be symbolised by the ordered pair (m, r) which indicates a modulus *m* and a residue class *r* within that modulus. The *r* can be thought of as a starting point of a sequence and *m* as a factor by which a sequence expands. In its basic format, a Sieve class defaults a value of *r* to 0, and provides control over the *modulus* and the limit of expansion[8]

Code 1. Basic Sieve Definition

```
//basic sieve definition
~sieve = Sieve(3, 12);
//returns
-> Sieve(points, List[ 0, 3, 6, 9, 12])
```

Such module *m* is a simple set which contains one element per period for an unspecified number of periods. A combination of sieves - through logical operator of union ("|"), intersection ("&"), difference ("-") and symmetric difference ("–") - produces composite sieves with variable period. For example:

Code 2. Union of Sieves

```
//define two sieves
~sieveOne = Sieve(3, 12);
~sieveTwo = Sieve(4, 12);
//union ("|") of above
~unionOf = ~sieveOne | ~sieveTwo;
//returns
-> Sieve(points, List[ 0, 3, 4, 6, 8, 9, 12])
```

As proposed by (Ariza, 2004, p. 41), the integers which are part of the sieve can be thought of as *points* and the remaining ones - sort of in-between points - which are not part of the sieve can be considered as *slots*. The nuPG implementation by default yields a sieve in a point format. A simple conversion method is provided to generate a slot representation:

Code 3. Sieve as Interval

```
//get slot representation, slots can be thought of as intervals between consecutive points
~unionOf.toIntervals;
//returns
-> Sieve(intervals, 0, List[ 3, 1, 2, 2, 1, 3])
```

For composite sieves, the period equals the lowest common multiple (LCM) of respective moduli. For above example - *m* LCM(3,4) = 12. Slots are symmetrically

---

[8] As seen in (Ariza, 2005) in theory sieves can extend from *r* infinitely and bidirectionally in positive and negative range of integers. The implementation presented here uses only integers starting from 0, expanding into a positive range and limited by a boundary value

distributed around a central pair [2,2]. Such a symmetrical arrangement is a special characteristic of the union of two or more sieves. An interpretation of these as time-points would result in "non-retrograde rhythm" as conceived by Oliver Messiaen (Messiaen, 1956). As explored by Daniel Meyer 2020 symmetric structures in periodic series occur as different types, depending if the period length is even or odd[9]. Visualisation of a complex sieve helps to grasp the overall shape of the distribution and inner symmetries (1.1). For the below examples, the limit value needs to be increased to observe the period of the resulting sequence. The LCM of 3, 4 and 7 = 84, then the limit should be at least double that to keep periodic repetition:

Code 4. Symmetry: Union of Three Sieves

```
//define the limit
~limit = 84 * 2;
//basic sieves
~sieveOne = Sieve(3, ~limit);
~sieveTwo = Sieve(4, ~limit);
~sieveThree = Sieve(7, ~limit);
//union of above
~unionOf = ~sieveOne | ~sieveTwo | ~sieveThree;
//plot values to see the symmetry
~plot = ~unionOf.toIntervals.list.asArray.plot;
```

The other logical operations possible on sieves yield different structures with various symmetries. For example a symmetrical difference of sieveOne, sieveTwo and sieveThree (1.2):

Code 5. Symmetry: Symmetric Difference of Three Sieves

```
//define the limit
~limit = 84 * 2;
//basic sieves
~sieveOne = Sieve(3, ~limit);
~sieveTwo = Sieve(4, ~limit);
~sieveThree = Sieve(7, ~limit);
//symmetric difference of above
~unionOf = ~sieveOne -- ~sieveTwo -- ~sieveThree;
//plot values to see the symmetry
~plot = ~unionOf.toIntervals.list.asArray.plot;
```

Xenakis introduced a set of transformations (*hyperbolae* to which sieves can be subjected. One such transformation is a transposition created by adding the transposition value to the shift of each residual class in the sieve. Transposition can also be thought of as an offset value. For example, an offset of 1 and 3 applied respectively to sieveOne and sieveTwo yields the following sequences:

---

[9] Meyer defines basic symmetries and provides supporting statements:

1. A period is called **symmetric** if it is equal to its reverse
2. A period is called **quasi-symmetric** if the continuation with the first element of the period is symmetric
3. If a sequence contains a symmetric or quasi-symmetric period, there exists a symmetric or quasi-symmetric period starting in its middle (or just right from it when odd); this can be denoted as a **coperiod**

A set of supporting statements proposed by 2020 state that:

1. If the period length is even, a symmetric period corresponds to a symmetric period, and a quasi-symmetric period corresponds to a quasi-symmetric period. Such correspondence can be explained as ...
2. If the period length is odd, a symmetric period corresponds to a quasi-symmetric period object Only a period of identical elements can be symmetric and quasi-symmetric at the same time

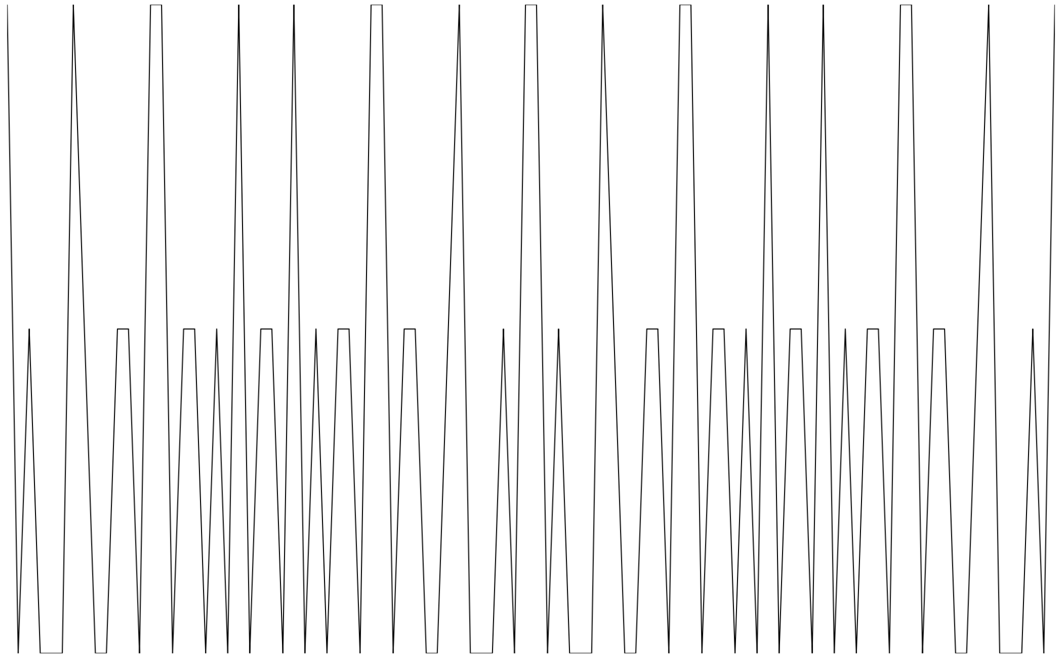


FIGURE 1.1: A plot of slots for the union of s.(3,0), s.(4,0) and s.(7,0)

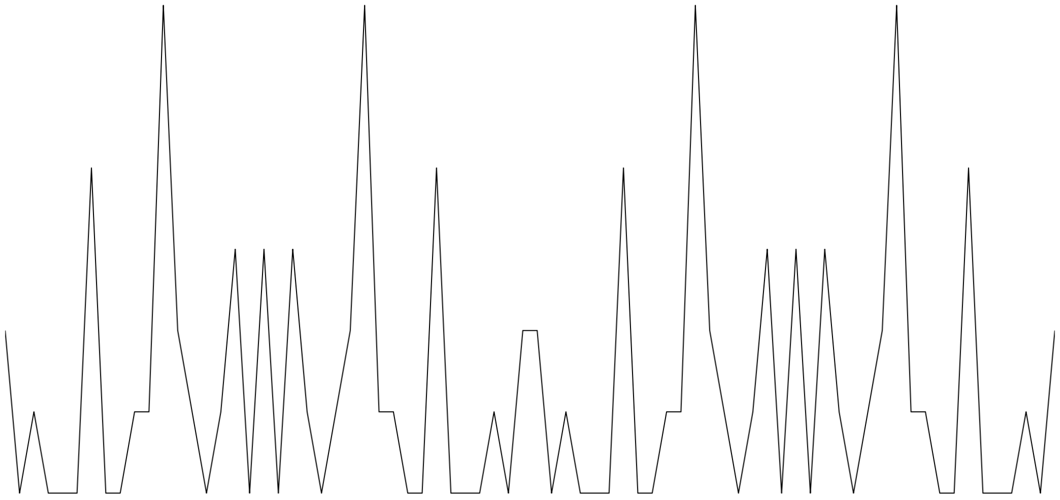


FIGURE 1.2: A plot of slots for the symmetric difference of s.(0,3), s.(0,4) and s.(0,7)

Code 6. Sieves Offset Definition

```
//set the limit
~limit = 12 * 24;
//define basic sieves with offset values
~sieveOne = Sieve.new_o(3, 1, ~limit);
~sieveTwo = Sieve.new_o(4, 3, ~limit);
```

A union which results in the following sequence:

Code 7. Sieves Offset Union

```
//a union of above
~unionOf = ~sieveOne | ~sieveTwo;
//convert to slots
unionOf.toIntervals;
//returns
-> Sieve(intervals, 1, List[ 2, 1, 3, 3, 1 ])
```

Such transformation represents a rotation of a previous sequence (i.e. [[3,1,2,2,1,(3)]]). All sieves formed through union result in symmetrical spacings of slots. The transposition allows the intervals within the spacings to transform by order rotation. If m of elementary sieves are mutually prime, transposition values will not change the sum of the period equal to the least common multiple but will only cause a shift. Things become more complicated when elementary sieves have prime factors in common: still, period sums are preserved, but symmetry types can change; asymmetric periods occur. The same set of m can cause different combinations of period types with different offsets. For example a union of Sieve(8, 48) and Sieve(12, 48) yields a symmetric set of slots (1.3), added transposition to one of the sieves disturbs such symmetry (1.4):

Code 8. Symmetry: Sieves Offset Union

```
//set the limit 8.lcm(12) = 24, 24 * 2
~limit = 24 * 2;
//define basic sieves
~sieveOne = Sieve(8, ~limit);
~sieveTwo = Sieve(12, ~limit);
//union of above
~unionOf = ~sieveOne | ~sieveTwo;
//plot to see the symmetry
~plot = ~unionOf.toIntervals.list.asArray.plot;
```

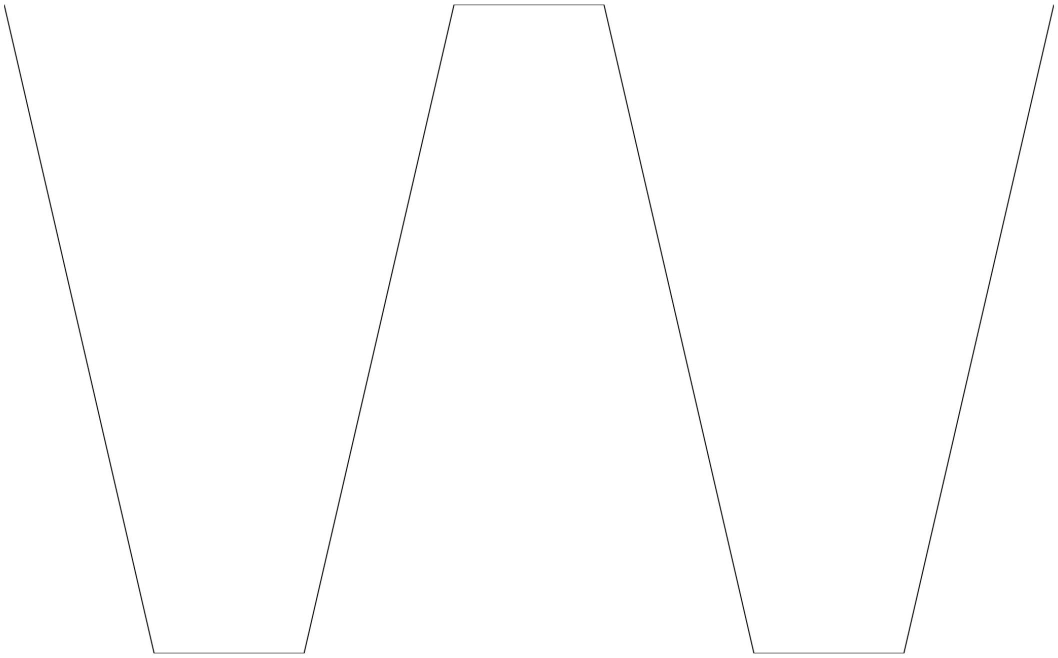


FIGURE 1.3: A plot of slots for a union of Sieve(8, 48) and Sieve(12, 48)

Code 9. Symmetry: Sieves Offset Union

```
//set limit
~limit = 24 * 2;
//define basic sieves, notice the offset value (1) for ~sieveTwo
~sieveOne = Sieve(8, ~limit);
~sieveTwo = Sieve.new_o(12, 1, ~limit);
~unionOf = ~sieveOne | ~sieveTwo;
~plot = ~unionOf.toIntervals.list.asArray.plot;
```

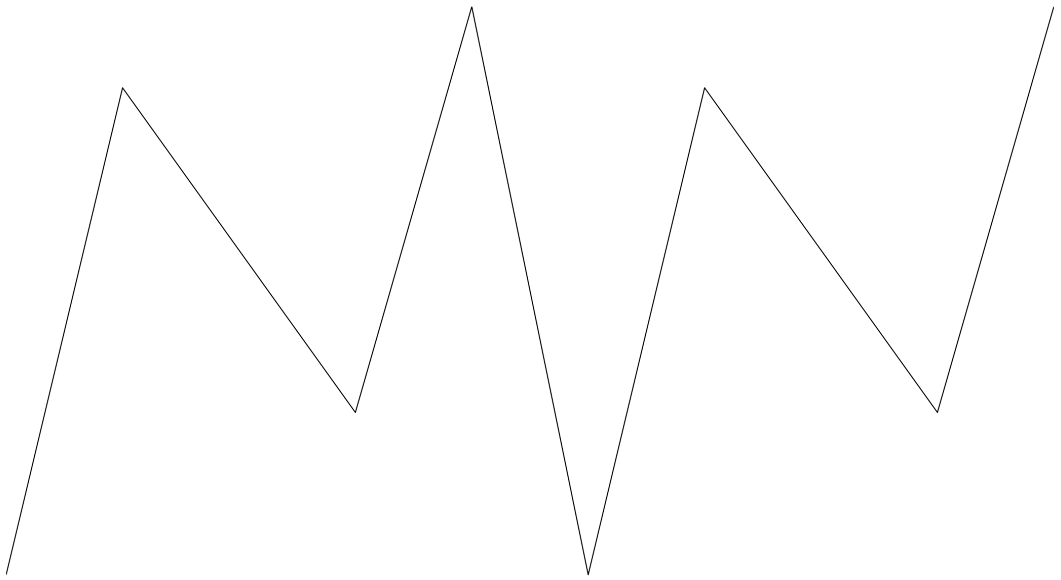


FIGURE 1.4: A plot of slots for a union of Sieve(8, 48) and transposed Sieve.new_o(12, 1, 48)

These must be formatted appropriately for sieves to apply to various parametrical scenarios within the nuPG program. Formatting, as with any translation process, preserves some features of the input material and discards others. Ariza, 2004 describes four practical representations, or formats, of a sieve segment: integer, width, unit segments and binary segments. These representations applied to a basic sieve Sieve(3,12) yield the following sequences:

Code 10. Sieves Formatting

(a) Integer segments (a default format):

```
 //a definition of two basic sieves to be reused in other examples of formatting
~limit = 84 * 2;
~sieveOne = Sieve(3, ~limit);
~sieveTwo = Sieve(4, ~limit);
//union of above
~unionOf = ~sieveOne | ~sieveTwo;
//returns
-> Sieve(points, List[ 0, 3, 4, 6, 8, 9, 12, 15, 16, 18, 20, 21, 24 ])

```

(b) Width segments measure the distance from one point to the next, counting the point itself and the intervening slots. Width segments can be equalled to slot value or an interval:

```
 //a conversion of the above to intervals
~unionOf.toIntervals
//returns
-> Sieve(intervals, 0, List[ 3, 1, 2, 2, 1, 3, 3, 1, 2, 2, 1, 3
 ])

```

(c) Unit segments map sieve sequence to the unit interval and translate segment points into real numbers between 0 and 1. Because unit segments do not treat points as integers, internal and external slots are discarded, and z, normally discrete, becomes a continuous range. A unit segment contains no information on the number of slots between points, yet accurate proportional spacing, including z-relative position, is retained:

```
 //unit segment format
 ~unionOf.list.asArray.normalise
 //returns
 [ 0.0, 0.125, 0.16666666666667, 0.25, 0.33333333333333, 0.375,
  0.5, 0.625, 0.66666666666667, 0.75, 0.83333333333333, 0.875,
1.0 ]

```

(d) Binary segments

1.2.2 Sieve as a Method of Subtraction

In a series of compositions, the sieve-based procedures were used as a type of subtraction - a procedure applied to extract selected parts of material from a bigger whole. In the context of work with the nuPG program, we can talk about two particular applications of such a method: 1) Sieve-Horizontal Subtraction, which operates in the time domain; in its basic form, the method introduces sieve-based gaps in the signal; 2) Sieve-Vertical Subtraction, which divides the signal into sieve based vertical slices (this method has its application as an amplitude selection which discard anything outside of a given range of values. The following sections provide a detailed view of these two applications of the sieving procedure.

```
(
//proof of concept, move across bands with mouse movement
//only selected bands are audible
~synthesis.trainInstances[0][2] = \filterIn -> {|in|
    var sig = in;
    var a, b, c, d;
    #a, b, c, d = BandSplitter4.ar(sig, 200, 600, 1000);
    SelectX.ar(MouseX.kr(0, 3), [a, b, c, d]);
};
)

//remove the process
~synthesis.trainInstances[0][2] = nil
```

Sieve as Time Parameter: Horizontal Subtraction

The sieve-horizontal subtraction method has been applied within the nuPG program as a pulsar masking technique[10]. This method creates sequences of pulses interspersed with silence in the range below the rhythm-tone threshold. The application

---

[10] For more information about different types of masks used in historical models of pulsar synthesis, see previous chapter 2.2

of sieve-based processes for pulsaret masking is an original development of the New Pulsar Generator program. For sieves to be used as a masking sequence, they must be properly formatted. As presented above, there are four possible formats of sieves: integer, width, unit and binary. In the context of pulsaret masking, a binary form has been used. Two models of binary formatting have been developed:

1. takes an integer representation (points) of a sieve, replaces each point with one and fills all spaces (slots) that are not occupied with zeros. E.g., Sieve = [3, 7, 11, 15] will be converted to [1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1]. This approach is based on the Python function 'discreteBinaryPad' developed by Christopher Ariza within his *athenaCL*. Let us call this method **sieve sequential binary**;
2. takes an integer representation (points) of a sieve, gets all the slots form gives sieve - this is always one value less than the input list - replaces each integer with a corresponding number of 0 and 1 in a sequence. E.g. input = [2,3,3] -> output [0,0,1,1,1,0,0,0]. This approach is an original development within the nuPG. Let us call this method **segmented sieve binary**

The technique of masking, when applied to discrete (infrasonic range) pulses creates rhythmic patterns of variable regularity; its application to continuous (audio range) produces amplitude modulation on timbre.

Figure 1.5 visualises a 16-step grid with members of four (a,b,c and d) elementary sieves highlighted in dark grey.

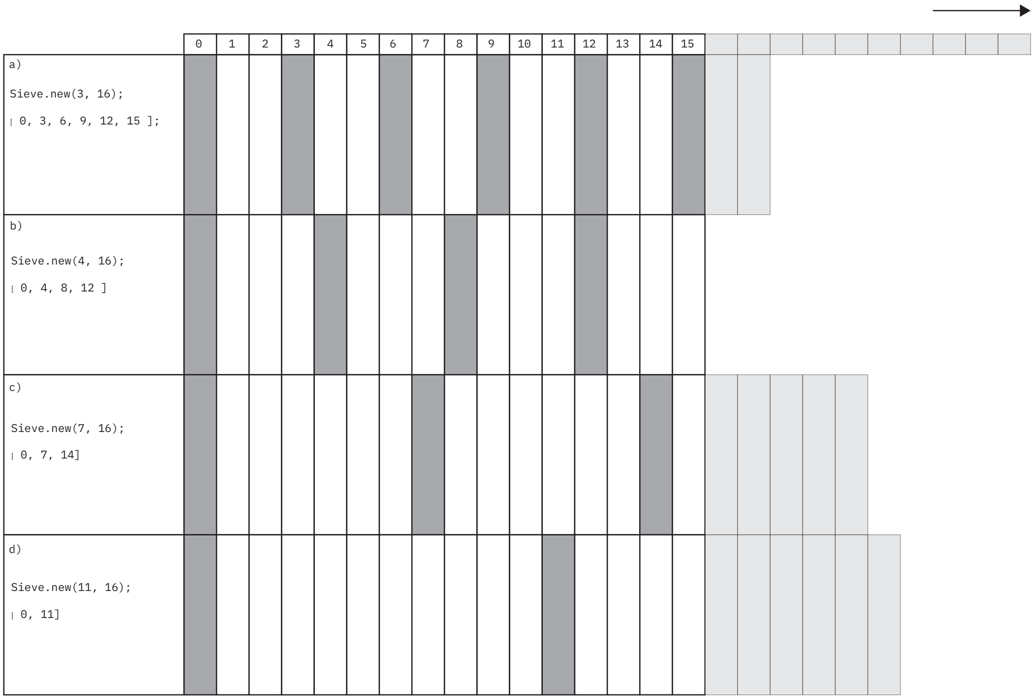


FIGURE 1.5: Examples of four sieve constructors limited to 16 steps (0-15) and only a positive range of integers. The limit is arbitrary and - as shown on the graphic - often disrupts the repetitive sequence. The light grey squares designate several units needed to complete a cycle. An arrow signalises a potential infinity of the sequence.

A starting point for sieve-based masking is a stream of continuous pulses (listen to the attached sound file: 'nuPG_continuousPulses.mp3')[11]. From sieve (1.5(a)), two masks with two different auditory renderings can be obtained:

A union of (a) and (b) results in complex non-symmetrical segments:

1. sieve segmented binary -> [0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 1],

   Audio 4. a | b: sieve segmented binary

   listen to the corresponding audio rendering:
   'nuPG_sieve_unionOfAandB_segmentedBinary.mp3' ;

2. sieve sequential binary -> [1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0],

   Audio 5. a | b: sieve sequential binary

   listen to the corresponding audio rendering:
   'nuPG_sieve_unionOfAandB_sequentialBinary.mp3' ;

A combination of all elementary sieves (a, b, c and d) in a complex logical string yields a variety of masking sequences with or without symmetry:

1. a | b | c | d ->
   - sieve segmented binary: [0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0],

     Audio 6. a | b | c | d: sieve sequential binary
     listen to the corresponding audio rendering:
     'nuPG_sieve_unionOfABCD_segmentedBinary.mp3' ;
   - sieve sequential binary: [1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0],

     Audio 7. a | b | c | d: sieve sequential binary
     listen to the corresponding audio rendering:
     'nuPG_sieve_unionOfABCD_sequentialBinary.mp3' ;
2. a–b&a | d ->
   - sieve segmented binary: [0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0],

     Audio 8. a–b&a | d: sieve segmented binary
     listen to the corresponding audio rendering:
     'nuPG_sieve_complex00_segmentedBinary.mp3' ;
   - sieve sequential binary: [1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0],

     Audio 9. a–b&a | d: sieve sequential binary
     listen to the corresponding audio rendering:
     'nuPG_sieve_complex00_sequentialBinary.mp3' ;
3. a–c | b–d ->

---

[11] The sequence was generated with following settings: trigger frequency = 16, no modulation

- sieve segmented binary: [0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1],

  Audio 10. a–c | b–d: sieve segmented binary
  listen to the corresponding audio rendering:
  'nuPG_sieve_complex01_segmentedBinary.mp3' ;
- sieve sequential binary: [1, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0],

  Audio 11. a–c | b–d: sieve sequential binary
  listen to the corresponding audio rendering:
  'nuPG_sieve_complex01_sequentialBinary.mp3' ;

Within the nuPG program, the application of sieve masking is accessible through a visual interface (1.6). The interface divides into four parts: (a) 'SIEVE GENERATORS', (b) 'LOGICAL OPERATORS' and (c) 'OUTPUT SEQUENCE'. The 'SIEVE GENERATORS' part provides a small script editor where basic sieves can be defined. In a default format, a sieve is assigned to a variable - letters of the alphabet - which are used in the next part ('LOGICAL OPERATORS') as constructors of the logical operator's chain. The last part ('OUTPUT SEQUENCE') displays a formatted sequence of 0s and 1s - designating pulsar OFF and ON. The functionality 'sieveAsTable' formats sieve-based logical operation into a table format. The sequence is automatically copied into the clipboard and can be pasted (keyboard key 'P') into any table-type object of the program. The interpolation between sieve-based values of the table is linear, creating stepped sequences of flat segments. Further sound examples exploring sieves in conjunction with frequency modulation and envelope dilation are located in the folder 'nuPg_sieves' of 'Sound Example Files'.

Sieves obtained through the procedures shown above can be transformed further through a selection of processes using a text editor as an interface:

1. rotate - elements of sieve sequence are in rotated order *n*. Positive values for *n* shift the sequence to the right, negative to the left. E.g., [0, 0, 1, 0, 1, 0, 1].rotate(3) -> [1, 0, 1, 0, 0, 1, 0];

2. pyramid - elements are reordered via one of 10 "counting" algorithms. E.g., [0, 0, 1, 0, 1, 0, 1].pyramid(1) -> [0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1];

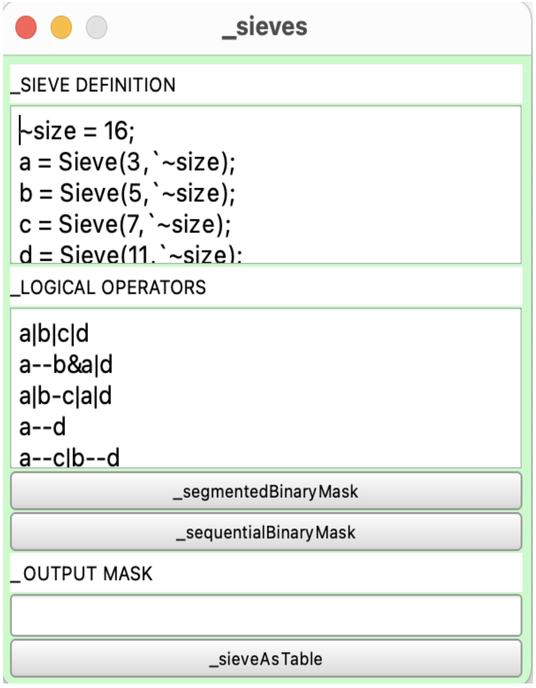
3. stutter - elements are repeated *n* times. E.g., [0, 0, 1, 0, 1, 0, 1].stutter(3) -> [0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1];

4. sputter - return a new sequence of length *maxlen* with the items partly repeated (random choice of given probability). E.g., [0, 0, 1, 0, 1, 0, 1].sputter(probability: 0.5, maxlen: 16) -> [0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1];

5. mirror - return a new sequence which is the receiver concatenated with a reversal of itself. The centre element is duplicated. E.g., [0, 0, 1, 0, 1, 0, 1].mirror2 -> [0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0];

Vertical Subtraction

An integral approach to sieve processing consists of a method of sieve-based frequency band splitting, which can be considered a type of vertical subtraction. The



```
~size = 16;
a = Sieve(3,`~size);
b = Sieve(5,`~size);
c = Sieve(7,`~size);
d = Sieve(11,`~size);
```

```
a|b|c|d
a--b&a|d
a|b-c|a|d
a--d
a--c|b--d
```

FIGURE 1.6: The sieve GUI. Four interface sections: (a) 'SIEVE GENERATORS' - allows to define of basic sieves of a given size; (b) 'LOGICAL OPERATORS' - allow construction of logical operators chains of chosen sieves. Two formats are possible: Segmented Binary and Sequential Binary; (c) 'OUTPUT SEQUENCE' - displays a formatted sequence of 0s and 1s - designating pulsar OFF and ON respectively; (d) 'SIEVE AS TABLE' - generates a table out of logical operation on sieves. The table is automatically copied into the clipboard and can be pasted (key P) into any table-type object of the program. The formulas defined by a user can be saved

method is likened to a historical technique of *selecteur d'amplitudes* used by Henri Pousseur in his composition *Scambi* (1957). The technique used by the composer functioned as a brick-wall type of filter which preserved those portions of an input signal above a given amplitude threshold and discarded any movement below the threshold. Pousseur supplied inharmonic sounds and noise into the amplitude selector; while manually controlling the amplitude threshold, he could "sculpt" discontinuous bursts of noises of variable temporal density and spectral properties. Agostino Di Scipio describes a similar technique used by Bruno Maderna in his *Syntaxis* (1957) (Di Scipio, 1994). The algorithm applied within the nuPG program uses a sequence of parallel crossover filters to split the input signal into two, eight and sixteen bands[12] that can be summed back to the original signal 1.7. The size of each band can be user-specified or generated through sieve-based formulas. These are identical to those used for pulsar masking (see above for a description of 'segmented binary' and 'sequential binary' methods). Band splitting allows for selective muting and voicing of frequency bands. An additional set of processes applied to band segments, such as modulation, spectral spatialisation or wavelet transform, opens the technique to further formal experiments.

Code 11. Sieves as an Iterator

---

[12] The number of bands is potentially unlimited as one can apply a chain of band splitting algorithms recursively

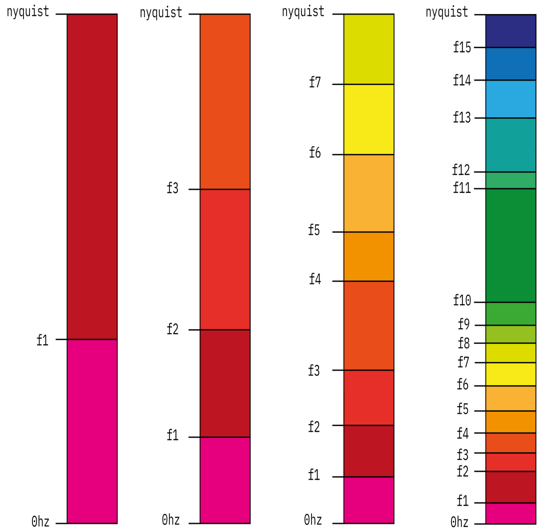


FIGURE 1.7: A visualisation of band splitting process. In a basic form, the process uses a sequence of parallel crossover filters to split the input signal into two, four or eight bands that can be summed back to the original signal. Each new band can be further transformed, e.g. muted, modulated in frequency. The process of band splitting can also be used as the first stage of a chain of processes, e.g., as segmentation for wavelet transform, which rather than being applied to the whole of the signal, becomes applied to selected bands. In this scenario, each band is processed by a separate wavelet transform with a unique set of analysis/re-synthesis parameters. I have used this method in two works: *Sieve Decantation i* and *Sieve Decantation ii*

```
//iterative windowed pitch shifting
(
//a reference to nuPG stream
~synthesis.trainInstances[0][2] = \filterIn -> { |in,
    fundamental_frequency|

  //chain of pitch shifting algorithms
  //variable window size
  //dynamic pitchRatio controlled by a set of sawtooth oscillators
  var p1 = PitchShift.ar(
    in: in,
    windowSize: 1.3,
    pitchRatio: LFSaw.ar(LFDNoise3.kr(0.1).range(0.01, 12.6)).exprange
    (0.9, 2.1),
    pitchDispersion: 0.0,
    timeDispersion: 0.0
  );
  var p2 = PitchShift.ar(
    in: in,
    windowSize: 0.7,
    pitchRatio: LFSaw.ar(0.1).exprange(1.9, 2.1),
    pitchDispersion: 0.0,
    timeDispersion: 0.0
  );
  var p3 = PitchShift.ar(
    in: in,
    windowSize:  0.5,
    pitchRatio: LFSaw.ar(LFDNoise3.kr(1).range(1.1, 2.6)).exprange(1.9,
     2.1),
```

```
    pitchDispersion: 0.0,
    timeDispersion: 0.0
  );

  //sequencing
  DXMix.ar(
    in: Dseq([0, 1, 2, 1, 1, 2, 0, 1, 0], inf),
    channelsArrayRef: ‘[p1, p2, p3],
    fadeTime: Dseq([0.4, 1.1, 0.9], inf),
    stepTime: Dseq([0.4, 1.4, 0.9], inf),
    fadeMode: 0
  );
};
)
```

In the example (11) the sieve generates durations for the iteration over pitch-shifting algorithms. The pitch shifting algorithm (PitchShift) is a time-domain granular pitch shifter. Grains have a triangular amplitude envelope and an overlap of 4:1 and use linear interpolation of the buffer. As such, the process is likened to the granularisation of live input.

1.3 Sieve as a Time Parameter

A formulation and transformation of sieves within the code editor allow their use in all nuPG program parameters. Examples of sieves' employment in conjunction with various temporal scales operationalised through pulsar synthesis are fascinating from a compositional perspective. At the micro-temporal level, the sieve functions as a generator of the table (e.g., the pulsaret waveform and envelope tables) and group offset values. The formulation of a sieve as a table generator operates through a three-step process:

1. A definition of constituent sieves, e.g., a, b, c
2. A formulation of logical operations between those sieves, e.g., a | b–c
3. A conversion of the resulting sieve sequence into a format acceptable by nuPG's table editor. The conversion consists of extraction of interval values from the sieve and remapping these into the desired range (-1 to 1) of 2048 values

A sieve-based sequence applied to the envelope table functions similarly to the spectral filter. The size and shape of the sieve can be interpreted as a resolution of the filtering over pulsaret waveform (1.8)

Curtis Roads noted that the envelope's shape significantly influences the spectrum of the pulsar synthesis output (Roads, 2004, p.146). The effect of filtering over a pulsar sinusoidal waveform (1.9) displays a vast range of transformations possible from a set number of sieves.

Audio 12. nuPG sieve over pulsaret
Listen to the accompanying example:
'nuPG_sieveOverPulsaret.mp3'

The audio (12) presents an iteration of envelope shapes (1.8) over a pulsar train. The method of sieving functions as a spectral filter, accentuating various collections of formant frequencies.

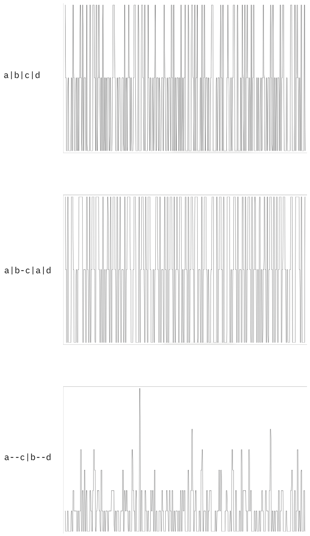


FIGURE 1.8: A sieve-based envelope tables. The Constitutive sieves were defined as follows: a = [0, 3, 6, 9, ...]; b = [0, 5, 10, 15, ...]; c = [0, 7, 14, 21, ...]; d = [0, 11, 22, 33, ...]. The size was set to 512 values. By varying logical operations on predefined sieves, one can achieve a variety of temporal structures. The application of these as an envelope accentuates and mutes selected spectral regions of the output.

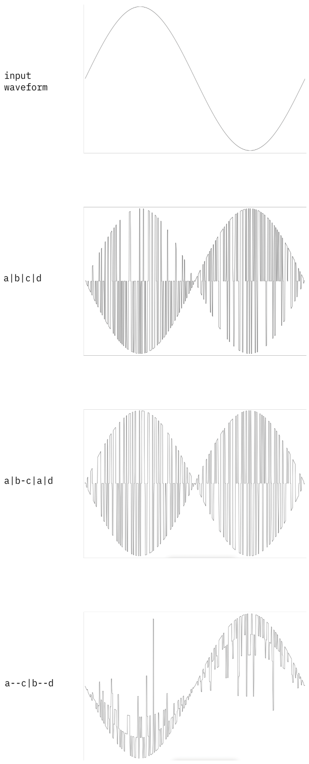


FIGURE 1.9: An effect of sieve-based envelope filtering over a pulsaret sinusoidal waveform

An original method developed within the nuPG program consists of the micro-temporal offset of constitutive groups of the formant, envelope dilation, panning and amplitude. In a default setting, each group is synchronous to a pulsar emission rate (fundamental frequency). The offset parameter allows for shifting in time a group delaying its onset and creating an asynchronous sequence of pulsars (1.10). At a discrete rate, the method functions as a delay offsetting the beginning of constituent groups. At the emission rate above 40 pulses per second, the method allows for cross-modulations and frequency beatings between formants.

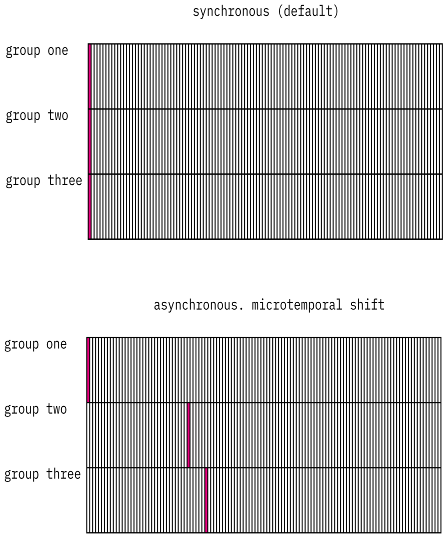


FIGURE 1.10: A micro-temporal shift between groups of the additive model of formant frequency, envelope dilation, panning and amplitude. The method allows for shifting the synchronous (based on the clock of the fundamental frequency) train into an asynchronous one. The shift value is based on milliseconds, and the maximum shift value is one second

1.4 Portfolio

The sieve-based methods have been used across many compositions included in the accompanying portfolio. Most prominently within a series of compositions from an album, 'Auditory Sieve' (Pietruszewski, 2020a). Tracks 08 ('Sieve Sequence') and 10 ('Sieve Sequence (DPOAE)') use a method of sieve for procedural masking of pulses. Both compositions incorporate frequency modulation along with a sample and hold mechanism (1.11) to generate a characteristic stepped Risset-like pattern[13].

On track 10 ('Sieve Sequence (Distortion Product Synthesis)'), in addition to sieve-based masking, the sieve method was used to generate fundamental and formant frequency values for the distortion product synthesis. The distortion appears as a tone with a frequency equal to a difference between the two partial tones (*f2* - *f1*). For example, the combination of 1800 Hz and 1500 Hz played at a high volume generates a difference tone of 300 Hz. The combination tone is not being generated externally. Still, it is clear and objectively perceived by the listener. Hermann von Helmholtz was the first to identify sum and difference tones as the result of auditory distortion in 1856. He used two prolonged tones played at high volume with an octave interval. An additional tone is sounded when two tones - a

---

[13] Pattern was created by a French composer Jean-Claude Risset, and it is considered an audio illusion. The Risset rhythm appears to accelerate or decelerate endlessly. See (Risset, 1986) for explanation of the phenomena

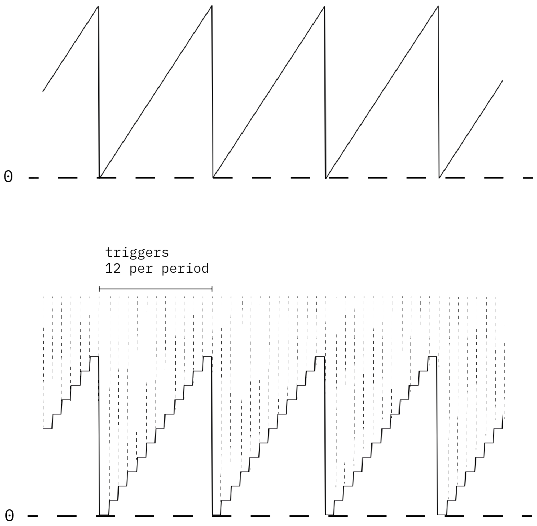


FIGURE 1.11: A visualisation of the sample and hold mechanism. A sawtooth wave function is sampled at regular intervals - 12 times per period - holding values between each sample trigger

lower and a higher - sound together. These phenomena were initially explained by high-intensity levels pushing linear mechanics into the nonlinear territory. Accordingly, the nonlinearity was believed to be located in the middle ear or the basilar membrane. (Helmholtz, 2013). According to the findings of Thomas Gold and David T. Kemp, the ear should not be considered passive but an active system with parts of its inner ear - specifically the outer hair cells of the basilar membrane - acting as a dynamic amplification system (Gold, 1948; Kemp, 1978). Currently, auditory distortion is studied in the area of otoacoustic emissions. It is defined as a cochlear amplifier, a type of "tiny loudspeaker in the ear" that provides positive feedback (Ashmore et al., 2010). In medical practice, the otoacoustic emission phenomena have been used, among other purposes, to test hearing in infants. See: `see:https://www.ncbi.nlm.nih.gov/pubmed/12764789`. In the field of sound art, Maryanne Amacher contributed significantly to research about auditory distortion. She used the Triadex Muse, a digital sequencer instrument developed by Edward Fredkin and Marvin Minsky at the Massachusetts Institute of Technology (MIT), for compositions such as "Head Rhythm 1" and "Playing Thing 2". Fast-paced interlocking patterns of short sine tones were generated at very high volumes by Amacher to create a distinct musical stream within the audible distortion (Amacher, 1999). Amacher laid the groundwork for systematic exploration and engagement with distortion product otoacoustic. Several examples of 20th-century music that explored auditory distortion as a material strategy were presented by Jonathan Kirk and Christopher Haworth (Kirk, 2010). Recent examples of creative investigation of phenomena include Christopher Haworth’s ’Correlation Number One’ (2010), Marcus Schmickler’s ’Fortuna Ribbon’ (2015), Thomas Ankersmit’s ’Otolith’ (2015), and Florian Hecker ’FAVN’ (2017). In October 2021, the composition ’Auditory Scene Re-Synthesis as Cochlear Wavepackets’ by me and Jan St Werner (Mouse on Mars, Lithops) was presented as part of the Sound of Distance festival at HKW Berlin (`https://www.hkw.de/de/programm/projekte/2021/the_sound_of_distance/start.php`). The work incorporated a re-synthesis of synthetic voice and field recording materials with imposed distortion product synthesis.

The work 'Sieve Sequence (Distortion Product Synthesis)' originated from an experimentation with the technique of quadratic difference tone (QDT) proposed by Kendall, Haworth, and Cádiz. The technique models a dynamic sinusoidal synthesis with a spectrum-based quadratic difference tone (*f2 - f1*) from an audio signal input. The technique successfully generates a difference tone from signals with dynamically varying fundamental frequency and amplitude. In the context of the application within the nuPG program, the pitch and amplitude of the QDT spectrum are made to follow the characteristics of a pulsar train. As such, the method functions between sieve and analysis-resynthesis contexts which are discussed as part of the next chapter. Due to the complexity of the nuPG's signal (e.g. defined transients, the composition of time-varying formant) the resulting sound does not exemplify a single distortion. Particularly interesting from the perceptual point is the occurrence of a chirp-like sound before the distortion. If we consider the distortion product as a sound in reverse (a response), then this chirp can be considered the stimulus. As such the work does not only play with distortion product emissions but makes the incitement audible.

An exploration of distortion product synthesis as a compositional material exhibits particularly interesting results in spatial imagery. As observed by Kendall, Haworth, and Cádiz, the sound image generated through the combination of tones appears very close to the head of the listener and closer than the actual sound source, the loudspeaker: "The perceived spatial location is governed by interaural time differences (ITDs) and interaural intensity differences (IIDs) in a way that is quite analogous to what happens with headphone lateralisation" (Kendall, Haworth, and Cádiz, 2014, p.16). The effect is especially strong with slight head moves during listening. The DPOAEs, in conjunction with pulsar synthesis, incorporate the tonal, spatial and micro-temporal aspects of composition. Possible further compositional experiments may involve the generation of combination tones via pulsaret waveform and envelope cross modulation. Included in the portfolio, the composition 'Oto Sieve' utilises combination tones generated through a pulsaret table of harmonic series combined with a hamming function as an envelope. Applied within the work methods of envelope dilation and sieve-based offset allowed the generation of continuous and asynchronous tone clusters.

Chapter 2

Analysis - Resynthesis

In an article “Similarity and Repetition”, Neuhaus, Knösche, and Friederici proposes a notion of *relational thinking* as a useful tool in discussing perceptual processing of musical form. The term *relational thinking* (Beziehendes Denken) was originally coined by Hugo Reimann to denote a process conjoining parts into the whole (Wason, Marvin, and Riemann, 1992; Riemann, 1992). Defined as such, the notion of *relational thinking* echoes Felix Salzer’s discussion on *structural hearing* - a strategy for reconstructing the logical flow of composition by conjoining musical patterns (Salzer, 1962). The *structural hearing* operates across horizontal and vertical axes. When related to horizontal order, the process of *structural hearing* focuses on linear relations between adjoining and non-adjoining objects and sections. The vertical order relates to the hierarchical structure of objects and layers superordinate and subordinate to each other. The process of *structural hearing* extracts objects on both axes at multiple temporal levels. It mobilises their relational analysis and a generative role in the emergence of form. As such, the model echoes described earlier in the text nested structure of the form proposed by James Tenney (see 3). In line with the discussion presented in 1 notions of analysis and synthesis are deeply intertwined. The chapter focuses on the compositional potential of computational methods modelling the structural analysis and resynthesis of audio signals. Particular attention is given to methods mobilising the micro-temporal level of composition.

Kromhout (2021) proposes an understanding of sound reproduction which goes beyond a simplified schema of input-output. In a simplified view, the analysis process is concerned with maximum fidelity to the input material. Jonathan Sterne describes this as the “desire to capture the world and reproduce it as it is” (Sterne, 2003, p.218). This is an idealised perspective based on the assumption that the process of analysis unfolds through a "chain of passive intermediaries" (Latour, 2010, p.482) an attempt to attain a complete similitude between the original and its copy. An extreme case of such a perspective makes the analytic process undetectable, and its role as a mediator vanishes. However, the logical channel leaves a trace in most real application cases. As observed by Bruno Latour, the channel matters; it is not just a "gap between causes and consequences” (Latour, 2010, p.483) which ought to be erased, but a site of an operative marking and shaping of the signal. In this view, the productive aspect of the channel, often called ’noise’, becomes an active element of the analysis/re-synthesis chain. Each chain of the synthetic processes is essentially constructive and leaves traces in the final audible output of the process. In work with the nuPG program, the signal is not simply channelled between input and output. Each chain of the synthetic process should be considered constructive, leaving traces in the final audible output. The channel is the origin of pulsations, modulations, compressions and amplifications.

Processes of analysis and re-synthesis are similar to the activity of making traces in sound. The result-synthesised sound bears a mark of trace-making analytic procedures. Peter Ablinger points out that the interest in analysis/re-synthesis processes is not in the precise reproduction itself but an exploration of the "border-zone between abstract musical structure and the sudden shift into recognition - the relationship between musical qualities and "photorealism": the observation of "reality" via "music" (Ablinger, 1999, p.1). In a series of works *Quadraturen*, Ablinger engaged with a set of concerns related to the analysis (segmentation) of sound sources into time-frequency grids and further employment of these as a score to be reproduced in different media, e.g., traditional instruments, player piano or synthetic sound. A computer program developed between 1995 and 1997 in collaboration with the Institut für Elektronische Musik und Akustik - IEM in Graz allowed Ablinger to analyse the frequency of a sound file using a semitone filter that allowed variable intervallic sizes. The process was called "a temporal screening" and consisted of a series of rapid analyses, possibly also in real-time. Ablinger's work emphasises the shift between continuity and discreteness of various sound instances represented within the digital medium. The analysis and sequencing of temporal and spectral screens highlight a procedure of "broken continuity": the composition as disassembling of sound material and its digital reconstruction in time[1]. Drawing a parallel between a digital representation of images and sound, Jennie Gottschalk calls such compositional procedure "pixelation" (Gottschalk, 2016, p.161). Similarly to a division of an image into discrete blocks of colour, the "temporal screening" segments the audio into discrete analytic blocks. Processes of segmentation and chaining are characteristic of the technique of Wavelet Transform (WT). An implementation of WT for the nuPG program and its compositional employment forms an integral part of this chapter. We begin with a brief introduction to the technique and its application in the domain of artistic sound practice.

2.1 Wavelets

Wavelets in the audio domain decompose a signal into a hierarchy of temporal segments. The transform can be thought to function on the verge of three temporalities. In wavelet analysis (2.1), the signal is sampled horizontally according to so-called "octaves" (each composed of a given set of "voices") which correspond to iteratively refined resolutions. With wavelets of constant shape, the number of samples is doubled from one octave to the next higher one: the resolution gets refined from octave to the next-level one. Analysed data function in time as a cascade of sampled widows of variable width.

The data of the wavelet transform analysis consists of a stage of abstraction. Within this stage, numerous transformations become possible (see (2.1) for an overview of established and experimental methods). The synthesis stage of the wavelet transform unfolds the abstraction in time: and re-temporalizes it anew. This new version of the input signal carries traces of both processes, analysis and abstraction.

WT is a signal analysis technique. It was developed as an alternative to the short-time Fourier Transform (STFT). The WT, unlike the STFT, treats low-frequency regions as long in time and narrow in the frequency range, whereas high-frequency regions are short in time and wide in the frequency range. The wavelet representation scales the duration of each windowed part according to its frequency. Although

---

[1] Abliger often starts with a recording on a CD. For example see *Quadraturen IV (Selbstportait mit Berlin)* (1995-1998) (Peter Ablinger (composer), 2000; Gottstein, 1998)

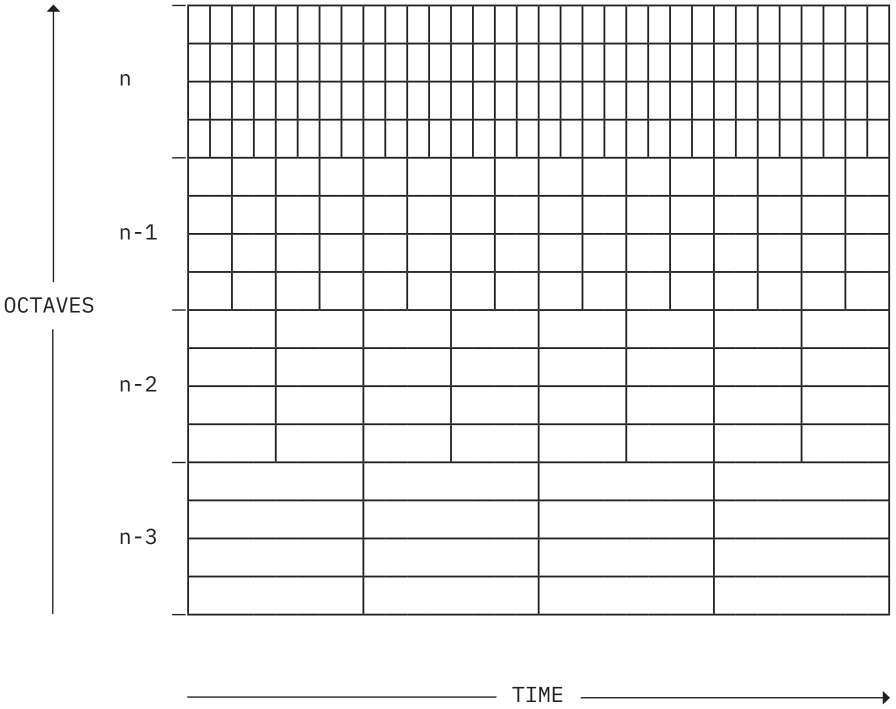


FIGURE 2.1: A simplified schematics of the wavelet sampling method. The process of sampling at each octave refines the resolution by the power of two. Low frequencies are analysed using a wide set of sampling segments (bottom of the graph), while high frequencies employ the highest resolution (top of the graph). Each octave is composed of a constant number of voices. In our case, the number of voices equals 4

wavelets represent a departure from Fourier analysis, they are also a natural extension[2].

The development of wavelets can be linked to several historical moments, starting with Alfred Haar (1909) work on a compact wavelet transform[3] and work by Dennis Gabor, 1947 on acoustical quanta, which are constructed similarly to wavelets, and applied to similar purposes (For an overview of historical origins of wavelets see: Meyer, 1993, pp. 13–31). The analysis method proposed by Denis Gabor, as

---

[2] The wavelet analysis can also be compared to constant Q analysis, which has been used in audio research since the late 1970s (Petersen, 1981; Stautner, 1983). A constant Q analysis implemented filter banks where the bandwidth varied proportionally with frequency (Roads, 1996b, p.578). In that respect, these techniques model the human ear, which exhibits similar time-frequency resolution characteristics. The auditory system performs filter band analysis with a filter-dependent bandwidth. These measured bandwidths are called *critical bandwidths* (Petersen and Boll, 1983).

[3] The Haar function, being an odd rectangular pulse pair, is the simplest and oldest orthonormal wavelet with close support. The transform has been widely used in image compression (see (Castelman, 1979))

shown in (1.1) uses an expansion of a function into a two-parameter family of elementary wavelets obtained from one basic wavelet by shifts in the time and the frequency domain (Feichtinger and Strohmer, 2012). A new representation, initially derived from Gabor's ideas by Jean Morlet, 1983[4], is based on the "wavelet," an elementary function of constant body rather than with a continuous envelope: to vary the frequency, this function is shrunk or stretched in time. This process is called dilation (or compression). The basic concept of wavelet analysis in audio research is to use distributions of suitable windowed and translated partials to make the decomposition more suitable to certain aspects of the sound (Hernández and Weiss, 1996). A signal is analysed using wavelets scaled in duration, frequency, and time. The reconstruction of certain class signals can be achieved by adding them in prescribed amounts. Each region of the time-frequency plane is associated with a time-domain signal (the wavelet), which can be thought of as a bandpass filter's impulse response (Vetterli and Herley, 1992). This is where the Wavelets theory can be incorporated into a discourse on pulsar synthesis. The wavelet process of dilation of a constant shape can be likened to the pulsar formant frequency modulation process. As shown in Section 1.1, the shape of the pulsaret waveform remains constant, but it gets stretched and shrunk in time while the fundamental and formant frequency varies (see Figure 1.4). Several authors noted the formal similarity between wavelets and granular synthesis (Kronland-Martinet, 1988; De Poli, 1989, See:). De Poli stating that" signal analysis by wavelets (functions limited both in time and frequency domain) justifies granular synthesis and permits to reconstruct signals by summation of opportune grains" (De Poli, 1989, p.1).

Various transformations of sound using wavelet analysis are possible. The uses of wavelets in speech and music have been examined in papers by Boyer, Grossman, Kronland-Martinet, and other researchers (Kronland-Martinet, 1988). MUSIC V instruments were developed by Kronland-Martinet to perform various synthesis and resynthesis tasks using wavelet transforms. A wavelet can be adapted to detect specific characteristics of a signal, he explains. In addition, he outlines three ways in which wavelet transforms can be used to create variations on a signal. As a first step, the analysis wavelet can be changed in the reconstruction to preserve some aspects of the movement while changing others. In essence, this is a cross-synthesis. The second option is to change the wavelet domain coefficients, which act much like an equaliser, changing the behaviour of the signal at a particular frequency level. Lastly, the reconstruction could use different translations and dilations than the analysis, or values could be removed or interpolated. The technique is used to shift pitch and compress time.

While changing the wavelet function during an inverse transform may seem straightforward, in practice, choosing the correct wavelet function can be crucial. For example, suppose one uses Mallat's smooth wavelet for analysis and Haar's boxcar wavelet for resynthesis. Since the difference between these wavelets is very large, the result is that a great deal of noise has been added to the signal. Such effect may be unwanted by a scientist searching for an accurate representation of the movement; however, artistic experimentation may be a fruitful area of exploration for new timbres. The real-time signal processor SYTER, designed by J. F. Allouis at INA-GRM and built by Digilog, allowed a set of transformations such as transposition without changing duration, slowing down or speeding up the sound without pitch transpositions, time-varying filtering, and cross-synthesis between two sounds

---

[4] The Morlet Wavelet has a Gaussian shape similar to Gabor transform

by resynthesis with the modulus information of one sound and the phase information of another sound (Allouis, 1982). (Risset and Wessel, 1982) calls such transformations *intimate* as they allow to address particular timescales of sound leaving others unaltered. The set of compositional methods discussed in 2.2 implements techniques discussed by Kronland-Martinet and experimental approaches incorporating elements of the process of pulsar synthesis as an abstract modulation between analysis and re-synthesis stages of the transform.

A phase vocoder is an example of such a transformation. Phase vocoders are analysis-synthesis systems that use a discrete Fourier spectrum of a time-varying input signal as a basis for analysis (Moorer, 1978; A discussion on a complete mechanics of an implementation of a digital phase vocoder can be found in: Portnoff, 1976). The shortcoming of Fourier representation includes its instability to signal deformations at high frequency and the emergence of the artefacts in re-synthesis of attack transients (Röbel, 2003). Lostanlen and Hecker, 2019 describe a technique of time-frequency scattering as a wavelet-based phase vocoder which overcomes such limitations. Additionally, scattering transform has been used for improved audio classification (Andén and Mallat, 2011).

Florian Hecker's collaboration with applied mathematician Vincent Lostanlen resulted in a series of compositions incorporating wavelets; for example, *Modulator (Scattering Transform)* released on a cassette tape by Editions Mego is a remix of Hecker's electronic piece *Modulator* (2012). *Inspection (Maida Vale Project)* seven-channel music for synthetic voice and computer-generated sound, performed live at BBC's Maida Vale studios in London and was broadcast on BBC Radio 3 in December 2016, marking the BBC's first ever live binaural broadcast. An extended version, *Inspection II*, was released as a CD by Editions Mego, Vienna and Urbanomic, Falmouth, the UK, in Fall 2019. The technical context of these works has been described in Lostanlen's PhD thesis (Lostanlen, 2017) and in a series of articles, see (Lostanlen, 2018) and (Lostanlen and Hecker, 2019). The time-frequency scattering algorithm works through a process of a two-stage cascade: a constant-Q wavelet transform (CQT) and the extraction of spectrotemporal modulations with wavelets in time and log-frequency[5]. According to Siedenburg, Fujinaga, and McAdams, the constant-Q transform "iteratively decomposes a signal into layers of coefficients by cascading wavelet transforms on the low-pass filtered modulus (i.e. envelope) of the previous layer. Its first layer of coefficients encodes a signal's frequency content, whereas the second and further layers mainly capture the temporal evolution" (Siedenburg, Fujinaga, and McAdams, 2016, p.6). The transform can be thought of as a series of filters, logarithmically spaced in frequency, which maps signal from time to frequency domain (Schörkhuber and Klapuri, 2010; Bradford, Ffitch, and Dobson, 2008). The transform extracts information from the signal by "iteratively translating, scaling and demodulating" (Hecker, 2021, p. 61). The analysis protocol gets updated at each iteration to produce—re-synthesise—signal result closer to the original input or its desired transformation. In Hecker's FAVN, sorting of time-frequency scattering paths by the relative amount of energy can be thought of as an abstraction of wavelet-based data, a transformation happening between the analysis and re-synthesis stages of the process.

The following section focuses on applying wavelet methods within the nuPG program.

---

[5] For a mathematical explanation of the time-frequency scattering formula see (Andén, Lostanlen, and Mallat, 2019; Andén, Lostanlen, and Mallat, 2018; Andén, Lostanlen, and Mallat, 2015)

2.2 nuPG Wavelets Implementation

The implementation of wavelet analysis/re-synthesis within the new pulsar generator program incorporates a two-stage process. First, the input signal is analysed using a Discrete Wavelet Transform (DWT). The analysis requires a memory size specification (a buffer) to hold analysis data, a hop value (an overlap between analysis frames), a data windowing type and an analysis wavelet type. The choice of analysis wavelets corresponds to the GNU Scientific Library code (2021)[6]. The result of the analysis chain is processed in the wavelet coefficient domain (See below for examples of such processes) or re-synthesised using an Inverse Discrete Wavelet Transform (IDWT). The IDWT object returns analysis data into the time domain. The IDWT uses the same or different wavelet type for re-synthesis.

The wavelet transform within the nuPG program is incorporated in two modes: per-pulsar and segment based. Per-pulsar wavelet transform generates a new analysis-resynthesis chain at each pulsar trigger. In its basic form, the transform takes a pulsaret as an input through a chain of analysis and resynthesis wavelets and returns a new sound particle. Let us call this new particle a *pulsarLET* (2.2). The user has control over a choice of analysis and re-synthesis wavelets, which can be set as a constant value or as a variable, e.g. a loop across all available wavelets where at each iteration a new shape is being selected (2.4). The transform has a radical effect on a spectrum of a pulsaret (2.3), shifting the frequency bins towards the shape of wavelets used for analysis and resynthesis.

---

[6] Each wavelet is given a numeric code, with k designating a specific wavelet member from the wavelet family:

- 0-8 gsl wavelet daubechies, k = 4,6,8,10,12,14,16,18,20
- 9-17 gsl wavelet Daubechies centered, k = 4,6,8,10,12,14,16,18,20
- 18 gsl wavelet haar, k=2
- 19 gsl wavelet haar centred, k=2
- 20-30 gsl wavelet spline, k=103, 105, 202, 204, 206, 208, 301, 303, 305, 307, 309
- 31-41 gsl wavelet spline centered k=103, 105, 202, 204, 206, 208, 301, 303, 305, 307, 309

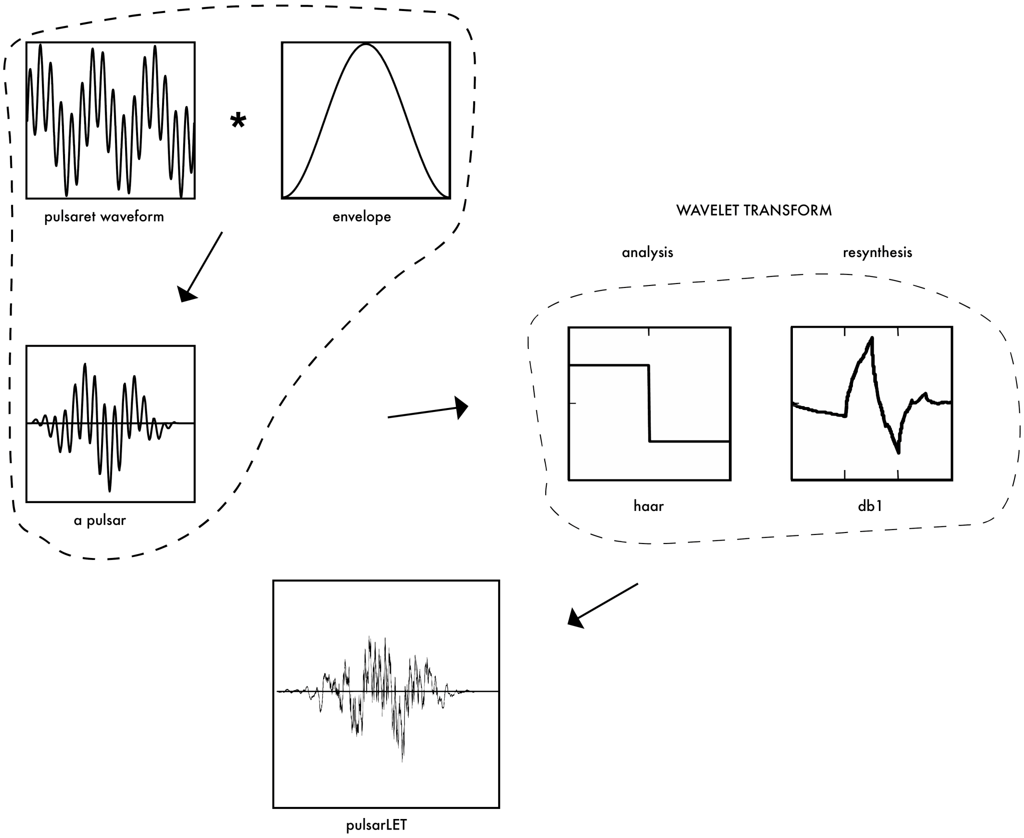


FIGURE 2.2: A process of wavelet transformation per-pulsar. The resulting sound particle here, called *pulsarLET*, is a result of a wavelet-based analysis process - using Haar wavelet - and wavelet-based resynthesis - using Daubechies wavelet.

The application of wavelet transform at the level of the pulsaret is linked to the transition process between discrete and continuous structures. The audio rendering (13) demonstrates the iterative process of wavelet transform operating on a pulsar train.

Audio 13. nuPG wavelet per pulsar
Listen to the accompanying example:
'nuPG_wlt_variableSegment.mp3'

FIGURE 2.3: A spectrogram in dBV (logarithmic frequencies, analysis window size = 2048) of a pulsar to pulsarLET conversion. Notice not only changes in the frequency domain but also widened temporal region - here visible as an increase from 2 to 4 columns

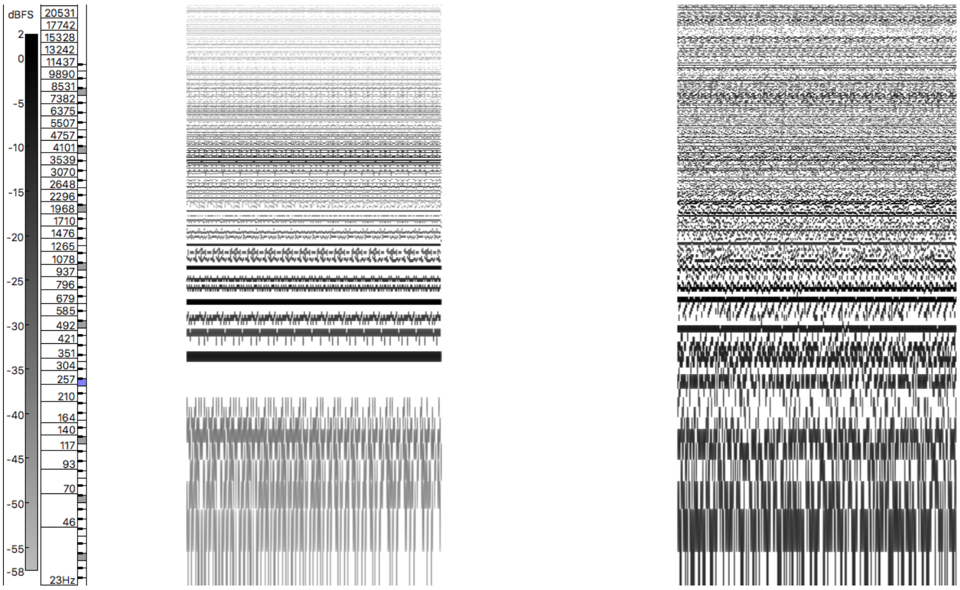


FIGURE 2.4: A spectrogram in dBV (logarithmic peak bins, analysis window size = 2048) of a pulsar to pulsarLET conversion effect on a continuous stream (341 pulses per second). The selection of analysis and resynthesis wavelets iterates over two lists analysis = [19, 12, 20, 31] and resynthesis = [1, 2, 3, 4], that is one analysis-resynthesis pair per 0.002932551319648 of a second and 0.011730205278592 of a second per iteration across four pairs

A segment-based model applies an analysis-resynthesis chain to the output of a pulsar stream based on user-specified duration[7].

The segment-based wavelet transform introduces a set of different processes to be selected between the analysis and resynthesis stages:

- Multiplication of wavelet coefficients between two sources. These could be two trains of the nuPG or nuPG and an external source, e.g. sound sample.
- Per-channel wavelet-based equaliser - gives powerful and dynamic controls to rebalance a sound's highs, lows, and midrange. Per-channel allows control of the equalisation of all channels independently or together (inspired by INA | GRM Spectral Transform tool 'Equalize' https://inagrm.com/en/store/product/1/spectral-transform). In many ways, wavelets can be used in the same way as traditional subtractive synthesis, i.e. they are compelling and flexible. To create a different and perhaps more exciting sound, start with a signal that has a high spectral content (some people might call it "noisy") and remove specific frequency ranges. Using wavelet representations, we can impose amplitude envelopes to varying scales in order to change the frequency content of the signal over time.
- Wavelet-based coefficients freeze. The process passes particular scales while suppressing (zeroing coefficients) from others
- Wavelet-based coefficients reversal - shifts magnitudes and phases upside-down
- Coefficients modulation by an arbitrary function

Code 12. Accessing wavelet data for arbitrary processing

Data can be accessed and processed between the analysis and re-synthesis stages. The method requires the conversion of Discrete Wavelet Transform (DWT) data into Fast Fourier Transform (FFT) format to access magnitudes and phases of analysis coefficients

```
//re-interpret DWT data as FFT data
(
~synthesis.trainInstances[0][3] = \filterIn -> {|in, trigFreq|
        var chain;
        chain = DWT(LocalBuf(1024), in);

        chain = PV_Copy(chain, LocalBuf(1024));

         chain = chain.pvcalc(1024, {|mags, phases|
    //Try uncommenting each of these lines in turn and re-running the
   synth:
    // Arbitrary filter, arbitrary phase shift
    //[mags * {1.5.rand}.dup(mags.size), phases + {pi.rand}.dup(phases.
   size)];
        [mags.reverse, phases.reverse]; // Upside-down!
        //[mags.differentiate, phases.differentiate]; // Differentiate
   along frequency axis
    // ".rotate" doesn't work directly, but this is equivalent
```

---

[7] nuPG uses Wavelet implementation for SuperCollider developed by Nick Collins (https://composerprogrammer.com/code.html). The choices of wavelet are those available in the gsl wavelet library code (http://www.gnu.org/s/gsl/manual/html_node/DWT-Initialization.html). Additionally, Paul Addison's The Illustrated Wavelet Transform Handbook' was an invaluable source for understanding the relationship between input, wavelets transform, and output (Addison, 2017)

```
        //[mags[30..] ++ mags[..30], phases[30..] ++ phases[..30]];
    }, frombin: 0, tobin: 250, zeroothers: 0);

        IDWT(chain).dup(2) // inverse DWT
};
)
```

2.3 Portfolio

In compositions *Sieve Decantation (i)* and *Sieve Decantation (ii)* a series of wavelet-based methods have been used. The basic material of the composition consists of a sieve-based sound material processed through a set of per-channel wavelet equalisers. The base sound material was void of apparent phrase-like articulation, void of recognisable logical progression, dense and spectrally compact. The wavelet transform operated on it through processes of subtraction, muting and attenuating of selected frequency regions. The aesthetic intended to explore formal articulations from within the source material rather than with it. Such an approach echoes Xenakis' *Bohor* (1962) and Tenney's *Fabric for Che* (1967) with their condensed and globally non-differentiated forms. At the time of their composition, both works were considered a violent attack on formal aspects of Western music. The sieve operates here in two modes. First, it controls frequency regions to mute. Second, it generates time intervals within which different muting sieves are applied. The first application operates on the spectrum (vertically) and the second on time (horizontally), which echoes a distinction made by Hugo Reimann in his conceptualisation of 'relational thinking. Peter Ablinger operationalised the vertical-horizontal relationships in the work *Weiss / Weisslich 22* (1986-96) (Ablinger, 1997). A material point of departure for the work by Ablinger was a recording collection of all symphonies by Haydn, Mozart, Beethoven, Schubert and Bruckner. Christian Scheib describes the transformation process as a vertical-horizontal axes flip and a shift from time experience as linear to momentary. The transformation consisted of a translation from sample values distributed on a horizontal axis (density vs time) into spectral data (frequency vs time). A new rendering, condensed into segments of forty seconds per composer, resulted in a saturated spectrally signal. As observed by Jennie Gottschalk, symphonies from each composer sound out distinct saturations, and through the process of "verticalisation", a complete collection of symphonies becomes "a quality of sound" (Gottschalk, 2016, p.160). Operationalisation of the vertical-horizontal relationship on a micro-temporal scale can be observed in a method of magnitudes and phases shifting as implemented within the nuPG program (12)

The wavelet-based transformation of signal functions in a chain of processes. For example, the analysis/re-synthesis output can be further processed through a wavelet-based pitch-shifting algorithm. The application of it as a glissando is most prominent in the work *Sieve Decantation (ii)* where a new iteration of the algorithm is applied at each phrase segment of approximately two seconds.

Chapter 3

Speculative Sonification

The field of sonification constitutes a rich domain of research focused on the auditory representation of data, processes and scientific models. Sonification can be a great complement for creating multimodal approaches to the interactive representation of data, models and processes, especially in contexts where phenomena are at stake that unfold in time, and where observation of parallel streams of events is desirable Hermann, Hunt, and Neuhoff, 2011. In an orthodox view, sonification research concerns subjects making judgments, finding regularities, being sensitive to similarities and differences, and making responses that can be evaluated for their truth. Incorporation of sonification into a creative practice opens up its applicability as a speculative tool problematising the source domain of data, the science of auditory representation and the aesthetics of compositional material.

A compositional practice with pulsar synthesis establishes a dialogue with the field of sonification through an object of astrophysical pulsar. The discovery of pulsars by Jocelyn Bell Burnell in 1971 happened through observation of radio signals (See Burnell (1984)). The astrophysical pulsars are phenomenal objects: rapidly rotating neutron stars that send out beams of radio waves which, like lighthouse beams, sweep around the sky as the star rotates. They are amazingly precise timing devices that can be used as clocks for testing relativity theory and may be used for timekeeping and navigation. With a diameter of only about 15 kilometres and a density comparable to that of the nucleus of an atom, they also provide a laboratory for some extreme physics. Pulsars appear to ‘pulse’ since the beam of light they emit can only be seen when it faces the Earth.

Described in this chapter, the work ’Synthetic Pulsar’ (2021), is a sound installation for a multichannel sound and synthetic voice commissioned by the CTM Festival and Deutschlandradio Kultur[1]. The work looks at the pulsar as a dazzling multidimensional object of scientific and creative focus. While the pulsar traverses a wide range of disciplines – astrophysics and sound technology – the workings and nature of such an object can never be fully captured, hence remaining incompletely understood. The compositional material for the work consisted of the nuPG program, rotational profiles of astrophysical data (formatted as comma-separated values)[2], a

---

[1] `https://www.ctm-festival.de/festival-2021/programme/exhibition/ventrilogues/synthetic-pulsar-by-marcin-pietruszewski-alex-freiheit`. A video documentation of the work can be accessed here: `https://vimeo.com/849933230`

[2] The source data for the sonification of astrophysical pulsars comes from the European Pulsar Network (EPN) Database a comprehensive cataloguing of pulsars undertaken by the Max Planck Institute for Radioastronomy in Effelsberg together with seven partner European institutions [3]. In a raw form, the EPN data sets consist of a timing index and rotational profile of selected pulsars. A common pulsar data set format contain also an intensity vs. longitude plot, as well as, a polarisation angle position vs longitude plot

digital model of attraction-repulsion system, libretto and a synthetic voice. The synthetic voice is based on the voice of Polish performance artist and vocalist Alex Freiheit (aka SIKSA) and as the narrator, thematises processes of synthetic formulation. In the staging of the work, the listener was invited to explore spatially distributed formations of sound clusters, discrete pulses, tones and textures sonifying processes of attractions and repulsions. To describe the work, I coined a term 'speculative sonification'. The term signifies a double position. First, the combination of the data (pulsar rotation profiles) with the attraction-repulsion model is conjectural rather than relational in regard to observed and measured phenomena. Second, by sonifying "impossible" objects and their behaviour the work mobilised the ways intuition and bodily (spatial) experience take an integral part in the formation of our worldly *episteme*. Knowledge does not restrict its corpus to the sum total of perfectly certain propositions but includes conjectures, articulations of possibilities and experimental settings.

A compositional practice derived from sonification research has gained recognition over the last decade, shifting focus from epistemological concerns to aesthetic[4]. In 2009, composer Marcus Schmickler together with programmer Alberto de Campo and astronomers Dr. Michael Geffert (Bonn University's Argelander Institute for Astronomy) and Dr. Kerstin Jaunich (Deutscher Musikrat) worked on a set of astrophysical data sonification pieces. Among them was one devoted to astrophysical pulsars (See `http://piethopraxis.org/projects/bonner-durchmusterung/`). The work *Pulsars/Neutron Stars* forms a part of a larger sound composition by composer Marcus Schmickler conceived as part of a data sonification project *Bonner Durchmusterung (Bonn Patternization)*. The project consists of ten compositions (all 3 minutes 30 seconds long) derived from the sonification of various sets of astrophysical data [5]. The rotational profile of a pulsar is sensitive to the observational approach and requires a significant period of time to be defined. As a basis for the work *Pulsars/Neutron Stars* 16 of the selected profiles were sonified using the PG (2004) program by Tommi Keranen 2.2. The rotational profile was mapped into synthesis parameters such as dynamic, fundamental frequency, formant frequency and spatialization. These sonification renderings formed a basis of the compositional arrangement, a form of "auditory comparability", where multiple rotational profiles can be heard simultaneously. An interesting case, not strictly a sonification, is the use of pulsars by Gerard Grisey. The French composer incorporated the signal of pulsars Vela and 0359-54 as integral elements of the composition *Le Noir De L'Etoile* written for percussion ensemble Les Percussions de Strasbourg. In two sections of the composition (22'07-24'53 and 35'53-37'59) a recording of pulsars' signal as detected by radiotelescope Nancay was played back over an array of 12 loudspeakers distributed around the audience[6].

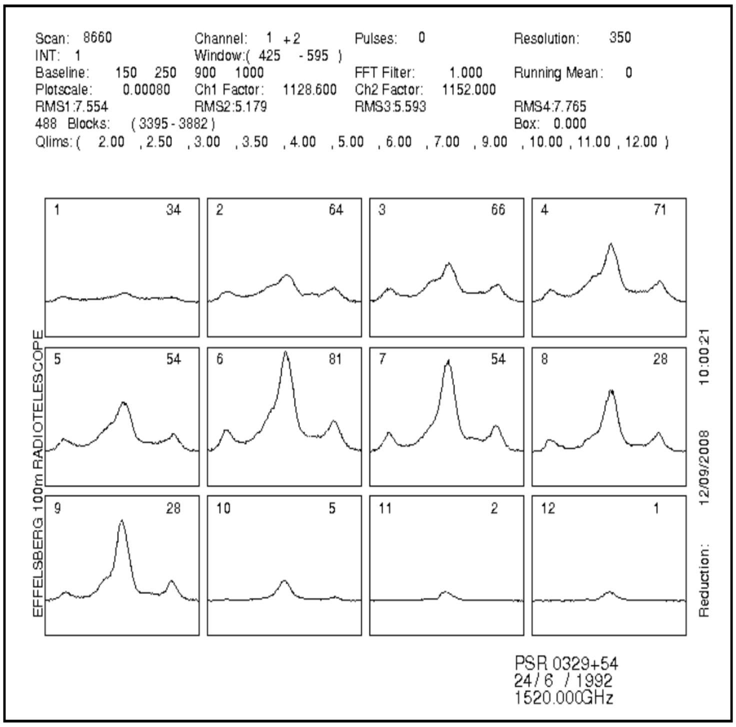
The following section details the sonification approach and the dynamic system employed in the composition

---

[4] see (Bovermann, Rohrhuber, and Campo, 2011), a talk by Marcus Schmickler on his compositional practice with various models of data `https://zkm.de/en/media/video/stroemungen-marcus-schmickler` and also his two releases 'Politiken der Frequenz' (2014) (together with Julian Rohrhuber) `http://editionsmego.com/release/eMEGO-191-TA-115` - which deals with sonification of various problems of contemporary mathematics - and 'Particle/Matter-Wave/Energy' (2019) - derived from sonification model of gravitational movements

[5] At the moment of writing the *Bonner Durchmusterung (Bonn Patternization)* was presented only in the form of a sound installation, no audio documentation of the project was released. The analysis presented here is based on a personal communication between the author and Marcus Schmickler

[6] At the premiere of the work in 1992, an initial idea to stream live pulsars' signal from Nancay to the concert hall in Brussels was not feasible technically. Thus the recording was used instead.



```
Scan:   8660          Channel:   1  +2         Pulses:    0            Resolution:     350
INT:   1              Window:(  425    - 595  )
Baseline:    150   250   900    1000           FFT Filter:     1.000     Running Mean:    0
Plotscale:     0.00080  Ch1 Factor:    1128.600  Ch2 Factor:   1152.000
RMS1:7.554            RMS2:5.179               RMS3:5.593               RMS4:7.765
488  Blocks:    ( 3395 - 3882 )                                         Box:   0.000
Qlims: (   2.00   , 2.50   , 3.00   , 3.50   , 4.00   , 5.00   , 6.00   , 7.00   , 9.00   , 10.00 , 11.00 , 12.00 )
```

FIGURE 3.1: A data of PSR B1919+21 (a first observed pulsar). It is worth noting the loop-like structure. Image source: (Kramer et al., 2003)

3.1 Physical Modelling Sonification: Gravity, Attraction, Repulsion

The work 'Synthetic Pulsar' (2021) integrates pulsar data sonification with an experimental system modelling the physical behaviour of various objects. The modelled behaviour includes gravity, attraction and repulsion. A direct inspiration to engage with these particular behaviours was the 'Organic' movement of composition *Clangtint* (1997) by Curtis Roads. The section's sound material consists of sound samples from the world of living creatures: animals, birds, and insects. The sound files are merged with electronically generated pulsars and sine waves. It is worth mentioning that 'Organic' is the first piece in which pulsar synthesis was used. Pulsations function as a common thread throughout this movement. Also in this section, Roads had approached sonically a process of attraction-repulsion. Edgar Varèse hinted at this idea when he observed: "When new instruments will allow me to write music as I conceive it, taking the place of the linear counterpoint, the movement of sound masses, or shifting planes, will be clearly perceived. When these sound masses collide the phenomena of penetration or repulsion will seem to occur". In 'Organic', sounds gravitate around designated zones of attraction. Sounds scatter around zones of repulsion. One zone of attraction is the period from 20.5 seconds to 23 seconds. Dozens of brief sounds burst in a cluster within this zone. A sparse zone of repulsion appears at 55 seconds. A second zone of attraction appears in the region between 3:00 and 3:06, the finale of the work. The attraction-repulsion process in the compositional practice of Roads was achieved through an iterative process of synthesis and editing. For the composition 'Synthetic Pulsar' (2021) a set of algorithms modelling physical behaviours was employed. The set of algorithms designed by Fredrik Olofsson and Mark d'Inverno lets the user build and experiment with systems as they are running. With the help of these tools, one can quickly try out ideas around simple data-to-sound mappings, as well as code very complex agents with strange behaviours[7]. The mapping strategy employed in the work separates the dynamic model from the sound synthesis. The dynamic model (3.2 and 3.2) computes objects with a given size and obeys simple physics (e.g. gravity). The interaction of these objects can be traced from within the system and their properties such as acceleration and spatial position retrieved. The retrieved values from the dynamic system can be used as an additional source modulating parameters of the nuPG program. Utilising the introduced already JiT approach mapping between the properties and objects of the system and the parameters of the synthesis is flexible.

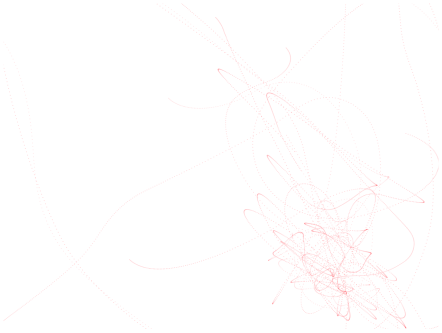


FIGURE 3.2: Dynamic System. Attraction of 6 objects

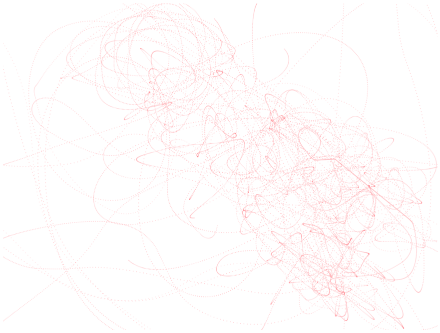


FIGURE 3.3: Dynamic System. Notice points of attraction, objects converging and oscillating in certain zones of the system

The mapping strategy used in 'Synthetic Pulsar' (2021) included: linking the object's acceleration and spatial position to formant frequency and spatial position in a binaural renderer, linking the acceleration of two or more objects into fundamental and formant frequency of the corresponding number of pulsar trains. Particularly interesting was the mapping of the system's physics into multiple parameters of synthesis (fundamental and formant frequencies, spatial position, modulation frequency and envelope dilation). The attraction system can be heard in the work at around 16 minutes and 20 seconds. Multiple pulsar streams attract each other, depending on the object's mass, heavier objects pull lighter ones into their orbit, causing accelerated oscillations. At certain points, the Doppler-like effect can be heard as an emergent property of the system and the mapping.

3.2 Libretto and Synthetic Voice

The text of the libretto is partially derived from this thesis and it incorporates the astrophysical and technological contexts of pulsar. The synthetic voice, designed in collaboration with an AI agency Birds on Mars[8], was based on the voice of Polish

---

[7] https://fredrikolofsson.com/f0blog/work-with-mark-reduniverse-a-simple-toolkit/

[8] https://www.birdsonmars.com/

performance artist and vocalist Alex Freiheit (aka SIKSA). The voice opens the sound installation with a brief text about pulsars as astrophysical objects and then appears again to expand their covering to include technological and aesthetic realms. The voice participated in formation of pulsar as a synthetic object. When working on the work, I was particularly interested in the writing of Anne-Françoise Schmid on epistemology. Schmid has developed the concept of "integrative object" (*objet intégratif*) in order to capture the increasingly interdisciplinary types of objects with which contemporary scientists have to deal and to reflect on how sciences create something new or how innovation happens:

> Integrative objects are no longer complex objects, explicable by convergence and the overlapping of disciplinary perspectives. They are not given, they are unknown, and their dimensions are fragments of disciplines, but articulated in a heterogeneity such that a milieu, a mid-site, is necessary to conceive and to receive them. They are formed of superpositions of bodies of knowledge, unified by the partial and mobile continuity of intention (...) The integrative object induces a relation to the virtual and to the future that cannot be neglected (Schmid and Hatchuel, 2014, p. 136)

Schmid proposes that we think of such an object as a multi-dimensional entity, each of whose dimensions is a different discipline or discourse, and whose contours are sketched out according to the points at which each of these disciplines falls short of capturing it. Since these dimensions can never be added to each other so as to synthesise a whole object, it is constituted ('made ready' for presentation) each time through the partial perspective and intentions of a given researcher. The richness of Schmid's model, and its application to contemporary objects, resides in this incomplete, problematic status that prevents integrative objects from ever being presented as 'readymade'. Knowledge emerges from the realm of messy confusion, of trying out, in the inertia of preconceived opinions; it is again and again in need of those epistemological acts that bring unexpected impulses into scientific development. This perspective resonated with my thinking on the pulsar as not only an object of sound technology but as a synthetic (integrative) object.

Conclusions

The project resulted in a new implementation of pulsar synthesis in the form of the New Pulsar Generator (nuPG) program, a portfolio of compositions, three album releases, and numerous sound installations and performances. Artistic outputs of the project centred around formal and conceptual topics mobilised by systematic engagement with the technique of pulsar synthesis and its computational implementation. The renewed interest in the technique and deepened reflection on its compositional possibilities was one of the main objectives of the research project.

A broad theme of the project was an engagement with the computer program, which goes beyond its use as a simple, functional tool. This engagement calls for opening up, disassembling and mobilising elements of the program within artistic practice. Opening up the computer program first allowed me to observe in a finer resolution the linkage between definition—in the form of a computer programming language—and the sound synthesis process. Existing sources point towards definitions encapsulated either in diagrammatic or symbolic forms. An identification of the link between the definition, domain of signal representation, and the audible consists of an essential addition to existing knowledge. The project brought up a more nuanced understanding of the definition of the synthesis process (diagrammatic and formal-symbolic), the computational paradigm within which such definition functions, and the formal compositional considerations. Notably, close attention to the relationship between computational objects and aesthetic processes revealed their interdependence. The central object of the compositional engagement with the nuPG program, the loop, can not be considered solely in the context of the artistic process. The object acquires its definition exactly as being in-between domains of creative and computational realms. The incorporation of information theoretical and media archaeological discourse allowed to mediate formal and aesthetic ramifications of the nuPG program and account for its particular relationship with temporality. A systematic engagement with this diverse nature of objects of computer music composition is one of the critical outcomes of the project.

One of the problems tackled by the research was the so-called "black boxiness" of technology, including computer programs. Throughout the design of the New Pulsar Generator and analysis of computational processes integral to the technique of pulsar synthesis, we have noticed that the notion of the "black box" persists. The process of its problematisation does not make the notion invalid, and it instead shifts its application to a different level of operationalisation. Within current research, we have attempted to open the "black box" conceived as a computer program, only to encounter the same notion at the level of the object-oriented programming paradigm. The notion of a "black box" is a valuable delimitation for operational engagement with computational objects.

The incorporation of a text-based conversational model of the interface was developed as a more flexible and faster way to respond to the issue of the program as a fixed and closed entity. The text of the programming language allows for grasping the formal and computational procedures behind the visual interface and the audible synthetic stream. Such a design decision has a direct didactic extension. The format of work where the function of objects is learned through their placement and relationship within an operating piece of the computer program seems useful. as a generalised example for learning programming and sound synthesis. An immediate connection between textual definition, a visual representation, and the sonorous output consists of a robust pedagogical environment. Programming concepts such as encapsulation or polymorphism become graspable and operational when listened to and seen.

A model of work which incorporates the simultaneous development of a computer program and understanding of a sound theory is applicable in other related fields of research. One such field is data sonification. Sonification work methods are conceived in the dialogue between sound designers and domain experts. The model of a computer program as a semi-open system is a practical mediating point as it allows quick implementation and perceptualisation of scientific data models. At the moment of writing these conclusions, the author has already applied such knowledge and skills as the Leverhulme Research Fellow in Data Perceptualisation and Aesthetics as part of Project Radical Sonification (`https://projectradical.github.io/about/`).

The technical analysis brackets out the human, political and social aspects of the functioning of the pulsar synthesis technique. The media archaeological method developed as part of this research avoids discussing the socio-historical contexts surrounding the emergence and functioning of media technologies. Further work can be done by looking at specificities of old and new – also connected to results of this research – communities of artists working with the technique of pulsar synthesis. A point for further research could consist of the open-source culture from within which the programming implementation of pulsar synthesis emerged and the context of openness and closure within these communities. Three volumes of compilations 'Pulsar Scramble' have been released to date, among other uses of thenuPG program by other artists[9].

Currently, there is a debate regarding the political economy of today's increasingly closed technologies, such as digital rights management and black-boxed media technologies, where one cannot open them without destroying them. The issue of open-source computer programs is pertinent to this debate. It is possible to overlook the current technological cultures of consumer devices that are impossible to tinker with by focusing on early computer programs for sound synthesis. The current work escapes this criticism as far as the archaeological method consists of only an element within a more extensive set of approaches applied here—the primary outcome being the process of composition.

An essential aspect of composition as a research method was a systematic engagement with formal aspects of the synthesis method. The technique of pulsar synthesis focuses on clearly delineated temporal dimensions: the micro and meso scales. A side effect of such focus is less attention to a global—macro-scale— a form of a composition. This aspect of composition often comes from outside of the technique (e.g. sieves, as presented within current research). This is not so much a criticism of the technique but a constraint which has to be noted. None of the techniques solves all problems across all possible time scales. The operationalisation of

---

[9] The 'Pulsar Scramble' compilations can be streamed through Bandcamp: volume 1 `https://pwgen20.bandcamp.com/album/pulsar-scramble`, volume 2 `https://pwgen20.bandcamp.com/album/pulsar-scramble-vol-2` and volume 3 `https://pwgen20.bandcamp.com/album/pulsar-scramble-vol-3-2`. Other uses of the nuPG program include: Jules Rawlinson 'Pulsar Retcon' (`https://pixelmechanics.bandcamp.com/album/pulsar-retcon`), Victor Moragues 'Scan Pulsar Detours' (`https://victormoragues.bandcamp.com/album/scan-pulsar-detours`)

one scale happens at the expense of the others, which may receive less attention. This can be summed down to the economy of choice and operation. In the context of formal considerations, one of the research findings is the prevalent appearance of a formal object which attempts to account for multiple time scales and relationships between them. Respecting all differences, similar formal considerations can be found within the process of cascading the time-frequency scattering algorithm, a model of compositional multi-scale form, and Gabor's acoustical quanta. From an analytic point of view, all of these address. All of them also fail at it. These models can be compared to different coordinate systems, each fitting to capture a particular object feature, its temporal placement and relationships. None of them is capable of capturing all of the objects and their characteristics. An essential detour in defining the synthetic process was to look at the corresponding analytic model. Thus simultaneously unfolding the diagrammatic and formal-symbolic definitions of the pulsar synthesis technique, the thesis proposed a close reading of the development of the Acoustic Quanta as introduced by Dennis Gabor. The model of Acoustic Quanta became a helpful lens through which synthetic processes of pulsar synthesis can be mediated and extended. Traces of this mediated reading are tangible within the nuPG program - see envelope dilation function and wavelet analysis-re-synthesis processes described in (2).

The project resulted in an extended set of compositional methods incorporated into the practice with the pulsar synthesis technique. A set of methods developed as part of the sieve implementation[10] connects a realm of microsound composition with formalism, on the one hand, inherited historically (i.e. Xenakis), and on the other, opening new avenues for experimentation with algorithmic structures in computer music composition. Particularly successful in this context are developed methods for sieve-based pulsar masking. A wide range of possible outputs from these formally simple methods can be seen as a valuable contribution to pattern-based approaches in composition. Further work can be developed in extending these methods to encompass polyrhythmic structures and parallel interlocked patterns. A perceptual aspect of pulsar synthesis/sieve conjunction requires additional experiments. For example, developing this line of work can contribute to the sonification of set-theoretical structures.

The incorporation of wavelet transform within the nuPG program mobilised a sound model fundamental for the emergence of the technique as a productive compositional strategy. Moreover, a reflection on the processes of analysis and re-synthesis allowed us to place the discussion on the nuPG program into broader information theoretical and aesthetic contexts. A set of wavelet methods applied per-pulsar and per-segment of pulsar train consists of the original development of the project and extends the timbral and compositional possibilities of the technique of pulsar synthesis. Further work in the context of analysis and auditory features classification of pulsar synthesis material is possible using recent developments in corpus-based methods. The album 'Pulsar Retcon' by researcher and composer Jules Rawlinson melds sound material sourced from the New Pulsar Generator (nuPG) program with corpus-based analysis using *Flucoma* toolset: `https://pixelmechanics.bandcamp.com/album/pulsar-retcon`.

The nuPG program featured within my works (sound installations, radio works and performances) beyond those discussed in the portfolio. The first major project

---

[10] See Appendix A for overview.

using the program was a performance ’(dia)Grammatology of Space’ (2016) conceived as a commission for the CTM Festival and Deutschlandradio Kultur[11]. The work incorporated a six-channel nuPG sound material, synthetic voice and a libretto written by the collective Laboria Cubonkis[12]. A stereo version of the performance was broadcast by the German Radio as part of Klangkunst series[13]. In 2021, I have developed a multichannel sound installation together with Jan St Werner (Mouse on Mars, Lithops) which combined the material generated with the nuPG program with distortion product synthesis. The work was presented at the House of The Culture of The World (HKW) in Berlin[14]. A full list of works with the nuPG is included in Appendix Five.

Aside from the artistic practice with the nuPG program, I have also developed a series of workshops popularising the technique among users. Between March and April 2020, I have run a series of workshops on the East Coast of Canada and USA, including The School for the Contemporary Arts, Simon Fraser University (Vancouver, BC, Canada), the Center for Computer Research in Music and Acoustics (CCRMA, Stanford, CA, USA) and The Center for Research in Electronic Art Technology (CREATE, Santa Barbara, CA, USA).

In addition to my own work, the nuPG program has been used extensively by other artists. A German electronic music duo Mouse on Mars has used the program for their sound installation ’Spatial Jitter’ (2022) which premiered at the Lenbachhaus in Munich. The installation was accompanied by an LP release[15]. A digital release ’Pulsar Acid’ (2020) by Jung and Tagen and Eric Frye features material recorded exclusively with the nuPG[16]. Their second release ’Phantom Acid’ (2022) combined the sound material from the nuPG with phantom word sequences by Diana Deutsch[17]. Released on a web label ETAT ’Taxonomy of Guffaws’ (2020) by RM Francis utilises short samples of laughs and synthetic voice loaded into tables of the nuPG program[18].

---

[11] https://archive2013-2020.ctm-festival.de/projects/commissioned-works/comissioned-works/diagrammatology-of-space/

[12] https://laboriacuboniks.net/

[13] https://www.hoerspielundfeature.de/ursendung-the-dia-grammatology-of-space-100.html

[14] https://archiv.hkw.de/media/en/texte/pdf/2021_1/programm_2021/the_sound_of_distance_begleitheft.pdf

[15] https://mouseonmars.bandcamp.com/album/spatial-jitter

[16] https://jungantagen.bandcamp.com/album/pulsar-acid

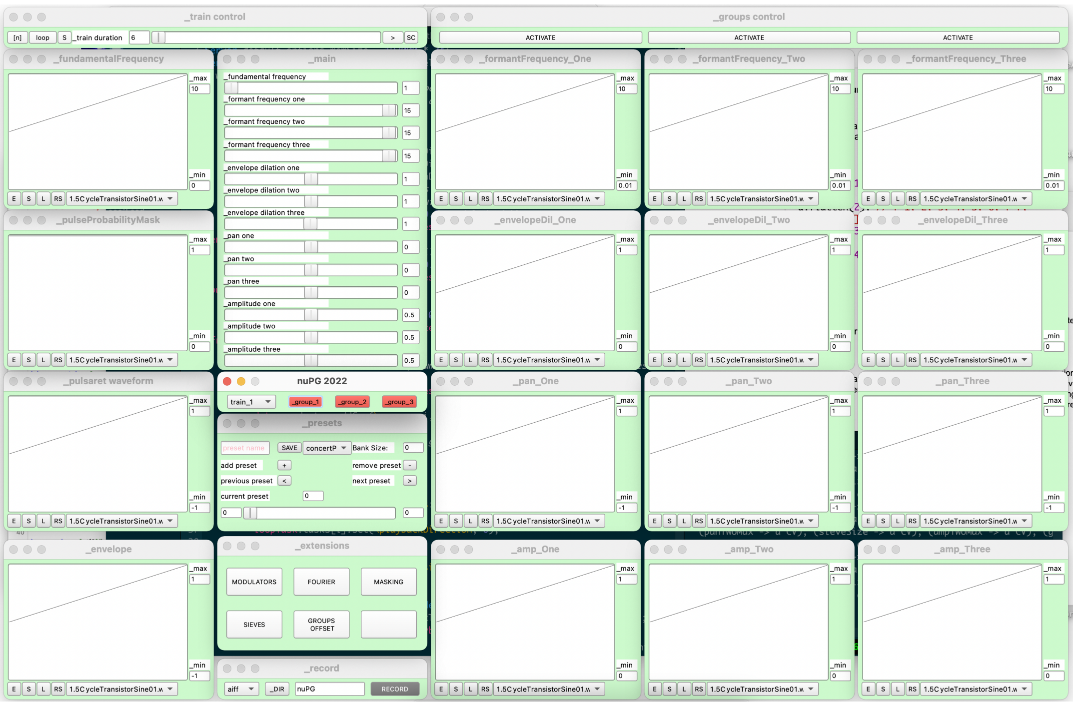
[17] https://ericfrye.bandcamp.com/album/phantom-acid

[18] https://etat.xyz/release/ATaxonomyofGuffaws

Appendix A

The New Pulsar Generator (nuPG) User Manual

A.1 Installation



_train duration 6

_fundamental frequency 1
_formant frequency one 15
_formant frequency two 15
_formant frequency three 15
_envelope dilation one 1
_envelope dilation two 1
_envelope dilation three 1
_pan one 0
_pan two 0
_pan three 0
_amplitude one 0.5
_amplitude two 0.5
_amplitude three 0.5

Bank Size: 0
current preset 0

FIGURE A.1: A global view of the New Pulsar Generator (nuPG) standalone program. The majority of parameters are controlled through a table editor. There are sixteen table editors ('fundamental frequency', 'pulse probability masking', 'pulsaret waveform', 'envelope', 'formant frequency one', 'envelope dilation one', 'panning one', 'amplitude one', 'formant frequency two', 'envelope dilation two', 'panning two', 'amplitude two', 'formant frequency three', 'envelope dilation three', 'panning three', 'amplitude three') visible at the startup and three additional ('frequency modulation amount', 'frequency modulation ratio', 'multiparameter modulation') accessible from within MODULATORS extension.

The New Pulsar Generator (nuPG) program comes packaged as a standalone application for the Mac OS system. Three architectures—Intel, M1 and M2—are supported. The program can also be launched from within SuperCollider IDE, running a script **nuPG_startUp2022.scd**. Running the program from the script requires the user to import source code and additional classes into the local SuperCollider directory. See (D) for details.

A.2 Table Editors

The program at the startup displays a set of graphic interface windows (A.1). The majority of synthesis parameters are controlled through a set of table editors. There are sixteen table editors visible at the startup and three additional accessible from the ’MODULATORS’ extension. Each table editor (A.2) consists of a table view, a set of functions (’RS’ - resize the table, ’L’ - load a table from a file, ’S’ - save a table to a file, ’E’ - open large table editor), a drop-down menu with a precompiled set of tables, and a set of values setting the table’s range. The table view display can be drawn on. There are 2048 values on the display. The data from a table can be copied pressing ’C’ button on the keyboard and pasted into any table-type object of the program by presssing ’P’ on the keyboard. Other keyboard shortcouts inculde ’R’ to reverse the data, ’I’ to invert it, ’M’ for multiplication

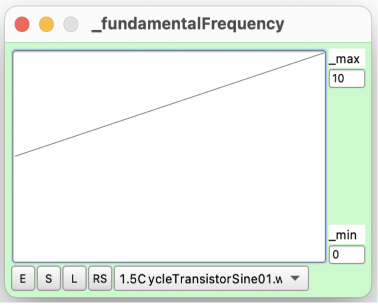
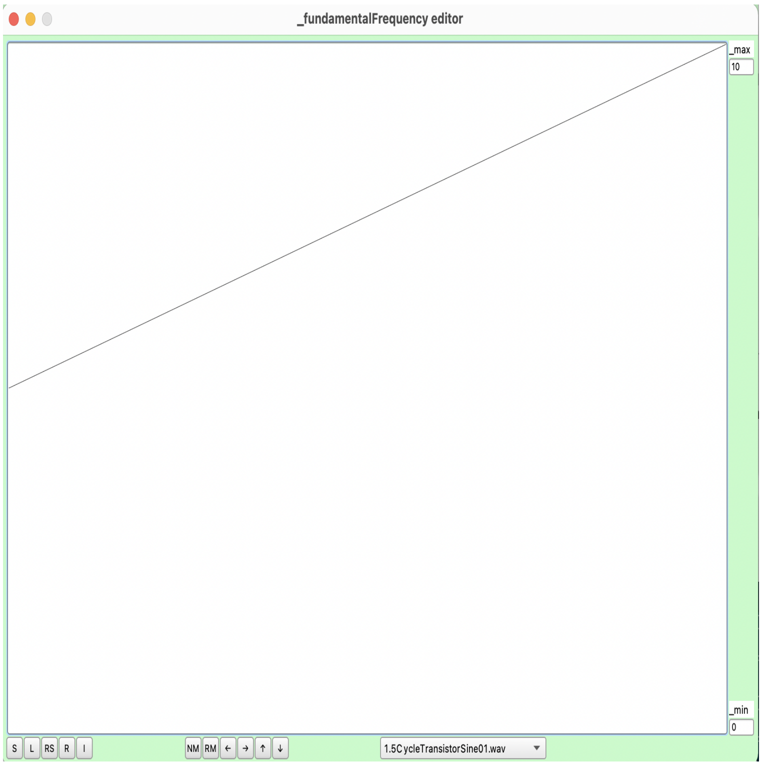


FIGURE A.2: Each table editor consists of a table view, a set of functions (’RS’ - resize the table, ’L’ - load a table from a file, ’S’ - save a table to file, ’E’ - open large table editor), a drop-down menu with precompiled tables, and a set of values setting the table’s range

Each table editor is supplied with an additional enlarged editor (A.3). Besides providing a finer resolution for table data, the editor gives access to additional methods: ’R’ - reverse table data, ’I’ - inverse table data along the horizontal axis, ’NM’ - multiplication, ’RM’ - ring modulation, ’←’ - shift left, ’→’ - shift right, ’↑’ - shift up, ’↓’ - shift down.

A.3 Train Control

The train playback is controlled via train control GUI (A.4). A set of controls displayed allows to interact with the pulsar train; the ’[n]’ button starts and stops the train, the ’loop’ button activates the looping function across tables, the ’S’ button synchronises multiple trains, ’train duration’ sets a duration of the train in seconds (available range is 1 to 120), the ’>’ button changes the direction of the table read, and the ’SC’ button opens a scanner editor (A.5). When the pulsar train starts, it only reads values from the ’MAIN’ control GUI. To activate the table reading, the ’loop’ function needs to be set to loop. The loop function by default loops through ’fundamental frequency’ and ’pulseProbabilityMask’ tables only. In order to activate one of three groups of ’formant freqency’, ’envelope dillation’, ’panning’ and ’amplitude’ tables set ACTIVATE to DEACTIVATE.



_max 10

_min 0

1.5CycleTransistorSine01.wav

FIGURE A.3: Each large table editor consists of a table view, a set of functions ('RS' - resize the table, 'L' - load a table from a file, 'S' - save a table to file, 'R' - reverse table data, 'I' - inverse table data along the horizontal axis, 'NM' - multiplication, 'RM' - ring modulation, '←' - shift left, '→' - shift right, '↑' - shift up, '↓' - shift down.), a drop-down menu with precompiled tables, and a set of values setting the table's range

The function is accessible via 'GROUPS CONTROL' GUI located at the top right of the screen.

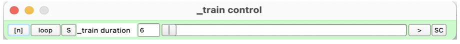


FIGURE A.4: A set of controls displayed allows for interaction with the pulsar train; the '[n]' button starts and stops the train, the 'loop' button activates the looping function across tables, the 'S' button synchronises multiple trains, 'train duration' sets a duration of the train in seconds (available range is 1 to 120), the '>' button changes the direction of the table read, and the 'SC' button opens a scanner editor.

The scanner GUI (A.5) allows reading values from tables according to user-specified path. Holding and dragging a grey bar across reads values from all active tables. The movement can be recorded by pressing 'R' button on the right side of the GUI. The movement is played back automatically after the release of the slider bar. The 'S' button on the left side of the GUI synchronises the scanning function across all program instances. The display to the right of the slider shows a table slot value between 0-2047 (2048).

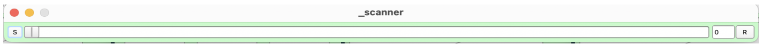


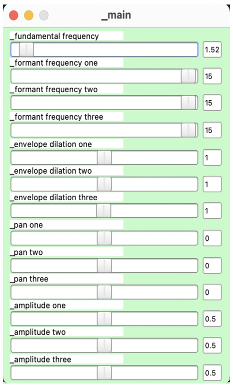
FIGURE A.5: The scanner GUI allows to read values from tables according to the user-specified path. Holding and dragging a grey bar across reads values from all active tables. The movement can be recorded by pressing 'R' button on the right side of the GUI. The movement is played back automatically after the release of the slider bar. The 'S' button on the left side of the GUI synchronises the scanning function across all program instances. The display to the right of the slider shows a table slot value between 0-2047 (2048).

A.4 Main Control

The 'MAIN CONTROL' GUI (A.6) provides a direct access to a set of key synthesis parameters: 'fundamental frequency', 'formant frequency one', formant frequency two', formant frequency three', 'envelope dilation one', 'envelope dilation two', 'envelope dilation three', 'pan one', 'pan two', 'pan three', 'amp one', 'amp two', and 'amp three'. The relationship between the parameters of the 'MAIN' and 'TABLES' GUI is a modulation (A.7). The table minimum and maximum values act as modulation depth. The value from the 'MAIN' GUI gets multiplied by or added to (this is the case for the panning parameter) a value from the table at each index according to to 'train duration' value. For example, with the 'train duration' of 6 seconds and 'MAIN' GUI's 'fundamental frequency' set to 2 cycles per second gets multiplied by a new table value at every 0.0029296875 of a second (6 ÷ 2048 = 0.0029296875).

A.5 Presets

The data of the program can be stored as a preset. The 'PRESETS' GUI (A.8) provides an access to saving, recalling and interpolating preset data. To add a preset, press the '+' button; to remove a preset, press the '-' button. Buttons '<' and '>' move up



_fundamental frequency 1.52
_formant frequency one 15
_formant frequency two 15
_formant frequency three 15
_envelope dilation one 1
_envelope dilation two 1
_envelope dilation three 1
_pan one 0
_pan two 0
_pan three 0
_amplitude one 0.5
_amplitude two 0.5
_amplitude three 0.5

FIGURE A.6: The 'MAIN CONTROL' GUI provides a direct access to a set of key synthesis parameters: 'fundamental frequency', 'formant frequency one', formant frequency two', formant frequency three', 'envelope dillation one', 'envelope dillation two', 'envelope dillation three', 'pan one', 'pan two', 'pan three', 'amp one', 'amp two', and 'amp three'.

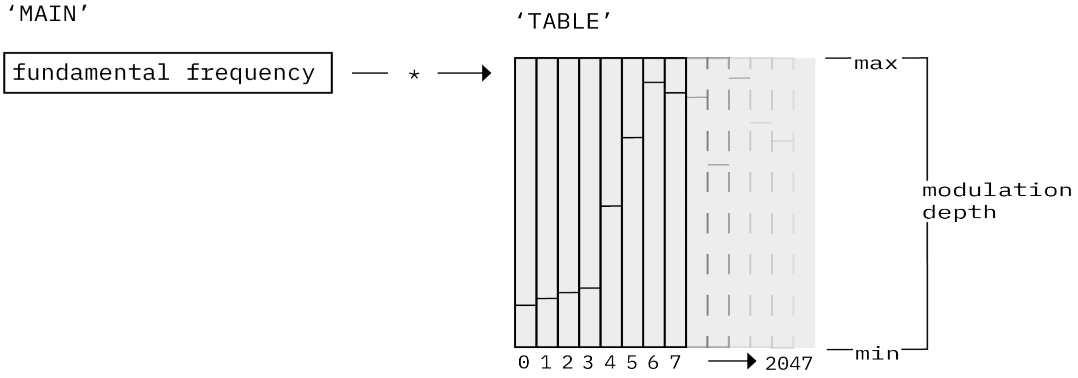


FIGURE A.7: The relationship between the parameters of the 'MAIN' and 'TABLES' GUI is a modulation. The table minimum and maximum values act as modulation depth. The value from the 'MAIN' GUI gets multiplied by or added to (this is the case for the panning parameter) a value from the table at each index according to to 'train duration' value. For example, with the 'train duration' of 6 seconds and 'MAIN' GUI's 'fundamental frequency' set to 2 cycles per second gets multiplied by a new table value at every 0.0029296875 of a second (6 ÷ 2048 = 0.0029296875) and down across a set of saved presets. Bank Size displays the number of available presets. Presets can be saved as a bank by giving it a name and pressing SAVE button. The new bank appears on the drop-down menu. At the bottom of the GUI user can access preset interpolation functionality. The interpolator requires a value setting for starting preset (current preset) and target preset. By moving the slider from right to left program interpolates between data stored on chosen presets.

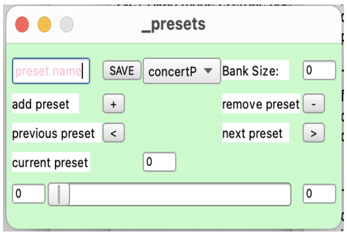


FIGURE A.8: The 'PRESETS' GUI provides an access to saving, recalling and interpolating preset data. To add a preset, press the '+' button; to remove a preset, press the '-' button. Buttons '<' and '>' move up and down across a set of saved presets. Bank Size displays the number of available presets. Presets can be saved as a bank by giving it a name and pressing SAVE button. The new bank appears on the drop-down menu. At the bottom of the GUI user can access preset interpolation functionality. The interpolator requires a value setting for starting preset (current preset) and target preset. By moving the slider from right to left program interpolates between data stored on chosen presets.

A.6 Recording

The output of the program can be recorded using built-in utility tool. The 'RECORD' GUI (A.9) facilitates recording of the audio output of the program. The user can specify the file format (.aiff, .wav, .caf), the directory (default directory on Mac OS is User Music SuperCollider Recordings), and the name of the file (by default files are named 'nuPG' + date + time).

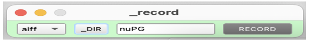


FIGURE A.9: The 'RECORD' GUI facilitates recording of the audio output of the program. The user can specify the file format (.aiff, .wav, .caf), the directory (default directory on Mac OS is User Music SuperCollider Recordings), and the name of the file (by default files are named 'nuPG' + date + time).

A.7 Extensions

The 'EXTENSIONS' GUI (A.10) gives access to additional processes embedded within the nuPG program.

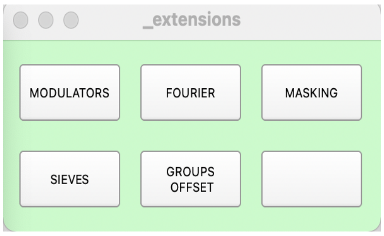


FIGURE A.10: The 'EXTENSIONS' GUI gives access to additional processes embedded within the nuPG program. Extensions include: 'MODULATORS', 'FOURIER', 'MASKING', 'SIEVES' and 'GROUP OFFSET'

A.7.1 Modulators

The 'MODULATORS' (A.11) extension provides two methods for modulation of the nuPG train: rate modulation and multiparameter modulation. The rate modulation applies a modulator to the emission rate of the train. The modulator consists of a sawtooth oscillator fed through a sample and hold function which holds input signal values when triggred. A combination of a modulator with a sample and hold function creates characteristic stepped interlocked patterns. The multiparameter modulation modulates fundamental frequency, formant frequency and amplitude by a dynamic cubic noise generator. The generator outputs polynomially interpolated random values at given rate.

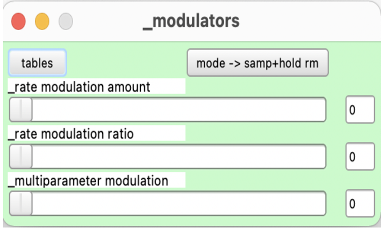


FIGURE A.11: A set of modulation functions (rate modulation and multiparameter modulation) with additional table controls

A.7.2 Fourier

The 'FOURIER' (A.12) editor facilitates a generation of tables through a summation of sinusoidal functions at given amplitude. Each of sixteen bars represents amplitude value from 0 to 1. The data from the editor (A.13) can be copied using 'C' button on the keyboard and pasted into any other table-type object of the program by pressing 'P' on the kayboard.

A.7.3 Masking

The 'MASKING' (A.14) GUI implements three types of pulsar masking: probability, burst and channel. The probability masking omits pulses according to a statistical gate function. When the function receives a trigger, it tosses a coin and either passes the trigger or doesn't. The probability value of 1 equals pulses all the time, and 0 equals no pulses. Values between 0.9 and 0.7 create a stochastically generated intermittent pattern of pulses and silences. The burst masking generates the

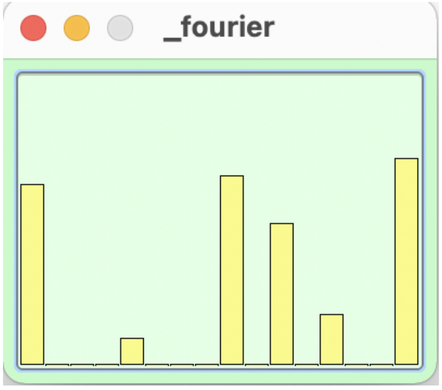


FIGURE A.12: Fourier Editor

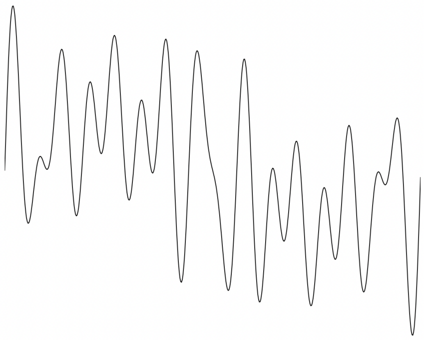
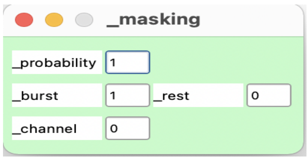


FIGURE A.13: A corresponding table data

mask according to two values: burst and rest. The burst value indicates the number of pulses to preserve, and the rest value sets the number of pulses to mask. The channel-based mask takes one value; the number of pulses to emmit interchangeably between available audio outputs. By default, the output of the nuPG program is set to stereo; the channel masking function adapts to variable outputs.



_probability 1

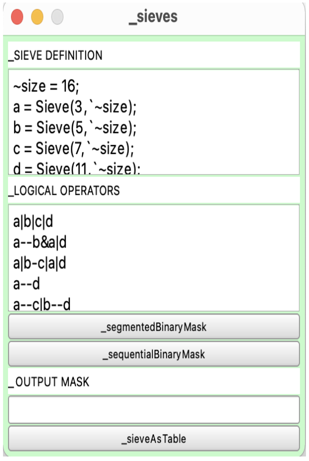
_burst 1 _rest 0

_channel 0

FIGURE A.14: The ’MASKING’ GUI implements three types of pulsar masking: probability, burst and channel

A.7.4 Sieves

Sieves are an original implementation within the nuPG program. The GUI (A.15) allows quick access to the definition of new sieves and their combination through logical operators (’|’ for union, ’&’ for section, ’–’ for symmetric difference, and ’-’ for difference), and the output as a masking sequence or table data. The sieve as mask functions in two modes: ’sequentialBinaryMask’ and ’segmentedBinaryMask’. The ’sequentialBinaryMask’ method takes an integer representation (points) of a sieve, replaces each point with one and fills all spaces (slots) that are not occupied with zeros. E.g., Sieve = [3, 7, 11, 15] will be converted to [1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1]. This approach is based on the Python function ’discreteBinaryPad’ developed by Christopher Ariza within his *athenaCL*. The ’segmentedBinaryMask’ method takes an integer representation (points) of a sieve, gets all the slots from given sieve - this is always one value less than the input list - replaces each integer with a corresponding number of 0 and 1 in a sequence. E.g. input = [2,3,3] -> output [0,0,1,1,1,0,0,0]. This approach is an original development within the nuPG. The technique of masking, when applied to discrete (infrasonic range) pulses creates rhythmic patterns of variable regularity; its application to continuous (audio range) produces amplitude modulation on timbre.



```
~size = 16;
a = Sieve(3,`~size);
b = Sieve(5,`~size);
c = Sieve(7,`~size);
d = Sieve(11,`~size);
```

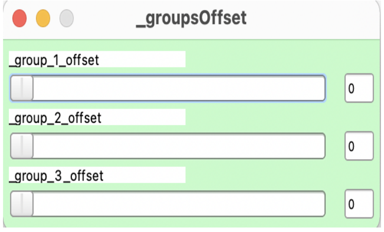
```
a|b|c|d
a--b&a|d
a|b-c|a|d
a--d
a--c|b--d
```

FIGURE A.15: The GUI allows quick access to the definition of new sieves and their combination through logical operators (' | ' for union, '&' for section, '–' for symmetric difference, and '-' for difference), and the output as a masking sequence or table data.

The output of the sieve method as a table generates data which can be pasted ('P') into any table-data object of the program. The size of the sieve can be made larger for finer resolution of the sieve-generated table.

A.7.5 Groups Offset

The method of group offset is an original development within the nuPG program. The 'GROUP OFFSET' GUI (A.16) provides access to a function offsetting one of three groups of formant frequency, envelope dilation, panning and amplitude by an offset value between 0 and 1 second.



_group_1_offset 0
_group_2_offset 0
_group_3_offset 0

FIGURE A.16: The interface allows to offset of one of three groups of formant frequency, envelope dilation, panning and amplitude by an offset value between 0 and 1 second

Appendix B

A printout of all 2048 values of a table

A list of floating point values representation of shape 3.3

```
0.0,
0.017817823216319,
0.035614419728518,
0.053368613123894,
0.071059301495552,
0.088665537536144,
0.1061665341258,
0.12354175001383,
0.14077085256577,
0.15783388912678,
0.17471121251583,
0.1913835555315,
0.2078320980072,
0.22403852641582,
0.23998495936394,
0.25565412640572,
0.27102929353714,
0.28609430789948,
0.30083373188972,
0.31523281335831,
0.32927742600441,
0.3429542183876,
0.35625064373016,
0.3691548705101,
0.38165590167046,
0.39374351501465,
0.40540847182274,
0.41664230823517,
0.42743736505508,
0.43778696656227,
0.44768542051315,
0.4571276307106,
0.466109842062,
0.47462889552116,
0.48268264532089,
0.49026989936829,
0.49739044904709,
0.50404471158981,
0.51023441553116,
0.51596194505692,
0.5212305188179,
0.52604442834854,
0.530408680439,
0.53432911634445,
0.53781253099442,
0.54086649417877,
0.54349917173386,
0.54571968317032,
0.5475378036499,
0.54896402359009,
0.55000948905945,
0.55068600177765,
0.55100607872009,
0.55098253488541,
0.55062907934189,
0.54995954036713,
0.54898852109909,
0.54773110151291,
0.54620236158371,
0.54441809654236,
0.54239422082901,
0.54014706611633,
0.53769302368164,
0.53504878282547,
0.532231092453,
0.52925688028336,
0.526143014431,
0.5229064822197,
0.51956427097321,
0.51613312959671,
0.5126296877861,
0.50907069444656,
0.50547248125076,
0.5018510222435,
0.49822217226028,
0.49460151791573,
0.49100413918495,
0.48744475841522,
0.48393771052361,
0.48049682378769,
0.47713533043861,
0.47386613488197,
0.47070133686066,
0.46765258908272,
0.4647308588028,
0.46194648742676,
0.45930901169777,
0.45682752132416,
0.45451006293297,
0.45236411690712,
0.45039647817612,
0.4486129283905,
0.44701865315437,
0.4456180036068,
0.44441431760788,
0.44341045618057,
0.4426081776619,
0.44200846552849,
0.44161152839661,
0.4414167702198,
0.44142258167267,
0.44162672758102,
0.44202613830566,
0.44261679053307,
0.44339397549629,
0.44435226917267,
0.44548526406288,
0.44678601622581,
```

0.44824674725533,
0.44985899329185,
0.45161357522011,
0.4535006582737,
0.45550984144211,
0.45763003826141,
0.4598496556282,
0.46215641498566,
0.46453762054443,
0.46698021888733,
0.46947038173676,
0.47199428081512,
0.47453734278679,
0.47708484530449,
0.47962185740471,
0.4821330010891,
0.48460277915001,
0.48701551556587,
0.4893553853035,
0.49160653352737,
0.49375295639038,
0.49577876925468,
0.49766811728477,
0.49940505623817,
0.50097393989563,
0.5023592710495,
0.5035457611084,
0.50451821088791,
0.50526201725006,
0.50576263666153,
0.50600612163544,
0.5059786438942,
0.50566720962524,
0.50505894422531,
0.50414174795151,
0.50290405750275,
0.50133472681046,
0.49942344427109,
0.49716034531593,
0.49453645944595,
0.49154350161552,
0.48817366361618,
0.48442032933235,
0.48027721047401,
0.47573918104172,
0.47080174088478,
0.46546134352684,
0.45971521735191,
0.45356151461601,
0.4469992518425,
0.44002825021744,
0.43264934420586,
0.42486426234245,
0.41667556762695,
0.40808668732643,
0.3991020321846,
0.38972693681717,
0.3799674808979,
0.36983081698418,
0.35932475328445,
0.34845814108849,
0.33724054694176,
0.32568237185478,
0.31379494071007,
0.30159020423889,
0.28908097743988,
0.27628082036972,
0.26320388913155,
0.24986518919468,
0.23628023266792,
0.22246526181698,
0.20843705534935,
0.19421294331551,
0.17981079220772,
0.16524897515774,
0.15054625272751,
0.13572186231613,
0.12079536169767,
0.10578664392233,
0.090715885162354,
0.075603537261486,
0.060470212250948,
0.045336656272411,
0.030223812907934,
0.015152608044446,
0.00014402809028979,
-0.014780982397497,
-0.029601499438286,
-0.044296752661467,
-0.05884613096714,
-0.07322920113802,
-0.087425790727139,
-0.1014159694314,
-0.11518017202616,
-0.12869922816753,
-0.14195430278778,
-0.15492710471153,
-0.16759975254536,
-0.17995493113995,
-0.1919758617878,
-0.20364640653133,
-0.21495102345943,
-0.22587485611439,
-0.23640371859074,
-0.24652419984341,
-0.25622355937958,
-0.26548993587494,
-0.27431216835976,
-0.28268003463745,
-0.29058402776718,
-0.29801559448242,
-0.30496707558632,
-0.31143167614937,
-0.31740343570709,
-0.32287740707397,
-0.32784956693649,
-0.3323167860508,
-0.33627679944038,
-0.339728474617,
-0.34267148375511,
-0.34510633349419,
-0.3470347225666,
-0.34845906496048,
-0.3493826687336,
-0.34980991482735,
-0.3497459590435,
-0.34919679164886,
-0.34816941618919,
-0.34667149186134,
-0.34471166133881,
-0.34229910373688,
-0.33944410085678,
-0.33615747094154,
-0.33245071768761,
-0.32833614945412,
-0.32382667064667,
-0.31893581151962,
-0.31367763876915,
-0.30806687474251,
-0.30211862921715,
-0.29584860801697,
-0.28927284479141,
-0.28240787982941,
-0.27527049183846,
-0.26787787675858,
-0.26024740934372,
-0.25239679217339,
-0.24434389173985,
-0.23610661923885,
-0.22770312428474,
-0.21915152668953,
-0.21046997606754,
-0.20167665183544,

-0.19278953969479,
-0.18382661044598,
-0.17480567097664,
-0.16574428975582,
-0.15665973722935,
-0.14756916463375,
-0.13848921656609,
-0.12943632900715,
-0.12042640894651,
-0.11147501319647,
-0.1025971993804,
-0.093807503581047,
-0.085119903087616,
-0.07654781639576,
-0.068104140460491,
-0.059800986200571,
-0.051649946719408,
-0.043661817908287,
-0.035846725106239,
-0.028214126825333,
-0.02077260427177,
-0.013530095107853,
-0.0064936331473291,
0.00033040496055037,
0.0069366036914289,
0.01332028582692l,
0.01947764120996,
0.025405636057258,
0.031102120876312,
0.036565840244293,
0.041796244680882,
0.046793770045042,
0.051559589803219,
0.056095737963915,
0.060405045747757,
0.06449119001627,
0.06835862249136,
0.072012551128864,
0.075458973646164,
0.078704632818699,
0.081756964325905,
0.084624134004116,
0.08731497079134,
0.089838951826096,
0.092206202447414,
0.094427414238453,
0.096513792872429,
0.098477192223072,
0.1003298535943,
0.1020845323801,
0.10375439375639,
0.10535298287868,
0.10689423233271,
0.10839232802391,
0.10986179858446,
0.11131732910872,
0.11277382820845,
0.11424635350704,
0.11575008928776,
0.11730024218559,
0.11891203373671,
0.12060075998306,
0.12238148599863,
0.12426933646202,
0.12627917528152,
0.12842574715614,
0.13072349131107,
0.13318663835526,
0.13582907617092,
0.1386643499136,
0.14170554280281,
0.14496546983719,
0.14845629036427,
0.15218977630138,
0.1561771184206,
0.16042898595333,
0.16495536267757,
0.16976563632488,
0.17486856877804,
0.18027217686176,
0.18598380684853,
0.19201000034809,
0.19835659861565,
0.20502862334251,
0.2120303362608,
0.21936510503292,
0.22703558206558,
0.23504345118999,
0.24338954687119,
0.25207397341728,
0.26109582185745,
0.27045339345932,
0.28014406561852,
0.290164321661,
0.30050978064537,
0.31117522716522,
0.32215458154678,
0.33344087004662,
0.34502628445625,
0.35690221190453,
0.36905914545059,
0.38148686289787,
0.39417430758476,
0.40710970759392,
0.42028042674065,
0.43367326259613,
0.44727426767349,
0.46106880903244,
0.47504165768623,
0.48917698860168,
0.50345832109451,
0.51786869764328,
0.53239071369171,
0.54700642824173,
0.56169754266739,
0.57644528150558,
0.5912304520607,
0.60603392124176,
0.62083595991135,
0.63561654090881,
0.65035575628281,
0.66503340005875,
0.67962914705276,
0.69412273168564,
0.70849376916885,
0.72272211313248,
0.73678737878799,
0.75066959857941,
0.76434874534607,
0.7778052687645,
0.79101973772049,
0.80397301912308,
0.81664615869522,
0.82902103662491,
0.84107953310013,
0.85280412435532,
0.864177942276,
0.87518441677094,
0.88580769300461,
0.8960325717926,
0.90584444999695,
0.91522926092148,
0.92417395114899,
0.93266588449478,
0.94069355726242,
0.94824576377869,
0.95531260967255,
0.96188473701477,
0.9679536819458,
0.9735119342804,
0.97855287790298,
0.98307079076767,
0.98706072568893,

```
0.99051880836487,
0.99344229698181,
0.99582886695862,
0.99767762422562,
0.99898833036423,
0.99976193904877,
1.0,
0.99970531463623,
0.99888151884079,
0.99753302335739,
0.99566543102264,
0.99328488111496,
0.99039870500565,
0.98701506853104,
0.98314279317856,
0.97879153490067,
0.97397196292877,
0.96869534254074,
0.96297371387482,
0.95681971311569,
0.95024710893631,
0.9432697892189,
0.9359022974968,
0.92816025018692,
0.92005944252014,
0.91161620616913,
0.902847468853,
0.89377027750015,
0.88440251350403,
0.87476223707199,
0.86486780643463,
0.85473757982254,
0.84439069032669,
0.83384615182877,
0.82312297821045,
0.81224048137665,
0.80121809244156,
0.79007506370544,
0.77883064746857,
0.76750409603119,
0.75611472129822,
0.74468106031418,
0.73322206735611,
0.72175621986389,
0.71030151844025,
0.69887614250183,
0.68749731779099,
0.67618209123611,
0.66494715213776,
0.65380877256393,
0.64278227090836,
0.63188296556473,
0.62112528085709,
0.61052304506302,
0.60008949041367,
0.58983719348907,
0.57977795600891,
0.56992292404175,
0.56028252840042,
0.55086636543274,
0.54168319702148,
0.53274118900299,
0.52404749393463,
0.51560854911804,
0.5074297785759,
0.49951612949371,
0.49187135696411,
0.48449835181236,
0.47739943861961,
0.4705758690834,
0.46402806043625,
0.45775556564331,
0.45175719261169,
0.44603088498116,
0.44057357311249,
0.43538174033165,
0.43045064806938,
0.42577520012856,
0.42134919762611,
0.41716584563255,
0.41321769356728,
0.40949648618698,
0.40599328279495,
0.40269863605499,
0.39960232377052,
0.39669373631477,
0.39396148920059,
0.39139384031296,
0.38897851109505,
0.38670289516449,
0.38455367088318,
0.38251745700836,
0.38058042526245,
0.37872847914696,
0.37694722414017,
0.37522199749947,
0.37353822588921,
0.3718808889389,
0.37023508548737,
0.36858579516411,
0.36691799759865,
0.36521676182747,
0.36346724629402,
0.36165457963943,
0.35976427793503,
0.35778197646141,
0.35569351911545,
0.35348510742188,
0.35114318132401,
0.34865471720695,
0.34600695967674,
0.34318763017654,
0.34018492698669,
0.33698761463165,
0.33358496427536,
0.3299668431282,
0.3261236846447,
0.3220466375351,
0.31772747635841,
0.31315872073174,
0.30833348631859,
0.30324575304985,
0.29789021611214,
0.29226237535477,
0.28635847568512,
0.28017562627792,
0.27371174097061,
0.266965508461,
0.25993660092354,
0.25262543559074,
0.24503330886364,
0.23716233670712,
0.22901557385921,
0.22059689462185,
0.21191097795963,
0.20296341180801,
0.19376061856747,
0.18430978059769,
0.17461897432804,
0.1646970063448,
0.15455354750156,
0.14419892430305,
0.13364432752132,
0.12290159612894,
0.11198326200247,
0.10090255737305,
0.089673385024071,
0.078310199081898,
0.066828049719334,
0.055242612957954,
0.043569970875978,
0.031826786696911,
0.020030111074448,
```

0.0081974389031529,
-0.0036533740349114,
-0.015504171140492,
-0.027336431667209,
-0.039131484925747,
-0.05087036639452,
-0.062533989548683,
-0.074103124439716,
-0.085558459162712,
-0.096880681812763,
-0.1080504655838,
-0.11904852092266,
-0.1298556625843,
-0.14045289158821,
-0.15082134306431,
-0.16094236075878,
-0.17079761624336,
-0.18036906421185,
-0.18963903188705,
-0.198590233922,
-0.20720578730106,
-0.21546931564808,
-0.2233649790287,
-0.23087744414806,
-0.23799200356007,
-0.24469450116158,
-0.25097152590752,
-0.25681027770042,
-0.26219871640205,
-0.2671255171299,
-0.27158018946648,
-0.27555298805237,
-0.27903491258621,
-0.28201791644096,
-0.28449484705925,
-0.28645926713943,
-0.28790578246117,
-0.28882986307144,
-0.28922784328461,
-0.28909707069397,
-0.28843578696251,
-0.2872431576252,
-0.28551930189133,
-0.28326535224915,
-0.28048330545425,
-0.27717608213425,
-0.27334758639336,
-0.26900267601013,
-0.26414704322815,
-0.25878739356995,
-0.25293120741844,
-0.24658691883087,
-0.23976381123066,
-0.23247194290161,
-0.2247222661972,
-0.21652649343014,
-0.20789711177349,
-0.19884730875492,
-0.18939103186131,
-0.17954286932945,
-0.16931810975075,
-0.15873257815838,
-0.14780271053314,
-0.13654553890228,
-0.12497854232788,
-0.11311963945627,
-0.10098727047443,
-0.088600151240826,
-0.075977444648743,
-0.063138529658318,
-0.050103113055229,
-0.0368910767138,
-0.023522483184934,
-0.010017519816756,
0.0036034893710166,
0.017320284619927,
0.031112477183342,
0.044959861785173,
0.058842230588198,
0.07273954898119,
0.086631961166859,
0.10049979388714,
0.1143237426877,
0.12808467447758,
0.14176388084888,
0.1553430557251,
0.1688042730093,
0.1821300983429,
0.19530355930328,
0.20830826461315,
0.22112835943699,
0.23374865949154,
0.24615448713303,
0.2583319246769,
0.27026775479317,
0.28194940090179,
0.29336509108543,
0.30450373888016,
0.31535515189171,
0.32590985298157,
0.33615919947624,
0.34609541296959,
0.35571160912514,
0.36500155925751,
0.37396016716957,
0.38258299231529,
0.39086654782295,
0.39880833029747,
0.40640652179718,
0.41366031765938,
0.42056971788406,
0.42713570594788,
0.43335983157158,
0.43924483656883,
0.44479408860207,
0.45001190900803,
0.45490321516991,
0.45947393774986,
0.46373063325882,
0.46768066287041,
0.47133201360703,
0.47469344735146,
0.4777744114399,
0.48058491945267,
0.48313558101654,
0.48543763160706,
0.48750281333923,
0.48934352397919,
0.49097222089767,
0.49240219593048,
0.49364703893661,
0.49472063779831,
0.49563708901405,
0.49641102552414,
0.4970571398735,
0.49759030342102,
0.49802559614182,
0.49837818741798,
0.49866336584091,
0.49889627099037,
0.49909228086472,
0.49926647543907,
0.49943393468857,
0.4996095597744,
0.49980813264847,
0.50004422664642,
0.50033187866211,
0.5006850361824,
0.50111746788025,
0.50164216756821,
0.50227200984955,
0.50301915407181,
0.5038954615593,

0.50491225719452,
0.50608015060425,
0.50740939378738,
0.50890934467316,
0.51058888435364,
0.51245617866516,
0.51451873779297,
0.51678329706192,
0.51925581693649,
0.52194154262543,
0.52484500408173,
0.52796983718872,
0.53131884336472,
0.53489410877228,
0.53869688510895,
0.54272753000259,
0.54698556661606,
0.55146968364716,
0.55617773532867,
0.56110692024231,
0.56625312566757,
0.57161194086075,
0.57717776298523,
0.58294439315796,
0.58890467882156,
0.59505081176758,
0.60137414932251,
0.60786533355713,
0.61451417207718,
0.62130987644196,
0.62824088335037,
0.63529509305954,
0.64245963096619,
0.64972120523453,
0.65706568956375,
0.66447860002518,
0.67194497585297,
0.67944920063019,
0.68697530031204,
0.69450706243515,
0.70202761888504,
0.7095200419426,
0.71696698665619,
0.72435092926025,
0.73165398836136,
0.73885828256607,
0.74594587087631,
0.75289845466614,
0.75969815254211,
0.76632660627365,
0.77276569604874,
0.77899765968323,
0.78500467538834,
0.79076892137527,
0.79627299308777,
0.8014999628067,
0.80643284320831,
0.811055123806,
0.81535089015961,
0.81930440664291,
0.8229004740715,
0.82612466812134,
0.82896262407303,
0.83140099048615,
0.83342677354813,
0.83502787351608,
0.83619266748428,
0.83691024780273,
0.83717066049576,
0.83696448802948,
0.83628302812576,
0.83511883020401,
0.83346492052078,
0.83131521940231,
0.82866454124451,
0.82550883293152,
0.82184445858002,
0.81766903400421,
0.81298106908798,
0.80778002738953,
0.80206602811813,
0.79584056138992,
0.78910577297211,
0.78186470270157,
0.77412152290344,
0.76588118076324,
0.75714951753616,
0.74793338775635,
0.73824065923691,
0.72807973623276,
0.71746021509171,
0.70639222860336,
0.69488704204559,
0.68295657634735,
0.67061346769333,
0.65787118673325,
0.6447439789772,
0.63124668598175,
0.61739486455917,
0.60320466756821,
0.58869296312332,
0.57387697696686,
0.55877465009689,
0.54340422153473,
0.52778470516205,
0.51193523406982,
0.49587532877922,
0.47962492704391,
0.46320429444313,
0.44663381576538,
0.42993414402008,
0.41312608122826,
0.39623057842255,
0.37926858663559,
0.36226105690002,
0.34522899985313,
0.32819333672523,
0.31117483973503,
0.29419422149658,
0.27727183699608,
0.26042792201042,
0.24368236958981,
0.22705481946468,
0.21056444942951,
0.19423007965088,
0.17807008326054,
0.16210237145424,
0.14634422957897,
0.13081255555153,
0.11552352458239,
0.10049276053905,
0.085735142230988,
0.07126497477293,
0.057095758616924,
0.043240364640951,
0.02971076965332,
0.016518278047442,
0.0036733192391694,
-0.0088144680485129,
-0.020936287939548,
-0.032684233039618,
-0.044051211327314,
-0.055031064897776,
-0.065618477761745,
-0.075809046626091,
-0.085599273443222,
-0.09498655050993,
-0.10396917909384,
-0.11254634708166,
-0.12071813642979,
-0.12848551571369,
-0.13585036993027,
-0.14281538128853,

-0.14938415586948,
-0.15556108951569,
-0.16135144233704,
-0.16676123440266,
-0.1717973202467,
-0.17646729946136,
-0.18077948689461,
-0.18474292755127,
-0.18836736679077,
-0.19166316092014,
-0.19464133679867,
-0.19731350243092,
-0.19969183206558,
-0.20178897678852,
-0.20361813902855,
-0.20519292354584,
-0.20652738213539,
-0.20763593912125,
-0.20853331685066,
-0.20923459529877,
-0.20975504815578,
-0.21011018753052,
-0.21031565964222,
-0.2103873193264,
-0.21034097671509,
-0.21019262075424,
-0.20995816588402,
-0.20965345203876,
-0.20929427444935,
-0.20889633893967,
-0.20847506821156,
-0.20804578065872,
-0.20762348175049,
-0.20722289383411,
-0.20685838162899,
-0.20654401183128,
-0.20629335939884,
-0.20611959695816,
-0.20603542029858,
-0.20605297386646,
-0.20618389546871,
-0.20643921196461,
-0.20682935416698,
-0.20736418664455,
-0.208052739501,
-0.20890355110168,
-0.20992428064346,
-0.21112203598022,
-0.21250295639038,
-0.21407262980938,
-0.21583573520184,
-0.21779617667198,
-0.21995703876019,
-0.22232063114643,
-0.22488842904568,
-0.22766101360321,
-0.23063823580742,
-0.23381905257702,
-0.23720155656338,
-0.24078315496445,
-0.24456022679806,
-0.24852846562862,
-0.25268277525902,
-0.25701719522476,
-0.26152503490448,
-0.26619878411293,
-0.27103021740913,
-0.27601045370102,
-0.28112977743149,
-0.28637790679932,
-0.2917437851429,
-0.2972157895565,
-0.30278170108795,
-0.30842882394791,
-0.31414365768433,
-0.31991249322891,
-0.32572093605995,
-0.33155426383018,
-0.33739727735519,
-0.34323450922966,
-0.34905004501343,
-0.35482788085938,
-0.36055153608322,
-0.36620444059372,
-0.37176987528801,
-0.37723100185394,
-0.38257092237473,
-0.38777253031731,
-0.39281898736954,
-0.39769333600998,
-0.40237876772881,
-0.40685868263245,
-0.41111651062965,
-0.41513600945473,
-0.41890123486519,
-0.42239639163017,
-0.42560622096062,
-0.42851576209068,
-0.43111047148705,
-0.43337628245354,
-0.43529972434044,
-0.43686771392822,
-0.43806782364845,
-0.43888825178146,
-0.43931779265404,
-0.43934592604637,
-0.43896290659904,
-0.43815958499908,
-0.43692767620087,
-0.43525967001915,
-0.43314877152443,
-0.43058910965919,
-0.427575558424,
-0.42410397529602,
-0.42017099261284,
-0.41577416658401,
-0.41091194748878,
-0.40558370947838,
-0.39978969097137,
-0.39353105425835,
-0.38681003451347,
-0.37962958216667,
-0.37199360132217,
-0.36390700936317,
-0.35537561774254,
-0.34640604257584,
-0.33700588345528,
-0.32718351483345,
-0.31694835424423,
-0.30631044507027,
-0.29528081417084,
-0.28387123346329,
-0.27209430932999,
-0.2599633038044,
-0.24749232828617,
-0.23469613492489,
-0.22159014642239,
-0.20819047093391,
-0.19451382756233,
-0.18057745695114,
-0.16639922559261,
-0.15199743211269,
-0.13739088177681,
-0.1225987970829,
-0.10764083266258,
-0.092536903917789,
-0.077307313680649,
-0.061972592025995,
-0.046553503721952,
-0.031070979312062,
-0.015546092763543,
-9.7477581562089e-14,
0.015546092763543,

0.031070979312062,
0.046553503721952,
0.061972592025995,
0.077307313680649,
0.092536903917789,
0.10764083266258,
0.1225987970829,
0.13739088177681,
0.15199743211269,
0.16639922559261,
0.18057745695114,
0.19451382756233,
0.20819047093391,
0.22159014642239,
0.23469613492489,
0.24749232828617,
0.2599633038044,
0.27209430932999,
0.28387123346329,
0.29528081417084,
0.30631044507027,
0.31694835424423,
0.32718351483345,
0.33700588345528,
0.34640604257584,
0.35537561774254,
0.36390700936317,
0.37199360132217,
0.37962958216667,
0.38681003451347,
0.39353105425835,
0.39978969097137,
0.40558370947838,
0.41091194748878,
0.41577416658401,
0.42017099261284,
0.42410397529602,
0.427575558424,
0.43058910965919,
0.43314877152443,
0.43525967001915,
0.43692767620087,
0.43815958499908,
0.43896290659904,
0.43934592604637,
0.43931779265404,
0.43888825178146,
0.43806782364845,
0.43686771392822,
0.43529972434044,
0.43337628245354,
0.43111047148705,
0.42851576209068,
0.42560622096062,
0.42239639163017,
0.41890123486519,
0.41513600945473,
0.41111651062965,
0.40685868263245,
0.40237876772881,
0.39769333600998,
0.39281898736954,
0.38777253031731,
0.38257092237473,
0.37723100185394,
0.37176987528801,
0.36620444059372,
0.36055153608322,
0.35482788085938,
0.34905004501343,
0.34323450922966,
0.33739727735519,
0.33155426383018,
0.32572093605995,
0.31991249322891,
0.31414365768433,
0.30842882394791,
0.30278170108795,
0.2972157895565,
0.2917437851429,
0.28637790679932,
0.28112977743149,
0.27601045370102,
0.27103021740913,
0.26619878411293,
0.26152503490448,
0.25701719522476,
0.25268277525902,
0.24852846562862,
0.24456022679806,
0.24078315496445,
0.23720155656338,
0.23381905257702,
0.23063823580742,
0.22766101360321,
0.22488842904568,
0.22232063114643,
0.21995703876019,
0.21779617667198,
0.21583573520184,
0.21407262980938,
0.21250295639038,
0.21112203598022,
0.20992428064346,
0.20890355110168,
0.208052739501,
0.20736418664455,
0.20682935416698,
0.20643921196461,
0.20618389546871,
0.20605297386646,
0.20603542029858,
0.20611959695816,
0.20629335939884,
0.20654401183128,
0.20685838162899,
0.20722289383411,
0.20762348175049,
0.20804578065872,
0.20847506821156,
0.20889633893967,
0.20929427444935,
0.20965345203876,
0.20995816588402,
0.21019262075424,
0.21034097671509,
0.2103873193264,
0.21031565964222,
0.21011018753052,
0.20975504815578,
0.20923459529877,
0.20853331685066,
0.20763593912125,
0.20652738213539,
0.20519292354584,
0.20361813902855,
0.20178897678852,
0.19969183206558,
0.19731350243092,
0.19464133679867,
0.19166316092014,
0.18836736679077,
0.18474292755127,
0.18077948689461,
0.17646729946136,
0.1717973202467,
0.16676123440266,
0.16135144233704,
0.15556108951569,
0.14938415586948,
0.14281538128853,
0.13585036993027,
0.12848551571369,

```
0.12071813642979,
0.11254634708166,
0.10396917909384,
0.09498655050993,
0.085599273443222,
0.075809046626091,
0.065618477761745,
0.055031064897776,
0.044051211327314,
0.032684233039618,
0.020936287939548,
0.0088144680485129,
-0.0036733192391694,
-0.016518278047442,
-0.02971076965332,
-0.043240364640951,
-0.057095758616924,
-0.07126497477293,
-0.085735142230988,
-0.10049276053905,
-0.11552352458239,
-0.13081255555153,
-0.14634422957897,
-0.16210237145424,
-0.17807008326054,
-0.19423007965088,
-0.21056444942951,
-0.22705481946468,
-0.24368236958981,
-0.26042792201042,
-0.27727183699608,
-0.29419422149658,
-0.31117483973503,
-0.32819333672523,
-0.34522899985313,
-0.36226105690002,
-0.37926858663559,
-0.39623057842255,
-0.41312608122826,
-0.42993414402008,
-0.44663381576538,
-0.46320429444313,
-0.47962492704391,
-0.49587532877922,
-0.51193523406982,
-0.52778470516205,
-0.54340422153473,
-0.55877465009689,
-0.57387697696686,
-0.58869296312332,
-0.60320466756821,
-0.61739486455917,
-0.63124668598175,
-0.6447439789772,
-0.65787118673325,
-0.67061346769333,
-0.68295657634735,
-0.69488704204559,
-0.70639222860336,
-0.71746021509171,
-0.72807973623276,
-0.73824065923691,
-0.74793338775635,
-0.75714951753616,
-0.76588118076324,
-0.77412152290344,
-0.78186470270157,
-0.78910577297211,
-0.79584056138992,
-0.80206602811813,
-0.80778002738953,
-0.81298106908798,
-0.81766903400421,
-0.82184445858002,
-0.82550883293152,
-0.82866454124451,
-0.83131521940231,
-0.83346492052078,
-0.83511883020401,
-0.83628302812576,
-0.83696448802948,
-0.83717066049576,
-0.83691024780273,
-0.83619266748428,
-0.83502787351608,
-0.83342677354813,
-0.83140099048615,
-0.82896262407303,
-0.82612466812134,
-0.8229004740715,
-0.81930440664291,
-0.81535089015961,
-0.811055123806,
-0.80643284320831,
-0.8014999628067,
-0.79627299308777,
-0.79076892137527,
-0.78500467538834,
-0.77899765968323,
-0.77276569604874,
-0.76632660627365,
-0.75969815254211,
-0.75289845466614,
-0.74594587087631,
-0.73885828256607,
-0.73165398836136,
-0.72435092926025,
-0.71696698665619,
-0.7095200419426,
-0.70202761888504,
-0.69450706243515,
-0.68697530031204,
-0.67944920063019,
-0.67194497585297,
-0.66447860002518,
-0.65706568956375,
-0.64972120523453,
-0.64245963096619,
-0.63529509305954,
-0.62824088335037,
-0.62130987644196,
-0.61451417207718,
-0.60786533355713,
-0.60137414932251,
-0.59505081176758,
-0.58890467882156,
-0.58294439315796,
-0.57717776298523,
-0.57161194086075,
-0.56625312566757,
-0.56110692024231,
-0.55617773532867,
-0.55146968364716,
-0.54698556661606,
-0.54272753000259,
-0.53869688510895,
-0.53489410877228,
-0.53131884336472,
-0.52796983718872,
-0.52484500408173,
-0.52194154262543,
-0.51925581693649,
-0.51678329706192,
-0.51451873779297,
-0.51245617866516,
-0.51058888435364,
-0.50890934467316,
-0.50740939378738,
-0.50608015060425,
-0.50491225719452,
-0.5038954615593,
-0.50301915407181,
-0.50227200984955,
```

```
-0.50164216756821,
-0.50111746788025,
-0.5006850361824,
-0.50033187866211,
-0.50004422664642,
-0.49980813264847,
-0.4996095597744,
-0.49943393468857,
-0.49926647543907,
-0.49909228086472,
-0.49889627099037,
-0.49866336584091,
-0.49837818741798,
-0.49802559614182,
-0.49759030342102,
-0.4970571398735,
-0.49641102552414,
-0.49563708901405,
-0.49472063779831,
-0.49364703893661,
-0.49240219593048,
-0.49097222089767,
-0.48934352397919,
-0.48750281333923,
-0.48543763160706,
-0.48313558101654,
-0.48058491945267,
-0.4777744114399,
-0.47469344735146,
-0.47133201360703,
-0.46768066287041,
-0.46373063325882,
-0.45947393774986,
-0.45490321516991,
-0.45001190900803,
-0.44479408860207,
-0.43924483656883,
-0.43335983157158,
-0.42713570594788,
-0.42056971788406,
-0.41366031765938,
-0.40640652179718,
-0.39880833029747,
-0.39086654782295,
-0.38258299231529,
-0.37396016716957,
-0.36500155925751,
-0.35571160912514,
-0.34609541296959,
-0.33615919947624,
-0.32590985298157,
-0.31535515189171,
-0.30450373888016,
-0.29336509108543,
-0.28194940090179,
-0.27026775479317,
-0.2583319246769,
-0.24615448713303,
-0.23374865949154,
-0.22112835943699,
-0.20830826461315,
-0.19530355930328,
-0.1821300983429,
-0.1688042730093,
-0.1553430557251,
-0.14176388084888,
-0.12808467447758,
-0.1143237426877,
-0.10049979388714,
-0.086631961166859,
-0.07273954898119,
-0.058842230588198,
-0.044959861785173,
-0.031112477183342,
-0.017320284619927,
-0.0036034893710166,
0.010017519816756,
0.023522483184934,
0.0368910767138,
0.050103113055229,
0.063138529658318,
0.075977444648743,
0.088600151240826,
0.10098727047443,
0.11311963945627,
0.12497854232788,
0.13654553890228,
0.14780271053314,
0.15873257815838,
0.16931810975075,
0.17954286932945,
0.18939103186131,
0.19884730875492,
0.20789711177349,
0.21652649343014,
0.2247222661972,
0.23247194290161,
0.23976381123066,
0.24658691883087,
0.25293120741844,
0.25878739356995,
0.26414704322815,
0.26900267601013,
0.27334758639336,
0.27717608213425,
0.28048330545425,
0.28326535224915,
0.28551930189133,
0.2872431576252,
0.28843578696251,
0.28909707069397,
0.28922784328461,
0.28882986307144,
0.28790578246117,
0.28645926713943,
0.28449484705925,
0.28201791644096,
0.27903491258621,
0.27555298805237,
0.27158018946648,
0.2671255171299,
0.26219871640205,
0.25681027770042,
0.25097152590752,
0.24469450116158,
0.23799200356007,
0.23087744414806,
0.2233649790287,
0.21546931564808,
0.20720578730106,
0.198590233922,
0.18963903188705,
0.18036906421185,
0.17079761624336,
0.16094236075878,
0.15082134306431,
0.14045289158821,
0.1298556625843,
0.11904852092266,
0.1080504655838,
0.096880681812763,
0.085558459162712,
0.074103124439716,
0.062533989548683,
0.05087036639452,
0.039131484925747,
0.027336431667209,
0.015504171140492,
0.0036533740349114,
-0.0081974389031529,
-0.020030111074448,
-0.031826786696911,
-0.043569970875978,
```

-0.055242612957954,
-0.066828049719334,
-0.078310199081898,
-0.089673385024071,
-0.10090255737305,
-0.11198326200247,
-0.12290159612894,
-0.13364432752132,
-0.14419892430305,
-0.15455354750156,
-0.1646970063448,
-0.17461897432804,
-0.18430978059769,
-0.19376061856747,
-0.20296341180801,
-0.21191097795963,
-0.22059689462185,
-0.22901557385921,
-0.23716233670712,
-0.24503330886364,
-0.25262543559074,
-0.25993660092354,
-0.266965508461,
-0.27371174097061,
-0.28017562627792,
-0.28635847568512,
-0.29226237535477,
-0.29789021611214,
-0.30324575304985,
-0.30833348631859,
-0.31315872073174,
-0.31772747635841,
-0.3220466375351,
-0.3261236846447,
-0.3299668431282,
-0.33358496427536,
-0.33698761463165,
-0.34018492698669,
-0.34318763017654,
-0.34600695967674,
-0.34865471720695,
-0.35114318132401,
-0.35348510742188,
-0.35569351911545,
-0.35778197646141,
-0.35976427793503,
-0.36165457963943,
-0.36346724629402,
-0.36521676182747,
-0.36691799759865,
-0.36858579516411,
-0.37023508548737,
-0.3718808889389,
-0.37353822588921,
-0.37522199749947,
-0.37694722414017,
-0.37872847914696,
-0.38058042526245,
-0.38251745700836,
-0.38455367088318,
-0.38670289516449,
-0.38897851109505,
-0.39139384031296,
-0.39396148920059,
-0.39669373631477,
-0.39960232377052,
-0.40269863605499,
-0.40599328279495,
-0.40949648618698,
-0.41321769356728,
-0.41716584563255,
-0.42134919762611,
-0.42577520012856,
-0.43045064806938,
-0.43538174033165,
-0.44057357311249,
-0.44603088498116,
-0.45175719261169,
-0.45775556564331,
-0.46402806043625,
-0.4705758690834,
-0.47739943861961,
-0.48449835181236,
-0.49187135696411,
-0.49951612949371,
-0.5074297785759,
-0.51560854911804,
-0.52404749393463,
-0.53274118900299,
-0.54168319702148,
-0.55086636543274,
-0.56028252840042,
-0.56992292404175,
-0.57977795600891,
-0.58983719348907,
-0.60008949041367,
-0.61052304506302,
-0.62112528085709,
-0.63188296556473,
-0.64278227090836,
-0.65380877256393,
-0.66494715213776,
-0.67618209123611,
-0.68749731779099,
-0.69887614250183,
-0.71030151844025,
-0.72175621986389,
-0.73322206735611,
-0.74468106031418,
-0.75611472129822,
-0.76750409603119,
-0.77883064746857,
-0.79007506370544,
-0.80121809244156,
-0.81224048137665,
-0.82312297821045,
-0.83384615182877,
-0.84439069032669,
-0.85473757982254,
-0.86486780643463,
-0.87476223707199,
-0.88440251350403,
-0.89377027750015,
-0.902847468853,
-0.91161620616913,
-0.92005944252014,
-0.92816025018692,
-0.9359022974968,
-0.9432697892189,
-0.95024710893631,
-0.95681971311569,
-0.96297371387482,
-0.96869534254074,
-0.97397196292877,
-0.97879153490067,
-0.98314279317856,
-0.98701506853104,
-0.99039870500565,
-0.99328488111496,
-0.99566543102264,
-0.99753302335739,
-0.99888151884079,
-0.99970531463623,
-1.0,
-0.99976193904877,
-0.99898833036423,
-0.99767762422562,
-0.99582886695862,
-0.99344229698181,
-0.99051880836487,
-0.98706072568893,
-0.98307079076767,
-0.97855287790298,

-0.9735119342804,
-0.9679536819458,
-0.96188473701477,
-0.95531260967255,
-0.94824576377869,
-0.94069355726242,
-0.93266588449478,
-0.92417395114899,
-0.91522926092148,
-0.90584444999695,
-0.8960325717926,
-0.88580769300461,
-0.87518441677094,
-0.864177942276,
-0.85280412435532,
-0.84107953310013,
-0.82902103662491,
-0.81664615869522,
-0.80397301912308,
-0.79101973772049,
-0.7778052687645,
-0.76434874534607,
-0.75066959857941,
-0.73678737878799,
-0.72272211313248,
-0.70849376916885,
-0.69412273168564,
-0.67962914705276,
-0.66503340005875,
-0.65035575628281,
-0.63561654090881,
-0.62083595991135,
-0.60603392124176,
-0.5912304520607,
-0.57644528150558,
-0.56169754266739,
-0.54700642824173,
-0.53239071369171,
-0.51786869764328,
-0.50345832109451,
-0.48917698860168,
-0.47504165768623,
-0.46106880903244,
-0.44727426767349,
-0.43367326259613,
-0.42028042674065,
-0.40710970759392,
-0.39417430758476,
-0.38148686289787,
-0.36905914545059,
-0.35690221190453,
-0.34502628445625,
-0.33344087004662,
-0.32215458154678,
-0.31117522716522,
-0.30050978064537,
-0.290164321661,
-0.28014406561852,
-0.27045339345932,
-0.26109582185745,
-0.25207397341728,
-0.24338954687119,
-0.23504345118999,
-0.22703558206558,
-0.21936510503292,
-0.2120303362608,
-0.20502862334251,
-0.19835659861565,
-0.19201000034809,
-0.18598380684853,
-0.18027217686176,
-0.17486856877804,
-0.16976563632488,
-0.16495536267757,
-0.16042898595333,
-0.1561771184206,
-0.15218977630138,
-0.14845629036427,
-0.14496546983719,
-0.14170554280281,
-0.1386643499136,
-0.13582907617092,
-0.13318663835526,
-0.13072349131107,
-0.12842574715614,
-0.12627917528152,
-0.12426933646202,
-0.12238148599863,
-0.12060075998306,
-0.11891203373671,
-0.11730024218559,
-0.11575008928776,
-0.11424635350704,
-0.11277382820845,
-0.11131732910872,
-0.10986179858446,
-0.10839232802391,
-0.10689423233271,
-0.10535298287868,
-0.10375439375639,
-0.1020845323801,
-0.1003298535943,
-0.098477192223072,
-0.096513792872429,
-0.094427414238453,
-0.092206202447414,
-0.089838951826096,
-0.08731497079134,
-0.084624134004116,
-0.081756964325905,
-0.078704632818699,
-0.075458973646164,
-0.072012551128864,
-0.06835862249136,
-0.06449119001627,
-0.060405045747757,
-0.056095737963915,
-0.051559589803219,
-0.046793770045042,
-0.041796244680882,
-0.036565840244293,
-0.031102120876312,
-0.025405636057258,
-0.01947764120996,
-0.013320285826921,
-0.0069366036914289,
-0.00033040496055037,
0.0064936331473291,
0.013530095107853,
0.02077260427177,
0.028214126825333,
0.035846725106239,
0.043661817908287,
0.051649946719408,
0.059800986200571,
0.068104140460491,
0.07654781639576,
0.085119903087616,
0.093807503581047,
0.1025971993804,
0.11147501319647,
0.12042640894651,
0.12943632900715,
0.13848921656609,
0.14756916463375,
0.15665973722935,
0.16574428975582,
0.17480567097664,
0.18382661044598,
0.19278953969479,
0.20167665183544,
0.21046997606754,
0.21915152668953,

```
0.22770312428474,
0.23610661923885,
0.24434389173985,
0.25239679217339,
0.26024740934372,
0.26787787675858,
0.27527049183846,
0.28240787982941,
0.28927284479141,
0.29584860801697,
0.30211862921715,
0.30806687474251,
0.31367763876915,
0.31893581151962,
0.32382667064667,
0.32833614945412,
0.33245071768761,
0.33615747094154,
0.33944410085678,
0.34229910373688,
0.34471166133881,
0.34667149186134,
0.34816941618919,
0.34919679164886,
0.3497459590435,
0.34980991482735,
0.3493826687336,
0.34845906496048,
0.3470347225666,
0.34510633349419,
0.34267148375511,
0.339728474617,
0.33627679944038,
0.3323167860508,
0.32784956693649,
0.32287740707397,
0.31740343570709,
0.31143167614937,
0.30496707558632,
0.29801559448242,
0.29058402776718,
0.28268003463745,
0.27431216835976,
0.26548993587494,
0.25622355937958,
0.24652419984341,
0.23640371859074,
0.22587485611439,
0.21495102345943,
0.20364640653133,
0.1919758617878,
0.17995493113995,
0.16759975254536,
0.15492710471153,
0.14195430278778,
0.12869922816753,
0.11518017202616,
0.1014159694314,
0.087425790727139,
0.07322920113802,
0.05884613096714,
0.044296752661467,
0.029601499438286,
0.014780982397497,
-0.00014402809028979,
-0.015152608044446,
-0.030223812907934,
-0.045336656272411,
-0.060470212250948,
-0.075603537261486,
-0.090715885162354,
-0.10578664392233,
-0.12079536169767,
-0.13572186231613,
-0.15054625272751,
-0.16524897515774,
-0.17981079220772,
-0.19421294331551,
-0.20843705534935,
-0.22246526181698,
-0.23628023266792,
-0.24986518919468,
-0.26320388913155,
-0.27628082036972,
-0.28908097743988,
-0.30159020423889,
-0.31379494071007,
-0.32568237185478,
-0.33724054694176,
-0.34845814108849,
-0.35932475328445,
-0.36983081698418,
-0.3799674808979,
-0.38972693681717,
-0.3991020321846,
-0.40808668732643,
-0.41667556762695,
-0.42486426234245,
-0.43264934420586,
-0.44002825021744,
-0.4469992518425,
-0.45356151461601,
-0.45971521735191,
-0.46546134352684,
-0.47080174088478,
-0.47573918104172,
-0.48027721047401,
-0.48442032933235,
-0.48817366361618,
-0.49154350161552,
-0.49453645944595,
-0.49716034531593,
-0.49942344427109,
-0.50133472681046,
-0.50290405750275,
-0.50414174795151,
-0.50505894422531,
-0.50566720962524,
-0.5059786438942,
-0.50600612163544,
-0.50576263666153,
-0.50526201725006,
-0.50451821088791,
-0.5035457611084,
-0.5023592710495,
-0.50097393989563,
-0.49940505623817,
-0.49766811728477,
-0.49577876925468,
-0.49375295639038,
-0.49160653352737,
-0.4893553853035,
-0.48701551556587,
-0.48460277915001,
-0.4821330010891,
-0.47962185740471,
-0.47708484530449,
-0.47453734278679,
-0.47199428081512,
-0.46947038173676,
-0.46698021888733,
-0.46453762054443,
-0.46215641498566,
-0.4598496556282,
-0.45763003826141,
-0.45550984144211,
-0.4535006582737,
-0.45161357522011,
-0.44985899329185,
-0.44824674725533,
-0.44678601622581,
-0.44548526406288,
-0.44435226917267,
```

```
-0.44339397549629,
-0.44261679053307,
-0.44202613830566,
-0.44162672758102,
-0.44142258167267,
-0.4414167702198,
-0.44161152839661,
-0.44200846552849,
-0.4426081776619,
-0.44341045618057,
-0.44441431760788,
-0.4456180036068,
-0.44701865315437,
-0.4486129283905,
-0.45039647817612,
-0.45236411690712,
-0.45451006293297,
-0.45682752132416,
-0.45930901169777,
-0.46194648742676,
-0.4647308588028,
-0.46765258908272,
-0.47070133686066,
-0.47386613488197,
-0.47713533043861,
-0.48049682378769,
-0.48393771052361,
-0.48744475841522,
-0.49100413918495,
-0.49460151791573,
-0.49822217226028,
-0.5018510222435,
-0.50547248125076,
-0.50907069444656,
-0.5126296877861,
-0.51613312959671,
-0.51956427097321,
-0.5229064822197,
-0.526143014431,
-0.52925688028336,
-0.532231092453,
-0.53504878282547,
-0.53769302368164,
-0.54014706611633,
-0.54239422082901,
-0.54441809654236,
-0.54620236158371,
-0.54773110151291,
-0.54898852109909,
-0.54995954036713,
-0.55062907934189,
-0.55098253488541,
-0.55100607872009,
-0.55068600177765,
-0.55000948905945,
-0.54896402359009,
-0.5475378036499,
-0.54571968317032,
-0.54349917173386,
-0.54086649417877,
-0.53781253099442,
-0.53432911634445,
-0.530408680439,
-0.52604442834854,
-0.5212305188179,
-0.51596194505692,
-0.51023441553116,
-0.50404471158981,
-0.49739044904709,
-0.49026989936829,
-0.48268264532089,
-0.47462889552116,
-0.466109842062,
-0.4571276307106,
-0.44768542051315,
-0.43778696656227,
-0.42743736505508,
-0.41664230823517,
-0.40540847182274,
-0.39374351501465,
-0.38165590167046,
-0.3691548705101,
-0.35625064373016,
-0.3429542183876,
-0.32927742600441,
-0.31523281335831,
-0.30083373188972,
-0.28609430789948,
-0.27102929353714,
-0.25565412640572,
-0.23998495936394,
-0.22403852641582,
-0.2078320980072,
-0.1913835555315,
-0.17471121251583,
-0.15783388912678,
-0.14077085256577,
-0.12354175001383,
-0.1061665341258,
-0.088665537536144,
-0.071059301495552,
-0.053368613123894,
-0.035614419728518,
-0.017817823216319
```

Appendix C

Spectral Analysis of Pulsaret

Spectrograms on page 63 displaying process of pulsaret acceleration have been generated with C library FFTW for Discrete Fourier Transform analysis[1]. The analysis parameters included: Audio frames per block = 8 (vertical axis), Window increment = 8 (horizontal axis). The display represents a linear The input pulsaret waveform is presented on Figure C.1 and the gaussian shaped envelope on Figure C.2. The example can be run from within the nuPG Program by selecting preset 'Gauss Accelerando'.

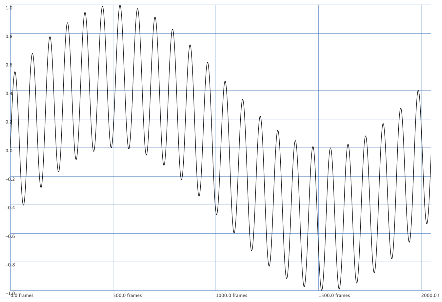


FIGURE C.1: Pulsaret

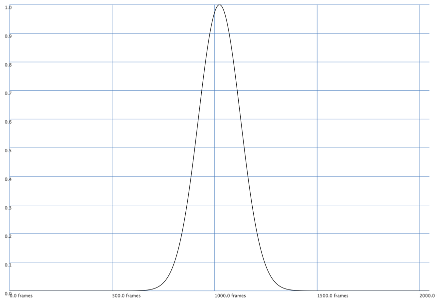


FIGURE C.2: Envelope

---

[1] See `https://www.fftw.org`

Appendix D

Loading the New Pulsar Generator (nuPG) from the script

The New Pulsar Generator (2022) is currently working on Mac OS. Follow the installation steps below:

1. Download SuperCollider 3.12 from: `https://supercollider.github.io/releases/2021/08/02/supercollider-3.12.0`;
2. Install SuperCollider;
3. Locate the Quarks folder; it can be retrieved from within SC IDE by calling Platform.userAppSupportDir. It is usually: '/Users/UserName/Library/Application Support/SuperCollider';
4. Create 'downloaded-quarks' if it doesn't exist and copy the 'Conductor' folder from the 'Script Files' folder. Ensure you don't download a version of the Conductor from the web. The quark is old and no longer supported; a version included here fixes several problems of the original code;
5. Copy 'miSCellaneous_lib' to the same location - 'downloaded-quarks'
6. Locate the 'Extensions' folder; it should be at the same level as 'downloaded-quarks'; if it is not there, create it;
7. Copy 'NuPG_2021_release' from the 'Script Files' folder
8. Recompile the library, CMD+Shift+L or through the menu on the top of your screen Language and Recompile Class Library
9. Run 'nuPG_2022_startUp.scd'; you should end up with a GUI like the one below (D.1)

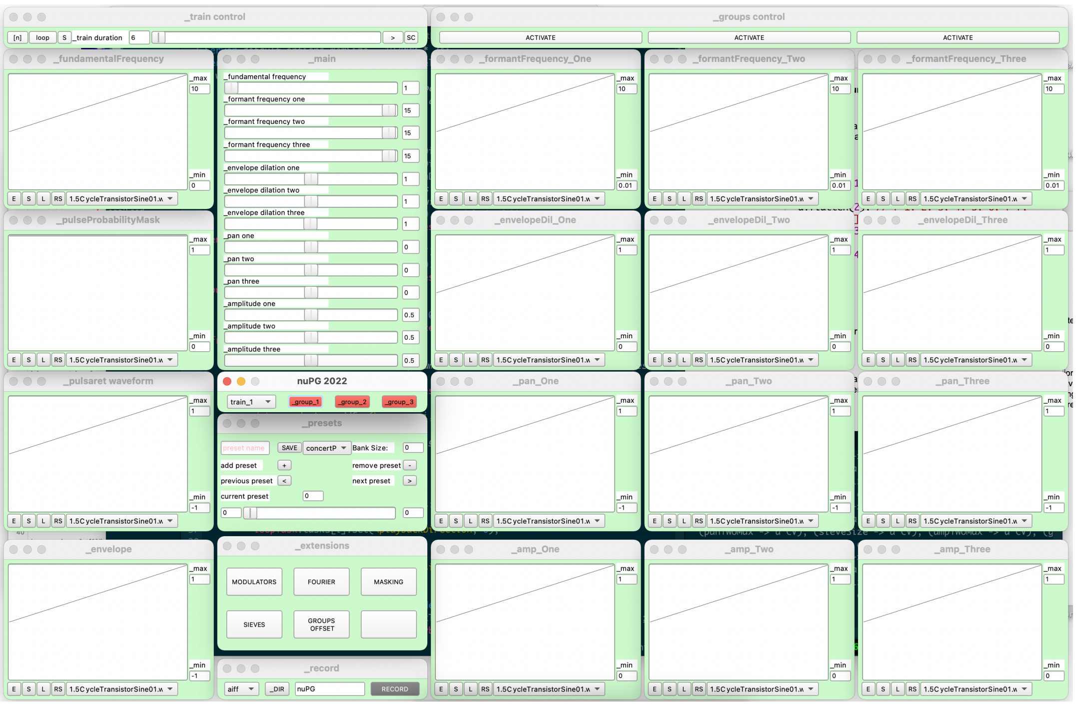


FIGURE D.1: A global view of the New Pulsar Generator (nuPG) standalone program. The majority of parameters are controlled through a table editor. There are sixteen table editors ('fundamental frequency', 'pulse probability masking', 'pulsaret waveform', 'envelope', 'formant frequency one', 'envelope dilation one', 'panning one', 'amplitude one', 'formant frequency two', 'envelope dilation two', 'panning two', 'amplitude two', 'formant frequency three', 'envelope dilation three', 'panning three', 'amplitude three') visible at the startup and three additional ('frequency modulation amount', 'frequency modulation ratio', 'multiparameter modulation') accessible from within MODULATORS extension.

1. To start the program, navigate to the '_train control' window (top-left corner and press the [n] button; this starts the pulsar train;
2. By default program plays a continuous train of pulsars. All tables can be interacted with by drawing, loading a sound file or a dataset from a menu. Each small table view has a large editor (button E);
3. cmd+c copies the content of a table (small and bigger ones) cmd+p pastes it;
4. 'AS TABLE' output of the sieve operation automatically copies the result into the memory; one can paste cmd+p it into any table-based object. Vary size of sieve operation for results of different resolutions. The result is always resampled to 2048;
5. A pulsar stream can be further processed using JiTLib functionality; see 'nupg_fx.scd' in the 'Script Files' repository;

Code 13. The New Pulsar Generator (nuPG) startup script

```
(
Server.default.options.memSize = 192000 * 24;
Server.default.options.numOutputBusChannels = 16;
```

```
s.waitForBoot({
  //settable number of instances = number of pulsar streams
  //this needs to be set at build time and cannot be changed after
  ~numberOfInstances = 3;
  //set number of output channels
  ~numChannels = 2;
  ~tablesPath = Platform.userExtensionDir ++ "/nuPG_2021_release/TABLES/";
  ~filesPath =  Platform.userExtensionDir ++ "/nuPG_2021_release/FILES/";
  ~presetsPath = Platform.userExtensionDir ++ "/nuPG_2021_release/PRESETS/";
  //get corresponding number of buffers
  //envelope random data to fill the buffer
  ~envelope = Signal.sineFill(2048, { 0.0.rand }.dup(7));
  ~envelope_buffers = ~numberOfInstances.collect{|i| Buffer.loadCollection(s, ~envelope, 1) };
  //pulsaret waveform random data to fill the buffer
  ~pulsaret = Signal.sineFill(2048, { 0.0.rand }.dup(7));
  ~pulsaret_buffers = ~numberOfInstances.collect{|i| Buffer.loadCollection(s, ~pulsaret, 1) };
  //frequency
  ~freq = Signal.newClear(2048).fill(1.0);
  ~frequency_buffers = ~numberOfInstances.collect{|i| Buffer.loadCollection(s, ~freq, 1) };
  //generate global data structure
  //add local data structures to it, number of local data = number of instances
  ~data = NuPG_Data.new;
  ~data.conductorInit(~numberOfInstances);
  ~numberOfInstances.collect{|i|
    ~data.conductor.addCon(\con_ ++ i, ~data.instanceGeneratorFunction(i));
  };
  //generate equal number of synthesis graphs
  ~synthesis = NuPG_Synthesis.new;
  ~synthesis.trains(~numberOfInstances, numChannels: ~numChannels);
  //map buffers<->data<->synthesis
  ~loopTask = NuPG_LoopTask.new;
  ~loopTask.load(data: ~data, synthesis: ~synthesis, n: ~numberOfInstances);
  ~numberOfInstances.collect{|i|
    ~loopTask.tasks[i].set(\playbackDirection, 0);
  };
  ~scrubbTask = NuPG_ScrubbTask.new;
  ~scrubbTask.load(data: ~data, synthesis: ~synthesis, n: ~numberOfInstances);
  ~numberOfInstances.collect{|i|};

  ~sliderRecordPlaybackTask = NuPG_SliderRecordPlaybackTasks.new;
  ~scrubbArray = ~ sliderRecordPlaybackTask.scrubbArray(n: ~numberOfInstances);
  ~scrubbRecordTask = ~sliderRecordPlaybackTask.scrubbRecordTask(
    data: ~data, array: ~scrubbArray, n: ~numberOfInstances);
  ~scrubbPlaybackTask = ~sliderRecordPlaybackTask.scrubbPlaybackTask(
    data: ~data, array: ~scrubbArray, n: ~numberOfInstances);

  //definitions of al l GUI objects and parameters
  //dimensions of objects, including localisation on screen
  ~guiDefinitions = NuPG_GUI_Definitions;
  //main (intermediary control)
  ~main = NuPG_GUI_Main.new;
```

```
~main.draw("_main", ~guiDefinitions.mainViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~main.data[i] = ~data.data_main[i];
  13.collect{|l|
    ~data.data_main[i][l].connect(~main.slider[i][l]);
    ~data.data_main[i][l].connect(~main.numberDisplay[i][l]);
  };
};
//modulation amount
~modulationTable = NuPG_GUI_Table_View.new;
~modulationTable.defaultTablePath = ~tablesPath;
~modulationTable.draw("_modulation amount", ~guiDefinitions.modulationAmountViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~modulationTable.data[i] = ~data.data_modulationAmount[i];
  ~data.data_modulationAmount[i].connect(~modulationTable.table[i]);
  2.collect{|l|
    ~data.data_modulationAmount_maxMin[i][l].connect(~modulationTable.minMaxValues[i][l])
  };
};
~modulationTable.visible(0);
//modulation table editor
~modulationTableEditor = NuPG_GUI_Table_Editor_View.new;
~modulationTableEditor.defaultTablePath = ~tablesPath;
~modulationTableEditor.draw("_modulation amount editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~modulationTableEditor.data[i] = ~data.data_modulationAmount[i];
  ~data.data_modulationAmount[i].connect(~modulationTableEditor.table[i]);
  2.collect{|l|
    ~data.data_modulationAmount_maxMin[i][l].connect(~modulationTableEditor.minMaxValues[i][l])
  };
};
~modulationTable.editorView = ~modulationTableEditor;

//modulation ratio editors
~modulationRatioTable = NuPG_GUI_Table_View.new;
~modulationRatioTable.defaultTablePath = ~tablesPath;
~modulationRatioTable.draw("_modulation ratio", ~guiDefinitions.modulationRatioViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~modulationRatioTable.data[i] = ~data.data_modulationRatio[i];
  ~data.data_modulationRatio[i].connect(~modulationRatioTable.table[i]);
  2.collect{|l|
    ~data.data_modulationRatio_maxMin[i][l].connect(~modulationRatioTable.minMaxValues[i][l])
  };
};
~modulationRatioTable.visible(0);
//modulation table editor
~modulationRatioEditor = NuPG_GUI_Table_Editor_View.new;
```

```
~modulationRatioEditor.defaultTablePath = ~tablesPath;
~modulationRatioEditor.draw("_modulation ratio editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~modulationRatioEditor.data[i] = ~data.data_modulationRatio[i];
  ~data.data_modulationRatio[i].connect(~modulationRatioEditor.table[i]);
  2.collect{|l|
    ~data.data_modulationRatio_maxMin[i][l].connect(~modulationRatioEditor.minMaxValues[i][l])
  };
};
~modulationRatioTable.editorView = ~modulationRatioEditor;

//multi parameter modulation editor
~multiparameterModulationTable = NuPG_GUI_Table_View.new;
~multiparameterModulationTable.defaultTablePath = ~tablesPath;
~multiparameterModulationTable.draw("_multi parameter modulation", ~guiDefinitions.multiParameterModulationViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~multiparameterModulationTable.data[i] = ~data.data_multiParamModulation[i];
  ~data.data_multiParamModulation[i].connect(~multiparameterModulationTable.table[i]);
  2.collect{|l|
    ~data.data_mulParamModulation_maxMin[i][l].connect(~multiparameterModulationTable.minMaxValues[i][l])
  };
};
~multiparameterModulationTable.visible(0);
//modulation table editor
~multiparameterModulationTableEditor = NuPG_GUI_Table_Editor_View.new;
~multiparameterModulationTableEditor.defaultTablePath = ~tablesPath;
~multiparameterModulationTableEditor.draw("_multi parameter modulation editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~multiparameterModulationTableEditor.data[i] = ~data.data_multiParamModulation[i];
  ~data.data_multiParamModulation[i].connect(~multiparameterModulationTableEditor.table[i]);
  2.collect{|l|
    ~data.data_mulParamModulation_maxMin[i][l].connect(~multiparameterModulationTableEditor.minMaxValues[i][l])
  };
};
~multiparameterModulationTable.editorView = ~multiparameterModulationTableEditor;

//modulators
~modulators = NuPG_GUI_Modulators.new;
~modulators.draw("_modulators", ~guiDefinitions.modulatorsViewDimensions, ~synthesis,  n: ~numberOfInstances);
~modulators.tables = [~multiparameterModulationTable, ~modulationRatioTable, ~modulationTable];
//mapping
~numberOfInstances.collect{|i|
```

```
  //gui <-> data
  ~modulators.data[i] = ~data.data_modulators[i];
  3.collect{|l|
    ~data.data_modulators[i][l].connect(~modulators.slider[i][l]);
    ~data.data_modulators[i][l].connect(~modulators.numberDisplay[i][l]);
  };
};

//groups offset
//modulators
~groupsOffest = NuPG_GUI_GroupsOffset.new;
~groupsOffest.draw("_groupsOffset", ~guiDefinitions.groupsOffsetViewDimensions,  n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~groupsOffest.data[i] = ~data.data_groupsOffset[i];
  3.collect{|l|
    ~data.data_groupsOffset[i][l].connect(~groupsOffest.slider[i][l]);
    ~data.data_groupsOffset[i][l].connect(~groupsOffest.numberDisplay[i][l]);
  };
};

//pulsaret table
~pulsaretTable = NuPG_GUI_Table_View.new;
~pulsaretTable.defaultTablePath = ~tablesPath;
~pulsaretTable.draw("_pulsaret waveform", ~guiDefinitions.pulsaretViewDimensions, buffer: 1, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~pulsaretTable.data[i] = ~data.data_pulsaret[i];
  ~data.data_pulsaret[i].connect(~pulsaretTable.table[i]);
  2.collect{|l|
    ~data.data_pulsaret_maxMin[i][l].connect(~pulsaretTable.minMaxValues[i][l])
  };
  //gui <-> buffers
  ~pulsaretTable.setBuffer[i] = ~pulsaret_buffers[i];
};

//pulsaret table editor
~pulsaretTableEditor = NuPG_GUI_Table_Editor_View.new;
~pulsaretTableEditor.defaultTablePath = ~tablesPath;
~pulsaretTableEditor.draw("_pulsaret editor",
  ~guiDefinitions.tableEditorViewDimensions, buffer: 1, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~pulsaretTableEditor.data[i] = ~data.data_pulsaret[i];
  ~data.data_pulsaret[i].connect(~pulsaretTableEditor.table[i]);
  2.collect{|l|
    ~data.data_pulsaret_maxMin[i][l].connect(~pulsaretTableEditor.minMaxValues[i][l])
  };
  //gui <-> buffers
  ~pulsaretTableEditor.setBuffer[i] = ~pulsaret_buffers[i];
};
~pulsaretTable.editorView = ~pulsaretTableEditor;

```

```
//envelope
~envelopeTable = NuPG_GUI_Table_View.new;
~envelopeTable.defaultTablePath = ~tablesPath;
~envelopeTable.draw("_envelope", ~guiDefinitions.envelopeViewDimensions, buffer: 1, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~envelopeTable.data[i] = ~data.data_envelope[i];
  ~data.data_envelope[i].connect(~envelopeTable.table[i]);
  2.collect{|l|
    ~data.data_envelope_maxMin[i][l].connect(~envelopeTable.minMaxValues[i][l])
  };
  //gui <-> buffers
  ~envelopeTable.setBuffer[i] = ~envelope_buffers[i];
};
//envelope table editor
~envelopeTableEditor = NuPG_GUI_Table_Editor_View.new;
~envelopeTableEditor.defaultTablePath = ~tablesPath;
~envelopeTableEditor.draw("_envelope editor",
  ~guiDefinitions.tableEditorViewDimensions, buffer: 1, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~envelopeTableEditor.data[i] = ~data.data_envelope[i];
  ~data.data_envelope[i].connect(~envelopeTableEditor.table[i]);
  2.collect{|l|
    ~data.data_envelope_maxMin[i][l].connect(~envelopeTableEditor.minMaxValues[i][l])
  };
  //gui <-> buffers
  ~envelopeTableEditor.setBuffer[i] = ~envelope_buffers[i];
};
~envelopeTable.editorView = ~envelopeTableEditor;

//frequency
/*~frequencyTable = NuPG_GUI_Table_View.new;
~frequencyTable.defaultTablePath = ~tablesPath;
~frequencyTable.draw("_frequency", ~guiDefinitions.frequencyViewDimensions, buffer: 1, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
//gui <-> data
~frequencyTable.data[i] = ~data.data_frequency[i];
~data.data_frequency[i].connect(~frequencyTable.table[i]);
2.collect{|l|
~data.data_frequency_maxMin[i][l].connect(~frequencyTable.minMaxValues[i][l])
};
//gui <-> buffers
~frequencyTable.setBuffer[i] = ~frequency_buffers[i];
};*/

//probability masking
~maskingTable = NuPG_GUI_Table_View.new;
~maskingTable.defaultTablePath = ~tablesPath;
~maskingTable.draw("_pulseProbabilityMask", ~guiDefinitions.maskingViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~maskingTable.data[i] = ~data.data_probabilityMask[i];
```

```
  ~data.data_probabilityMask[i].connect(~maskingTable.table[i]);
  2.collect{|l|
    ~data.data_probabilityMask_maxMin[i][l].connect(~maskingTable.minMaxValues[i][l])
  };
};
//probability table editor
~probabilityTableEditor = NuPG_GUI_Table_Editor_View.new;
~probabilityTableEditor.defaultTablePath = ~tablesPath;
~probabilityTableEditor.draw("_pulseProbabilityMask editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~probabilityTableEditor.data[i] = ~data.data_probabilityMask[i];
  ~data.data_probabilityMask[i].connect(~probabilityTableEditor.table[i]);
  2.collect{|l|
    ~data.data_probabilityMask_maxMin[i][l].connect(~probabilityTableEditor.minMaxValues[i][l])
  };
};
~maskingTable.editorView = ~probabilityTableEditor;
//fundamental frequency
~fundamentalTable = NuPG_GUI_Table_View.new;
~fundamentalTable.defaultTablePath = ~tablesPath;
~fundamentalTable.draw("_fundamentalFrequency", ~guiDefinitions.fundamentalViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~fundamentalTable.data[i] = ~data.data_fundamentalFrequency[i];
  ~data.data_fundamentalFrequency[i].connect(~fundamentalTable.table[i]);
  2.collect{|l|
    ~data.data_fundamentalFrequency_maxMin[i][l].connect(~fundamentalTable.minMaxValues[i][l])
  };
};
~fundamentalTable.pattern[0] = ~fundPatt;

//fundamental table editor
~fundamentalTableEditor = NuPG_GUI_Table_Editor_View.new;
~fundamentalTableEditor.defaultTablePath = ~tablesPath;
~fundamentalTableEditor.draw("_fundamentalFrequency editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~fundamentalTableEditor.data[i] = ~data.data_fundamentalFrequency[i];
  ~data.data_fundamentalFrequency[i].connect(~fundamentalTableEditor.table[i]);
  2.collect{|l|
    ~data.data_fundamentalFrequency_maxMin[i][l].connect(~fundamentalTableEditor.minMaxValues[i][l])
  };
};
~fundamentalTable.editorView = ~fundamentalTableEditor;
~fundamentalTableEditor.parentView = ~fundamentalTable;

//formant frequency one
~formantOneTable = NuPG_GUI_Table_View.new;
~formantOneTable.defaultTablePath = ~tablesPath;
```

```
~formantOneTable.draw("_formantFrequency_One", ~guiDefinitions.formantOneViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~formantOneTable.data[i] = ~data.data_formantFrequencyOne[i];
  ~data.data_formantFrequencyOne[i].connect(~formantOneTable.table[i]);
  2.collect{|l|
    ~data.data_formantFrequencyOne_maxMin[i][l].connect(~formantOneTable.minMaxValues[i][l])
  };
};
//formant one table editor
~formantOneTableEditor = NuPG_GUI_Table_Editor_View.new;
~formantOneTableEditor.defaultTablePath = ~tablesPath;
~formantOneTableEditor.draw("_formantFrequency_One editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~formantOneTableEditor.data[i] = ~data.data_formantFrequencyOne[i];
  ~data.data_formantFrequencyOne[i].connect(~formantOneTableEditor.table[i]);
  2.collect{|l|
    ~data.data_formantFrequencyOne_maxMin[i][l].connect(~formantOneTableEditor.minMaxValues[i][l])
  };
};
~formantOneTable.editorView = ~formantOneTableEditor;
~formantOneTableEditor.parentView = ~formantOneTable;

//formant frequency two
~formantTwoTable = NuPG_GUI_Table_View.new;
~formantTwoTable.defaultTablePath = ~tablesPath;
~formantTwoTable.draw("_formantFrequency_Two", ~guiDefinitions.formantTwoViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~formantTwoTable.data[i] = ~data.data_formantFrequencyTwo[i];
  ~data.data_formantFrequencyTwo[i].connect(~formantTwoTable.table[i]);
  2.collect{|l|
    ~data.data_formantFrequencyTwo_maxMin[i][l].connect(~formantTwoTable.minMaxValues[i][l])
  };
};
//formant two table editor
~formantTwoTableEditor = NuPG_GUI_Table_Editor_View.new;
~formantTwoTableEditor.defaultTablePath = ~tablesPath;
~formantTwoTableEditor.draw("_formantFrequency_Two editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~formantTwoTableEditor.data[i] = ~data.data_formantFrequencyTwo[i];
  ~data.data_formantFrequencyTwo[i].connect(~formantTwoTableEditor.table[i]);
  2.collect{|l|
    ~data.data_formantFrequencyTwo_maxMin[i][l].connect(~formantTwoTableEditor.minMaxValues[i][l])
  };
};
~formantTwoTable.editorView = ~formantTwoTableEditor;
```

```
~formantTwoTableEditor.parentView = ~formantTwoTable;

//formant frequency three
~formantThreeTable = NuPG_GUI_Table_View.new;
~formantThreeTable.defaultTablePath = ~tablesPath;
~formantThreeTable.draw("_formantFrequency_Three", ~guiDefinitions.formantThreeViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~formantThreeTable.data[i] = ~data.data_formantFrequencyThree[i];
  ~data.data_formantFrequencyThree[i].connect(~formantThreeTable.table[i]);
  2.collect{|l|
    ~data.data_formantFrequencyThree_maxMin[i][l].connect(~formantThreeTable.minMaxValues[i][l])
  };
};
//formant three table editor
~formantThreeTableEditor = NuPG_GUI_Table_Editor_View.new;
~formantThreeTableEditor.defaultTablePath = ~tablesPath;
~formantThreeTableEditor.draw("_formantFrequency_Three editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~formantThreeTableEditor.data[i] = ~data.data_formantFrequencyThree[i];
  ~data.data_formantFrequencyThree[i].connect(~formantThreeTableEditor.table[i]);
  2.collect{|l|
    ~data.data_formantFrequencyThree_maxMin[i][l].connect(~formantThreeTableEditor.minMaxValues[i][l])
  };
};
~formantThreeTable.editorView = ~formantThreeTableEditor;
~formantThreeTableEditor.parentView = ~formantThreeTable;

//envelope multiplication one
~envelopeMult_One = NuPG_GUI_Table_View.new;
~envelopeMult_One.defaultTablePath = ~tablesPath;
~envelopeMult_One.draw("_envelopeDil_One", ~guiDefinitions.envelopeOneViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~envelopeMult_One.data[i] = ~data.data_envelopeMulOne[i];
  ~data.data_envelopeMulOne[i].connect(~envelopeMult_One.table[i]);
  2.collect{|l|
    ~data.data_envelopeMulOne_maxMin[i][l].connect(~envelopeMult_One.minMaxValues[i][l])
  };
};
//envelope multiplication one table editor
~envelopeMult_One_Editor = NuPG_GUI_Table_Editor_View.new;
~envelopeMult_One_Editor.defaultTablePath = ~tablesPath;
~envelopeMult_One_Editor.draw("_envelopeDil_One editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~envelopeMult_One_Editor.data[i] = ~data.data_envelopeMulOne[i];
  ~data.data_envelopeMulOne[i].connect(~envelopeMult_One_Editor.table[i]);
  2.collect{|l|
```

```
    ~data.data_envelopeMulOne_maxMin[i][l].connect(~
 envelopeMult_One_Editor.minMaxValues[i][l])
  };
};
~envelopeMult_One.editorView = ~envelopeMult_One_Editor;
~envelopeMult_One_Editor.parentView = ~envelopeMult_One;

//envelope multiplication two
~envelopeMult_Two = NuPG_GUI_Table_View.new;
~envelopeMult_Two.defaultTablePath = ~tablesPath;
~envelopeMult_Two.draw("_envelopeDil_Two", ~guiDefinitions.
 envelopeTwoViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~envelopeMult_Two.data[i] = ~data.data_envelopeMulTwo[i];
  ~data.data_envelopeMulTwo[i].connect(~envelopeMult_Two.table[i]);
  2.collect{|l|
    ~data.data_envelopeMulTwo_maxMin[i][l].connect(~envelopeMult_Two.
 minMaxValues[i][l])
  };
};
//envelope multiplication two table editor
~envelopeMult_Two_Editor = NuPG_GUI_Table_Editor_View.new;
~envelopeMult_Two_Editor.defaultTablePath = ~tablesPath;
~envelopeMult_Two_Editor.draw("_envelopeDil_Two editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~envelopeMult_Two_Editor.data[i] = ~data.data_envelopeMulTwo[i];
  ~data.data_envelopeMulTwo[i].connect(~envelopeMult_Two_Editor.table
 [i]);
  2.collect{|l|
    ~data.data_envelopeMulTwo_maxMin[i][l].connect(~
 envelopeMult_Two_Editor.minMaxValues[i][l])
  };
};
~envelopeMult_Two.editorView = ~envelopeMult_Two_Editor;
~envelopeMult_Two_Editor.parentView = ~envelopeMult_Two;

//envelope multiplication three
~envelopeMult_Three = NuPG_GUI_Table_View.new;
~envelopeMult_Three.defaultTablePath = ~tablesPath;
~envelopeMult_Three.draw("_envelopeDil_Three", ~guiDefinitions.
 envelopeThreeViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~envelopeMult_Three.data[i] = ~data.data_envelopeMulThree[i];
  ~data.data_envelopeMulThree[i].connect(~envelopeMult_Three.table[i
 ]);
  2.collect{|l|
    ~data.data_envelopeMulThree_maxMin[i][l].connect(~
 envelopeMult_Three.minMaxValues[i][l])
  };
};
//envelope multiplication three table editor
~envelopeMult_Three_Editor = NuPG_GUI_Table_Editor_View.new;
~envelopeMult_Three_Editor.defaultTablePath = ~tablesPath;
~envelopeMult_Three_Editor.draw("_envelopeDil_Three editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
```

```
 ~envelopeMult_Three_Editor.data[i] = ~data.data_envelopeMulThree[i];
 ~data.data_envelopeMulThree[i].connect(~envelopeMult_Three_Editor.table[i]);
 2.collect{|l|
  ~data.data_envelopeMulThree_maxMin[i][l].connect(~envelopeMult_Three_Editor.minMaxValues[i][l])
 };
};
~envelopeMult_Three.editorView = ~envelopeMult_Three_Editor;
~envelopeMult_Three_Editor.parentView = ~envelopeMult_Three;


//pan one
~panOneTable = NuPG_GUI_Table_View.new;
~panOneTable.defaultTablePath = ~tablesPath;
~panOneTable.draw("_pan_One", ~guiDefinitions.panOneViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
 //gui <-> data
 ~panOneTable.data[i] = ~data.data_panOne[i];
 ~data.data_panOne[i].connect(~panOneTable.table[i]);
 2.collect{|l|
  ~data.data_panOne_maxMin[i][l].connect(~panOneTable.minMaxValues[i][l])
 };
};
//pan one table editor
~panOneTable_Editor = NuPG_GUI_Table_Editor_View.new;
~panOneTable_Editor.defaultTablePath = ~tablesPath;
~panOneTable_Editor.draw("_pan_One editor",
 ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
 //gui <-> data
 ~panOneTable_Editor.data[i] = ~data.data_panOne[i];
 ~data.data_panOne[i].connect(~panOneTable_Editor.table[i]);
 2.collect{|l|
  ~data.data_panOne_maxMin[i][l].connect(~panOneTable_Editor.minMaxValues[i][l])
 };
};
~panOneTable.editorView = ~panOneTable_Editor;
~panOneTable_Editor.parentView = ~panOneTable;

//pan two
~panTwoTable = NuPG_GUI_Table_View.new;
~panTwoTable. defaultTablePath = ~tablesPath;
~panTwoTable.draw("_pan_Two", ~guiDefinitions.panTwoViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
 //gui <-> data
 ~panTwoTable.data[i] = ~data.data_panTwo[i];
 ~data.data_panTwo[i].connect(~panTwoTable.table[i]);
 2.collect{|l|
  ~data.data_panTwo_maxMin[i][l].connect(~panTwoTable.minMaxValues[i][l])
 };
};
//pan two table editor
~panTwoTable_Editor = NuPG_GUI_Table_Editor_View.new;
~panTwoTable_Editor.defaultTablePath = ~tablesPath;
```

```
~panTwoTable_Editor.draw("_pan_Two editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~panTwoTable_Editor.data[i] = ~data.data_panTwo[i];
  ~data.data_panTwo[i].connect(~panTwoTable_Editor.table[i]);
  2.collect{|l|
    ~data.data_panTwo_maxMin[i][l].connect(~panTwoTable_Editor.
 minMaxValues[i][l])
  };
};
~panTwoTable.editorView = ~panTwoTable_Editor;
~panTwoTable_Editor.parentView = ~panTwoTable;

//pan three
~panThreeTable = NuPG_GUI_Table_View.new;
~panThreeTable.defaultTablePath = ~tablesPath;
~panThreeTable.draw("_pan_Three", ~guiDefinitions.
 panThreeViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~panThreeTable.data[i] = ~data.data_panThree[i];
  ~data.data_panThree[i].connect(~panThreeTable.table[i]);
  2.collect{|l|
    ~data.data_panThree_maxMin[i][l].connect(~panThreeTable.
 minMaxValues[i][l])
  };
};
//pan three table editor
~panThreeTable_Editor = NuPG_GUI_Table_Editor_View.new;
~panThreeTable_Editor.defaultTablePath = ~tablesPath;
~panThreeTable_Editor.draw("_pan_Three editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~panThreeTable_Editor.data[i] = ~data.data_panThree[i];
  ~data.data_panThree[i].connect(~panThreeTable_Editor.table[i]);
  2.collect{|l|
    ~data.data_panThree_maxMin[i][l].connect(~panThreeTable_Editor.
 minMaxValues[i][l])
  };
};
~panThreeTable.editorView = ~panThreeTable_Editor;
~panThreeTable_Editor.parentView = ~panThreeTable;

//amp one
~ampOneTable = NuPG_GUI_Table_View.new;
~ampOneTable.defaultTablePath = ~tablesPath;
~ampOneTable.draw("_amp_One", ~guiDefinitions.ampOneViewDimensions, n
  : ~numberOfInstances);
//mapp\ing
~numberOfInstances.collect{|i|
  //gui <-> data
  ~ampOneTable.data[i] = ~data.data_ampOne[i];
  ~data.data_ampOne[i].connect(~ampOneTable.table[i]);
  2.collect{|l|
    ~data.data_ampOne_maxMin[i][l].connect(~ampOneTable.minMaxValues[
 i][l])
  };
};
//amp one table editor
~ampOneTable_Editor = NuPG_GUI_Table_Editor_View.new;
```

```
~ampOneTable_Editor.defaultTablePath = ~tablesPath;
~ampOneTable_Editor.draw("_pan_Three editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~ampOneTable_Editor.data[i] = ~data.data_ampOne[i];
  ~data.data_ampOne[i].connect(~ampOneTable_Editor.table[i]);
  2.collect{|l|
    ~data.data_ampOne_maxMin[i][l].connect(~ampOneTable_Editor.minMaxValues[i][l])
  };
};
~ampOneTable.editorView = ~ampOneTable_Editor;
~ampOneTable_Editor.parentView = ~ampOneTable;

//amp two
~ampTwoTable = NuPG_GUI_Table_View.new;
~ampTwoTable.defaultTablePath = ~tablesPath;
~ampTwoTable.draw("_amp_Two", ~guiDefinitions.ampTwoViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~ampTwoTable.data[i] = ~data.data_ampTwo[i];
  ~data.data_ampTwo[i].connect(~ampTwoTable.table[i]);
  2.collect{|l|
    ~data.data_ampTwo_maxMin[i][l].connect(~ampTwoTable.minMaxValues[i][l])
  };
};
//amp two table editor
~ampTwoTable_Editor = NuPG_GUI_Table_Editor_View.new;
~ampTwoTable_Editor.defaultTablePath = ~tablesPath;
~ampTwoTable_Editor.draw("_pan_Three editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~ampTwoTable_Editor.data[i] = ~data.data_ampTwo[i];
  ~data.data_ampTwo[i].connect(~ampTwoTable_Editor.table[i]);
  2.collect{|l|
    ~data.data_ampTwo_maxMin[i][l].connect(~ampTwoTable_Editor.minMaxValues[i][l])
  };
};
~ampTwoTable.editorView = ~ampTwoTable_Editor;
~ampTwoTable_Editor.parentView = ~ampTwoTable;

//amp three
~ampThreeTable = NuPG_GUI_Table_View.new;
~ampThreeTable.defaultTablePath = ~tablesPath;
~ampThreeTable.draw("_amp_Three", ~guiDefinitions.ampThreeViewDimensions, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  //gui <-> data
  ~ampThreeTable.data[i] = ~data.data_ampThree[i];
  ~data.data_ampThree[i].connect(~ampThreeTable.table[i]);
  2.collect{|l|
    ~data.data_ampThree_maxMin[i][l].connect(~ampThreeTable.minMaxValues[i][l])
  };
};
//amp two table editor
```

```
~ampThreeTable_Editor = NuPG_GUI_Table_Editor_View.new;
~ampThreeTable_Editor.defaultTablePath = ~tablesPath;
~ampThreeTable_Editor.draw("_pan_Three editor",
  ~guiDefinitions.tableEditorViewDimensions, n: ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~ampThreeTable_Editor.data[i] = ~data.data_ampThree[i];
  ~data.data_ampThree[i].connect(~ampThreeTable_Editor.table[i]);
  2.collect{|l|
    ~data.data_ampThree_maxMin[i][l].connect(~ampThreeTable_Editor.
  minMaxValues[i][l])
  };
};
~ampThreeTable.editorView = ~ampThreeTable_Editor;
~ampThreeTable_Editor.parentView = ~ampThreeTable;



//amp three
~record = NuPG_GUI_Record_View.new;
~record.draw(~guiDefinitions.recordViewDimensions, n: ~
  numberOfInstances);


//groups control
~groupsControl = NuPG_GUI_GroupControl_View.new;
~groupsControl.draw(~guiDefinitions.groupsControlViewDimensions, ~
  synthesis, n: ~numberOfInstances);

//scrubber
~scrubber = NuPG_GUI_ScrubberView.new;
~scrubber.draw(~guiDefinitions.scrubberViewDimensions, data: ~data,
  tasks: ~scrubbTask, synthesis: ~synthesis, n: ~numberOfInstances);
~scrubber.path = ~filesPath;
~numberOfInstances.collect{|i|
  ~data.data_scrubber[i].connect(~scrubber.trainProgress[i]);
  ~data.data_scrubber[i].connect(~scrubber.progresDisplay[i]);
};
~scrubber.sliderRecordTask = ~scrubbRecordTask;
~scrubber.sliderPlaybackTask = ~scrubbPlaybackTask;
//train control
~trainControl = NuPG_GUI_TrainControl_View.new;
~trainControl.draw(~guiDefinitions.trainControlViewDimensions, ~
  loopTask, ~scrubber, ~synthesis, ~progressSlider, n: ~
  numberOfInstances);
//progress slider tasks
~progressSlider = NuPG_ProgressSliderPlay.new;
~progressSlider.load(~data, ~trainControl, n: ~numberOfInstances);
//mapping
~numberOfInstances.collect{|i|
  ~data.data_trainDuration[i].connect(~trainControl.trainDuration[i])
  ;
  ~trainControl.scrubbTask[i] = ~scrubbTask.tasks[i];
  ~trainControl.progresTask[i] = ~progressSlider.tasks[i];
  ~progressSlider.tasks[i].set(\progressDirection, 0);
};


//presets
~presets = NuPG_GUI_Presets_View.new;
~presets.defaultPresetPath = ~presetsPath;
~presets.draw("_presets", ~guiDefinitions.presetsViewDimensions, n: ~
  numberOfInstances);
```

```
~presets.data = ~data;
~numberOfInstances.collect{|i|
  ~data.conductor[(\con_ ++ i).asSymbol].preset.presetCV.connect(~
  presets.currentPreset[i]);
  ~data.conductor[(\con_ ++ i).asSymbol].preset.presetCV.connect(~
  presets.interpolationFromPreset[i]);
  ~data.conductor[(\con_ ++ i).asSymbol].preset.targetCV.connect(~
  presets.interpolationToPreset[i]);
  ~data.conductor[(\con_ ++ i).asSymbol]
  .preset.interpCV.connect(~presets.presetInterpolationSlider[i]);
  ~presets.pulsaretBuffers[i] = ~pulsaret_buffers[i];
  ~presets.envelopeBuffers[i] = ~envelope_buffers[i];

  ~presets.interpolationFromPreset[i].keyDownAction_({arg view,char,
  modifiers,unicode,keycode;
    if(keycode == 36,
      {
        ~pulsaret_buffers[i].sendCollection(~data.data_pulsaret[i].
  value);
        ~envelope_buffers[i].sendCollection(~data.data_envelope[i].
  value);

      },
      {});
    if(keycode == 76,
      {   ~pulsaret_buffers[i].sendCollection(~data.data_pulsaret[i].
  value);
        ~envelope_buffers[i].sendCollection(~data.data_envelope[i].
  value);

      },
      {});
  });
  ~presets.interpolationToPreset[i].mouseUpAction_{
    ~pulsaret_buffers[i].sendCollection(~data.data_pulsaret[i].value)
  ;
        ~envelope_buffers[i].sendCollection(~data.data_envelope[i].
  value);
  };
};

//fourier
~fourier = NuPG_GUI_Fourier_View.new;
~fourier.draw("_fourier", ~guiDefinitions.fourierViewDimensions, n: ~
  numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~fourier.data[i] = ~data.data_fourier[i];
  //~data.data_fourier[i].connect(~fourier.table[i]);
};

//masking
~masking = NuPG_GUI_Masking.new;
~masking.draw("_masking", ~guiDefinitions.maskingControlDimensions, n
  : ~numberOfInstances);
~numberOfInstances.collect{|i|
  //gui <-> data
  ~data.data_probabilityMaskSingular[i].connect(~masking.probability[
  i]);
  2.collect{|l|
    ~data.data_burstMask[i][l].connect(~masking.burtsRest[i][l])
  };
  ~data.data_channelMask[i][0].connect(~masking.channel[i][0]);
```

```
  };

  //sieves
  ~sieves = NuPG_GUI_Sieves.new;
  ~sieves.path = ~filesPath;
  ~sieves.draw("_sieves", ~guiDefinitions.sieveViewDimensions,
   synthesis: ~synthesis,  n: ~numberOfInstances);
  ~numberOfInstances.collect{|i|
    2.collect{|l| ~sieves.data[i][l] = ~data.data_sieveMask[i][l] };
  };


  //extensions
  ~extensions = NuPG_GUI_Extensions_View.new;
  ~extensions.draw(~guiDefinitions.extensionsViewDimensions, viewsList:
     [~modulators, ~fourier, ~masking, ~sieves, ~groupsOffest],
    n: ~numberOfInstances);
  //control
  ~control = NuPG_GUI_Control_View.new;
  ~control.draw(~guiDefinitions.controlViewDimensions,
    viewsList: [
      ~pulsaretTable, ~envelopeTable,
      ~main,
      ~maskingTable, ~fundamentalTable,
      ~formantOneTable, ~formantTwoTable, ~formantThreeTable,
      ~envelopeMult_One, ~envelopeMult_Two, ~envelopeMult_Three,
      ~panOneTable, ~panTwoTable, ~panThreeTable,
      ~ampOneTable, ~ampTwoTable, ~ampThreeTable,
      ~groupsControl, ~trainControl,
      ~fourier, ~sieves, ~masking, ~modulators,
      ~pulsaretTableEditor, ~envelopeTableEditor,
      ~probabilityTableEditor, ~fundamentalTableEditor,
      ~formantOneTableEditor, ~formantTwoTableEditor, ~
   formantThreeTableEditor,
      ~envelopeMult_One_Editor, ~envelopeMult_Two_Editor, ~
   envelopeMult_Three_Editor,
      ~panOneTable_Editor, ~panTwoTable_Editor, ~panThreeTable_Editor,
      ~ampOneTable_Editor, ~ampTwoTable_Editor, ~ampThreeTable_Editor,
      ~presets,
      ~modulationTable, ~modulationTableEditor,
      ~modulationRatioTable, ~modulationRatioEditor,
      ~multiparameterModulationTable, ~
   multiparameterModulationTableEditor

    ],
    n: ~numberOfInstances
  );
})
.doWhenBooted({
  ~numberOfInstances.collect{|i|
    ~synthesis.trainInstances[i].set(\pulsaret_buffer, ~
   pulsaret_buffers[i].bufnum);
    ~synthesis.trainInstances[i].set(\envelope_buffer, ~
   envelope_buffers[i].bufnum);
    ~synthesis.trainInstances[i].set(\frequency_buffer, ~
   frequency_buffers[i].bufnum);

    ~synthesis.trainInstances[i].setControls([
      fundamental_frequency: ~data.data_main[i][0],
      formant_frequency_One: ~data.data_main[i][1],
      formant_frequency_Two: ~data.data_main[i][2],
      formant_frequency_Three: ~data.data_main[i][3],
      envMul_One: ~data.data_main[i][4],
```

```
            envMul_Two: ~data.data_main[i][5],
            envMul_Three: ~data.data_main[i][6],
            pan_One: ~data.data_main[i][7],
            pan_Two: ~data.data_main[i][8],
            pan_Three: ~data.data_main[i][9],
            amplitude_One: ~data.data_main[i][10],
            amplitude_Two: ~data.data_main[i][11],
            amplitude_Three: ~data.data_main[i][12],
            fmAmt: ~data.data_modulators[i][0],
            fmRatio: ~data.data_modulators[i][1],
            allFluxAmt: ~data.data_modulators[i][2],
            burst: ~data.data_burstMask[i][0],
            rest: ~data.data_burstMask[i][1],
            chanMask: ~data.data_channelMask[i][0],
            centerMask: ~data.data_channelMask[i][1],
            sieveMod: ~data.data_sieveMask[i][0],
            sieveSequence: ~data.data_sieveMask[i][1],
            probability: ~data.data_probabilityMaskSingular[i],
            offset_1: ~data.data_groupsOffset[i][0],
            offset_2: ~data.data_groupsOffset[i][1],
            offset_3: ~data.data_groupsOffset[i][2]
        ]);
};})
)
```

Appendix E

List of performances, sound installations and other works with the nuPG program

Performances and Installations

- A work commission for Deutschlandradio Kultur/CTM - multichannel work '(dia)grammatology of space' for female voice, synthetic voice and computer premiered at Berghain during CTM/Transmediale Festival in Berlin, February 2016. The work featured a libretto written especially by the Laboria Cuboniks collective. A stereo version was broadcast on Deutschlandradio Kultur as part of Hörspiele/Klangkunst series on the 2nd of September 2016.
- A multichannel performance with visuals by Diann Bauer at the Institute of Contemporary Art (ICA) in London as part of Technology Now series during Frieze 2016. 30 September 2016:
- An artistic residency and commission performance with Tristan Clutterbuck (modular synthesizer) at Irish Museum of Modern Art (IMMA) in Dublin as part of Sunset Birth, IMMA Summer Party curated by Linder and Maxwell Sterling. Dublin, 15 July 2017.
- 'pulsarAutomata' a pluriphonic 12-channel sound installation (with contributions from Curtis Roads, Florian Hecker, Marcus Schmickler and Lorenzo Senni) at West Court, Edinburgh College of Art. February 2019.
- A performance of '/siv/' for synthetic voice and computer-generated pluriphonic sound at Klangdom, ZKM | Center for Art and Media in Karlsruhe. September 2018. The performance took place during a research symposium devoted to the work of a Greek composer Iannis Xenakis.
- A solo exhibition 'The New Pulsar Generator Manual' for 4-channel audio and synthetic voice at Remote Viewing, Philadelphia. October - December 2019.
- A premiere of a new composition 'Pulsar Sieve' on a 45.4-channel sound system at the Royal College of Music, Stockholm. October 2019. The work and the software implementation of Xenakis' sieves developed during an artistic residency at the Hertz-Lab at ZKM | Center for Art and Media Karlsruhe, September - October 2018.
- A new work 'Synthetic Pulsar' commissioned by the CTM Festival premiered virtually as a binaural audio-video installation (January 2021) and as an on-site

pluriphonic installation at Vollgutlager in Berlin (September 2021). The work is a collaboration with vocalist Alex Freiheit, Jocelyn Bell Burnell who had provided an interpretation of astrophysical data, and Birds on Mars AI experts who had designed a WaveNet-based synthetic voice model, especially for the project. The on-site version of the work is made possible with the support of Meyer Sound (Berkley) and utilises their 'spacemap' system as a spatialization method. The work was broadcast as a radio work on Deutschlandradio Kultur as part of Hörspiele/Klangkunst series.

- A sound installation, 'Modal Interlacement. 3x3 Channel' at Centrala Gallery in Birmingham, UK. May - July 2021. The exhibition funded by Arts Council England
- 'Auditory Scene ReSynthesis as Cochlear Wavepackets (2021)' collaborative sound installation with Jan St. Werner (Mouse on Mars) commissioned by Haus der Kulturen der Welt (HKW) in Berlin. Work presented as part of 'The Sound of Distance' festival.
- A live performance of at the Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, April 2022, Stanford, California, USA
- A multichannel live performance at SonicActs Festival, October 2022, Amsterdam.
- Premiere of a new composition 'Fraunhofer Lines' at the Institute for Computer Music and Sound Technology (ICST) of the Zurich University of the Arts (ZHdK), December 2022, Zurich. The work and associated computer programs were produced during the artistic residency at the ICST in June 2022.
- Premiere of a new sound work 'Matrix Partritions (on a clear day)' at OHM Gallery, January 2023, Berlin.

Recordings

- A digital release of a record 'Groups: Articulations of the Real' via Superpang. November 2020.
- A digital release of a record 'Auditory Sieve' with an accompanying essay 'Auditory Sieve: a protocol for pendular transitions between temporal resolutions' on ETAT label. October 2020.
- A CD release 'The New Pulsar Generator Recordings Volume One' on Fancyyyyy Label. October 2020. The release was accompanied by a text 'Sound Composition with Pulsars' by Curtis Roads.

Bibliography

(2020). URL: `https://github.com/dkmayer/miSCellaneous_lib`.

(2021). URL: `https://www.gnu.org/software/gsl/doc/html/dwt.html#`.

A. Piccalli S. Cavaliere, I. Ortosecco (1986). "Synthesis by Formants: A New Approach". In: *Proc. 1986 IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Acoustics, Speech, and Signal Processing Society, New York , 1986)*.

Abadi, Martin and Luca Cardelli (2012). *A theory of objects*. Springer Science & Business Media.

Abelson, Harold and Gerald Jay Sussman (1996). *Structure and interpretation of computer programs*. The MIT Press.

Ablinger, Peter (1997). *Elektronischer Frühling 1992-96*. GEM CD 9701.

— (Dec. 1999). *Quadraturen*. URL: `https://ablinger.mur.at/docu11.html`.

Addison, Paul S (2017). *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance*. CRC press.

Agon, Carlos et al. (2004). "Formal Aspects of Iannis Xenakis'" Symbolic Music": A Computer-Aided Exploration of Compositional Processes". In: *Journal of New Music Research* 33.2, pp. 145–159.

Allouis, Jean François (1982). *The SYTER project: Sound processor design and software overview*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library.

Alt, Casey (2011). "13. Objects of Our Affection: How Object Orientation Made Computers a Medium". In: *Media Archaeology*. University of California Press, pp. 278–301.

Amacher, Maryanne (1999). *Sound characters:(making the third ear)*. Tzadik.

Amiot, Emmanuel (1986). *Duration structure generation and recognition in musical writing*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library.

Andén, Joakim, Vincent Lostanlen, and Stéphane Mallat (2015). "Joint time-frequency scattering for audio classification". In: *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, pp. 1–6.

— (2018). "Classification with joint time-frequency scattering". In: *arXiv preprint arXiv:1807.08869*.

— (2019). "Joint time–frequency scattering". In: *IEEE Transactions on Signal Processing* 67.14, pp. 3704–3718.

Andén, Joakim and Stéphane Mallat (2011). "Multiscale Scattering for Audio Classification." In: *ISMIR*. Miami, FL, pp. 657–662.

Andersen, Christian Ulrik and Søren Bro Pold (2018). *The metainterface: The art of platforms, cities, and clouds*. MIT Press.

Anderson, John B, Tor Aulin, and Carl-Erik Sundberg (2013). *Digital phase modulation*. Springer Science & Business Media.

Ariza, Christopher (2001). "athenaCL Tutorial Manual: Version 2.0. 0a15". In.

— (2004). "An Object-Oriented Model of the Xenakis Sieve for Algorithmic Pitch, Rhythm, and Parameter Generation." In: *ICMC*.

— (2005). "The Xenakis sieve as object: A new model and a complete implementation". In: *Computer Music Journal* 29.2, pp. 40–60.

— (2009). “Sonifying sieves: Synthesis and signal processing applications of the xenakis sieve with python and csound”. In: *ICMC*.

Ashmore, J et al. (2010). “The remarkable cochlear amplifier”. In: *Hearing research* 266.1, pp. 1–17.

Assis, Paulo de (2018). *Logic of Experimentation. Rethinking Music Performance through Artistic Research*. Leuven University Press.

Bachelard, Gaston (1984). “The new scientific spirit”. In.

Badiou, Alain (2014). *Mathematics of the Transcendental*. A&C Black.

— (2019). *Logics of worlds: Being and event II*. Bloomsbury Publishing.

Barthes, Roland (1980). *S/Z: an essay*. Siglo XXI.

— (1981). *Camera lucida: Reflections on photography*. Macmillan.

Bell, Eamonn (2021). “Cybernetics, listening, and sound-studio phenomenotechnique in Abraham Moles’s Théorie de l’information et perception esthétique (1958)”. In: *Resonance: The Journal of Sound and Culture* 2.4, pp. 523–558.

Bernstein, Stephen and Ruth Bernstein (1999). *Schaum’s outline of elements of statistics I: Descriptive statistics and probability*. McGraw Hill Professional.

Bertalanffy, Ludwig von (1969). “General system theory: Foundations, development, applications”. In.

Black, Harold S (1953). *Modulation theory*. van Nostrand.

Bonds, Mark Evan (1991). “Wordless rhetoric: Musical form and the metaphor of the oration”. In.

Bovermann, Till, Julian Rohrhuber, and Alberto de Campo (2011). “Laboratory methods for experimental sonification”. In: *The Sonification Handbook, Berlin: Logos Verlag*, pp. 237–272.

Bradford, Russell, John Ffitch, and Richard Dobson (2008). “Sliding with a constant Q”. In: *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08), Espoo, Finland*, pp. 363–369.

Brandtsegg, Øyvind, Sigurd Saue, and Thom Johansen (2011). “Particle synthesis–a unified model for granular synthesis”. In: *Linux Audio Conference*.

Bresson, Jean and Carlos Agon (2007). “Musical representation of sound in computer-aided composition: A visual programming framework”. In: *Journal of New Music Research* 36.4, pp. 251–266.

Brun, Herbert and Herbert Brün (2004). *When Music Resists Meaning: The Major Writings of Herbert Brün*. Vol. 1. Wesleyan University Press.

Burkholder, J Peter, Donald Jay Grout, and Claude V Palisca (2019). *A History of Western Music: Tenth International Student Edition*. WW Norton & Company.

Burnell, Jocelyn Bell (1984). “The discovery of pulsars”. In: *Serendipitous Discoveries in Radio Astronomy*, p. 160.

Burns, Kristine Helen (1995). “The history and development of algorithms in music composition, 1957-1993.” In.

Castelman, KR (1979). “Digital Image Processing Prentiss Hall: Englewood Cliffs NJ pp 429”. In.

Cavaliere, Sergio and Aldo Piccialli (1997). “Granular synthesis of musical signals”. In: *Musical Signal Processing*, pp. 155–186.

Cochrane, Louis J and Derek Gatherer (2020). “Dynamic programming algorithms applied to musical counterpoint in process composition: an example using Henri Pousseur’s Scambi”. In.

Cooper, Grosvenor W, Grosvenor Cooper, and Leonard B Meyer (1963). *The rhythmic structure of music*. University of Chicago press.

Corbusier, Le (2004). *The Modulor and Modulor 2*. Vol. 1. Springer Science & Business Media.

Cott, Jonathan and Karlheinz Stockhausen (1974). *Stockhausen: conversations with the composer*. Pan.

Cowell, Henry and David Nicholls (1996). *New musical resources*. Cambridge University Press.

Cramer, Florian (2003). *Exe. cut [up] able statements: The Insistence of Code*. na.

Dack, John (2017). "The electroacoustic music of Henri Pousseur and the 'open'form". In: *The modernist legacy: Essays on new music*. Routledge, pp. 177–190.

Dagognet, François (1982). *Faces, surfaces, interfaces*. Vrin.

David, EE, Max V Mathews, and HS McDonald (1958). "Description and results of experiments with speech using digital computer simulation". In: *Proceedings of the 1958 National Electronics Conference*, pp. 766–775.

Waveform Transformation Techniques for Granular Synthesis (1989). 13th International Congress on Acoustics.

De Poli, Giovanni and Aldo Piccialli (1991). "Pitch-synchronous granular synthesis". In: *Representations of musical signals*. MIT Press, pp. 187–219.

De Poli, Giovanni, Aldo Piccialli, and Curtis Roads (1991). *Representations of musical signals*. MIT press.

DeMarinis, Paul (2011). "Erased dots and rotten dashes, or how to wire your head for a preservation". In: *Media Archaeology: Approaches, Applications, and Implications*, pp. 211–238.

Di Scipio, Agostino (1994). "Micro-time sonic design and timbre formation". In: *Contemporary Music Review* 10.2, pp. 135–148.

— (1995). "Inseparable models of materials and of musical design in electroacoustic and computer music". In: *Journal of new music research* 24.1, pp. 34–50.

— (1998). "Compositional models in Xenakis's electroacoustic music". In: *Perspectives of New Music*, pp. 201–243.

— (2003). "'Sound is the interface': from interactive to ecosystemic signal processing". In: *Organised Sound* 8.3, pp. 269–277.

Döbereiner, Luc (2011). "Models of constructed sound: Nonstandard synthesis as an aesthetic perspective". In: *Computer Music Journal* 35.3, pp. 28–39.

Doornbusch, Paul (2004). "Computer sound synthesis in 1951: The music of CSIRAC". In: *Computer Music Journal* 28.1, pp. 10–25.

Drucker, Johanna (2011). "Humanities approaches to graphical display". In: *Digital Humanities Quarterly* 5.1, pp. 1–21.

Eimert, Herbert (1959). *Debussy's Jeux*. Universal.

Eimert, Herbert, Karlheinz Stockhausen, and Hans G Helms (1956). *Electronic music*. Vol. 1. National Research Council of Canada.

Eimert, Herbert, Karlheinz Stockhausen, et al. (1958). *Electronic music*. Tech. rep. T. Presser Company.

Emerson, Lori (2013). "Archives, Materiality and the 'Agency of the Machine': An Interview with Wolfgang Ernst". In: *The Signal: Digital Preservation* 8.

Ernst, Wolfgang (2009). "Experimenting with Media Temporality: Pythagoras, Hertz, Turing."" In: *Digital Memory and the Archive*, pp. 184–91.

— (2011). "11. Media Archaeography: Method and Machine versus History and Narrative of Media". In: *Media archaeology*. University of California Press, pp. 239–255.

— (2013). "From media history to Zeitkritik". In: *Theory, Culture & Society* 30.6, pp. 132–146.

— (2016a). *Chronopoetics: The Temporal Being and Operativity of Technological Media*. Rowman & Littlefield.

— (2016b). *Sonic time machines*. Amsterdam University Press.

— (2018). "The archive as metaphor. From archival space to archival time". In: *Nimio* 5, p. 12.

— (2019). "As Slow as Possible? On the Machinic (Non-) Sense of the Sonic Present and Digital Indifference toward Time". In: *ASAP/Journal* 4.3, pp. 671–688.

Ernst, Wolfgang (2020a). "Existing in Discrete States: On the Techno-Aesthetics of Algorithmic Being-in-Time". In: *Theory, Culture & Society*, p. 0263276420966396.

— (2020b). "The media epistemic value of sonic analytics tools. A commentary". In: *Internet Histories* 5.1, pp. 48–56.

Feichtinger, Hans G and Thomas Strohmer (2012). *Gabor analysis and algorithms: Theory and applications*. Springer Science & Business Media.

Fell, Mark (2022). *Structure and Synthesis: The Anatomy of Practice*. MIT Press.

Feser, Kim, Matthias Pasdzierny, and Timor Kaul (2016). *Techno Studies: Ästhetik und Geschichte Elektronischer Tanzmusik*. B_books.

Fildes, Jonathan (2008). "'Oldest'computer music unveiled". In: *BBC News* 17.

Foucault, Michel (2013a). *Archaeology of knowledge*. Routledge.

— (2013b). *Politics, philosophy, culture: Interviews and other writings, 1977-1984*. Routledge.

Fraser, Lee (2018). *Cor Unvers*. Ge-stell CD.

Fuller, Matthew (2003). "Behind the blip: software as culture". In: *New York: Autonomedia*.

— (2019). "Sonic Tensors and Mental Ears in Universes of Synthesis". In.

Gabor, Dennis (1946). "Theory of communication. Part 1: The analysis of information". In: *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering* 93.26, pp. 429–441.

— (1947). *Acoustical quanta and the theory of hearing*.

Galison, Peter (1988). "History, philosophy, and the central metaphor". In: *Science in Context* 2.1, pp. 197–212.

Galloway, Alexander R (2012). *The interface effect*. Polity.

Gamma, Erich et al. (1993). "Design patterns: Abstraction and reuse of object-oriented design". In: *European Conference on Object-Oriented Programming*. Springer, pp. 406–431.

Gibson, Benoît (2011). *The Instrumental Music of Iannis Xenakis. Theory, Practice, Self-Borrowing*. Pendragon Press.

Gibson, Ross (2010). "The known world". In: *Text* 14.Special 8, pp. 1–11.

Gold, Thomas (1948). "Hearing. II. The physical basis of the action of the cochlea". In: *Proceedings of the Royal Society of London B: Biological Sciences* 135.881, pp. 492–498.

Goldmann, Stefan (2015). *Jeff Mills, Robert Hood and FM Synthesis as a Metaphor*. URL: `https://sgoldmann.wordpress.com/archive/fm-techno/`.

Goldstine, Herman Heine, John Von Neumann, and John Von Neumann (1947). "Planning and coding of problems for an electronic computing instrument". In.

Gonul, Aziz Ege (2016). "SUPERCOLLIDER DX-7 EMULATOR". In.

Gottschalk, Jennie (2016). *Experimental Music Since 1970*. Bloomsbury Publishing USA.

Gottstein, Bjorn (1998). "Misik als Subjekt". The programe notes for 'Quadraturen IV (Selbstportrait mit Berlin' by Peter Ablinger. URL: `https://ablinger.mur.at/docs/Gottstein_Musik-als-Subjekt.pdf`.

Grisey, Gérard (1987). "Tempus ex Machina: A composer's reflections on musical time". In: *Contemporary music review* 2.1, pp. 239–275.

Haar, Alfred (1909). *Zur theorie der orthogonalen funktionensysteme*. Georg-August-Universitat, Gottingen.

Harman, Graham (2013). "On vicarious causation". In.

Haswell, Russell (2011). *ACID nO!se Synthesis*. CD.

Hawkins, David (1958). "Mathematical sieves". In: *Scientific American* 199.6, pp. 105–114.

Haworth, Christopher (2015). "Sound synthesis procedures as texts: An ontological politics in electroacoustic and computer music". In: *Computer Music Journal* 39.1, pp. 41–58.

— (2018). "Protentions and Retentions of Xenakis and Cage: Nonhuman Actors, Genre and Time in Microsound". In: *Contemporary Music Review* 37.5-6, pp. 606–625.

Hecker Florian; Mackay, Robin (2021). "Synthetic Listening". In: *Spectres. Ghosts in the Machine*. Ed. by François Sanson Bartolomé; Bonnet. Vol. 3. Shelter Press.

Heisenberg, Werner (1974). *The uncertainty principle*. Educational Materials and Equipment.

Helmholtz, Hermann (2013). *On the sensations of tone*. Courier Corporation.

Hermann, Thomas, Andy Hunt, and John G Neuhoff (2011). *The sonification handbook*. Logos Verlag Berlin.

Hernández, Eugenio and Guido Weiss (1996). *A first course on wavelets*. CRC press.

Hofer, Sonya (2014). "'Atomic'music: Navigating experimental electronica and sound art through microsound". In: *Organised Sound* 19.3, pp. 295–303.

Hoffmann, Peter (2009). "Music Out of Nothing? A Rigorous Approach to Algorithmic Composition by Iannis Xenakis". In.

Holtzman, Steven R (1978). *A description of an automated digital sound synthesis instrument*. University of Edinburgh, Department of Artificial Intelligence.

Hong, Ding (2014). *Rethinking Form Through Transformation: Analytical Studies of Debussy's" Prélude À L'après-midi D'un Faune" and" Jeux"*. Chinese University of Hong Kong.

Hookway, Branden (2014). *Interface*. MIT Press.

Huhtamo, Erkki and Jussi Parikka (2011). *Media archaeology: Approaches, applications, and implications*. Univ of California Press.

Iverson, Jennifer (2017). "Learning the studio: Sketches for mid-century electronic music". In: *Contemporary Music Review* 36.5, pp. 362–387.

— (2018). *Electronic inspirations: Technologies of the cold war musical avant-garde*. Oxford University Press.

Johnson, Tom (2006). *Symmetries*. CD Karnatic Lab Records – KLR 010.

Jones, Evan (2001). "Residue-class sets in the music of Iannis Xenakis: an analytical algorithm and a general intervallic expression". In: *Perspectives of New Music*, pp. 229–261.

Jong, Taeke M de and DJM Van Der Voordt (2002). *Ways to study and research: urban, architectural, and technical design*. Ios Press.

Kaegi, Werner (1967). *Was ist elektronische Musik*. Orell Füssli.

Kaegi, Werner and Stan Tempelaars (1978). "Vosim-a new sound synthesis system". In: *Journal of the Audio Engineering Society* 26.6, pp. 418–425.

Kemp, David T (1978). "Stimulated acoustic emissions from within the human auditory system". In: *The Journal of the Acoustical Society of America* 64.5, pp. 1386–1391.

Kendall, Gary S, Christopher Haworth, and Rodrigo F Cádiz (2014). "Sound synthesis with auditory distortion products". In: *Computer Music Journal*.

Kirk, Jonathon (2010). *Otoacoustic emissions as a compositional tool*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library.

Kittler, Friedrich, Dorothea Von Mücke, and Philippe L Similon (1987). "Gramophone, film, typewriter". In: *October* 41, pp. 101–118.

Koenig, Gottfried Michael (1978). “Composition processes”. In: *UNESCO Computer Music: Report on an International Project Including the International Workshop Held at Aarhus, Denmark in*, pp. 105–126.

— (1987). “Genesis of form in technically conditioned environments”. In: *Journal of New Music Research* 16.3, pp. 165–175.

Koenigsberg, Christopher K (1991). “Karlheinz Stockhausen’s new morphology of musical time”. In: *Oakland, Mills College*.

Kramer, Jonathan D (1988). “The time of music: New meanings, new temporalities, new listening strategies”. In.

Kramer, M et al. (2003). “Simultaneous single-pulse observations of radio pulsars-IV. Flux density spectra of individual pulses”. In: *Astronomy & Astrophysics* 407.2, pp. 655–668.

Krasner, Glenn (1980). “Machine tongues VIII: The design of a Smalltalk music system”. In: *Computer Music Journal* 4.4, pp. 4–14.

Kromhout, Melle Jan (2020). “Hearing pastness and presence: the myth of perfect fidelity and the temporality of recorded sound”. In: *Sound Studies* 6.1, pp. 29–44.

— (2021). “Electronic Contingencies: Goeyvaerts’s Sine Wave Music and the Ideal of Perfect Sound”. In: *Imperfections: Studies in Mistakes, Flaws, and Failures*, p. 129.

Kronland-Martinet, Richard (1988). “The wavelet transform for analysis, synthesis, and processing of speech and music sounds”. In: *Computer Music Journal* 12.4, pp. 11–20.

Latour, Bruno (2010). “An attempt at a" compositionist manifesto"”. In: *New literary history* 41.3, pp. 471–490.

Lau, Juan Sebastián Lach (2017). “Proportion, Perception, Speculation: Relationship Between Numbers and Music in the Construction of a Contemporary Pythagoreanism”. In: *The Musical-Mathematical Mind*. Springer, pp. 131–141.

Levin, Thomas Y (2003). ““Tones from out of Nowhere”: Rudolph Pfenninger and the Archaeology of Synthetic Sound”. In: *Grey Room*, pp. 32–79.

London, Justin (2012). *Hearing in time: Psychological aspects of musical meter*. Oxford University Press.

Lostanlen, Vincent (2017). “Convolutional operators in the time-frequency domain”. PhD thesis. Université Paris sciences et lettres.

— (2018). “On time-frequency scattering and computer music”. In: *arXiv preprint arXiv:1810.04506*.

Lostanlen, Vincent and Florian Hecker (2019). “The shape of RemiXXXes to come: audio texture synthesis with time-frequency scattering”. In: *arXiv preprint arXiv:1906.09334*.

Luque, Sergio (2006). “Stochastic synthesis: Origins and extensions”. In: *Master’s Thesis, Institute of Sonology, Royal Conservatory, The Netherlands*.

— (2009). “The stochastic synthesis of iannis xenakis”. In: *Leonardo Music Journal*, pp. 77–84.

Magnusson, Thor (2019). *Sonic writing: technologies of material, symbolic, and signal inscriptions*. Bloomsbury Academic.

Malherbe, Claudy, Gérard Assayag, and Michèle Castellengo (1985). “Functional Integration of Complex Instrumental Sounds in Musical Writing”. In: *ICMC*.

Malina, Roger F and Sean Cubitt (2008). *Software studies: A lexicon*. Mit Press.

Manning, Peter (2013). *Electronic and computer music*. Oxford University Press.

Marino, Mark C (2006). “Critical code studies”. In: *Electronic book review* 4.

Maxwell, James Clerk and Peter Pesic (2001). *Theory of heat*. Courier Corporation.

Mazzola, Guerino (2012). *The topos of music: geometric logic of concepts, theory, and performance*. Birkhäuser.

McDermott, Josh H, Andrew J Oxenham, and Eero P Simoncelli (2009). "Sound texture synthesis via filter statistics". In: *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*. IEEE, pp. 297–300.

McGinness, John (1998). "From Movement to Moment: Issues of Expression, Form, and Reception in Debussy's Jeux". In: *Cahiers Debussy* 22, pp. 51–74.

Mesnage, Marcel and André Riotte (1993). *Modélisation informatique de partitions, analyse et composition assistée*.

Messiaen, Olivier (1956). *The Technique of My Musical Language: Text. 1st Volume*. Vol. 1. Alphonse Leduc.

Meyer, Yves (1993). "Wavelets-algorithms and applications". In: *Wavelets-Algorithms and applications Society for Industrial and Applied Mathematics Translation., 142 p.*

Moholy-Nagy, László (2004). "Production-Reproduction: Potentialities of the Phonograph". In: *Audio Culture: Readings in Modern Music. London: Continuum.(Chapter originally published 1922.)*

Moles, Abraham A (1966). "Information theory and esthetic perception". In.

Molitor, Claudia and Thor Magnusson (2021). "Curating experience: Composition as cultural technology–a conversation". In: *Journal of New Music Research* 50.2, pp. 184–189.

Moorer, James A (1978). "The use of the phase vocoder in computer music applications". In: *Journal of the Audio Engineering Society* 26.1/2, pp. 42–45.

Morlet, Jean (1983). "Sampling theory and wave propagation". In: *Issues in acoustic Signal—image processing and recognition*. Springer, pp. 233–261.

Morris, Stephen J and OCZ Gotel (2006). "Flow diagrams: rise and fall of the first software engineering notation". In: *International Conference on Theory and Application of Diagrams*. Springer, pp. 130–144.

Nakai, You (2021). *Reminded by the Instruments: David Tudor's Music*. Oxford University Press.

Neuhaus, Christiane, Thomas R Knösche, and Angela D Friederici (2009). "Similarity and Repetition". In: *Annals of the New York Academy of Sciences* 1169.1, pp. 485–489.

Nilson, Click (2007). "Live coding practice". In: *Proceedings of the 7th international conference on New interfaces for musical expression*, pp. 112–117.

Noll, Thomas, Moreno Andreatta, and Carlos Agon (2006). "Computer-aided transformational analysis with tone sieves". In.

Novati, Maria Maddalena and John Dack (2012). *The Studio Di Fonologia: a musical journey 1954-1983, update 2008-2012*. Ricordi.

O'Farrell, Clare (2005). *Michel Foucault: Key Concepts*. URL: `https://michel-foucault.com/key-concepts/`.

Parikka, Jussi (2011). "Operative media archaeology: Wolfgang Ernst's materialist media diagrammatics". In: *Theory, Culture & Society* 28.5, pp. 52–74.

— (2013). *What is media archaeology?* John Wiley & Sons.

Pasler, Jann (1982). "Debussy," Jeux": Playing with Time and Form". In: *Nineteenth-Century Music*, pp. 60–75.

Patteson, TW (2012). "The time of roland kayn's cybernetic music". In: *Travelling Time, Sonic Acts XIV*, pp. 47–67.

Perry, Imani (2004). *Prophets of the hood: Politics and poetics in hip hop*. Duke University Press.

Peter Ablinger (composer) Klangforum Wien (ensamble), Silvain Cambreling (soloist) (2000). *Quadraturen IV (Selbstportrait mit Berlin)*. CD, Kairos.

Petersen, Tracy and S Boll (1983). "Critical band analysis-synthesis". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 31.3, pp. 656–663.

Petersen, Tracy Lind (1981). “ACOUSTIC SIGNAL PROCESSING IN THE CONTEXT OF A PERCEPTUAL MODEL.” In.

Pietruszewski, Marcin (2020a). *Auditory Sieve*. CD etat.

Pietruszewski, Marcin (2020b). “From Xenakis's UPIC to Graphic Notation Today”. In: ed. by Ludger Brümmer Peter Weibel and Sharon Kanach. Cambridge University Press. Chap. The Digital Instrument as an Artefact, pp. 105–107.

Portnoff, M (1976). “Implementation of the digital phase vocoder using the fast Fourier transform”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 24.3, pp. 243–248.

Pousseur, Henri and David Behrman (1966). “The question of order in new music”. In: *Perspectives of New Music*, pp. 93–111.

Rheinberger, Hans-Jörg (1992). “Experiment, difference, and writing: I. Tracing protein synthesis”. In: *Studies in History and Philosophy of Science Part A* 23.2, pp. 305–331.

Rheinberger, Hans-Jorg (1997). “Experimental complexity in biology: Some epistemological and historical remarks”. In: *Philosophy of Science* 64, S245–S254.

Rheinberger, Hans-Jörg (2004). “„Experimental Systems: Entry Encyclopedia for the History of the Life Scineces “”. In: *The Virtual Laboratory: Essays and Resources on the Experimentalization of Life*.

Rheinberger, Hans-Jörg (2013). “Forming and being informed: Hans-Jörg Rheinberger in conversation with Michael Schwab”. In: *Experimental systems: future knowledge in artistic research*. Leuven University Press, pp. 198–219.

Rheinberger, Hans-Jörg and Joseph S Fruton (1997). *Toward a history of epistemic things: Synthesizing proteins in the test tube*. Stanford University Press Stanford, CA.

Riemann, Hugo (1992). “Ideas for a Study" On the Imagination of Tone"”. In: *Journal of Music Theory* 36.1, pp. 81–117.

Riotte, André (1992). “Formalisation des échelles de hauteurs en analyse et en composition”. In: *Actes du Colloque” Musique et Assistance Informatique.” Marseille: Musique et Informatique de Marseille*.

Risset, Jean-Claude (1986). “Pitch and rhythm paradoxes: Comments on “Auditory paradox based on fractal waveform”[J. Acoust. Soc. Am. 7 9, 186–189 (1986)]”. In: *The journal of the Acoustical Society of America* 80.3, pp. 961–962.

— (2005). “Horacio Vaggione: Towards a syntax of sound”. In: *Contemporary Music Review* 24.4-5, pp. 287–293.

Risset, Jean-Claude and David L Wessel (1982). “Exploration of timbre by analysis and synthesis”. In: *The psychology of music* 2.151, pp. 26–58.

Roads, Curtis (1985). *Composers and the Computer*. Vol. 2. AR Editions.

— (1988). “Introduction to granular synthesis”. In: *Computer Music Journal* 12.2, pp. 11–13.

— (1996a). “Early electronic music instruments: Time line 1899-1950”. In: *Computer Music Journal* 20.3, pp. 20–23.

— (1996b). *The computer music tutorial*. MIT press.

— (2001). “Sound composition with pulsars”. In: *Journal of the Audio Engineering Society* 49.3, pp. 134–147.

— (2004). *Microsound*. MIT press.

— (2015). *Composing electronic music: a new aesthetic*. Oxford University Press, USA.

— (2020). “Sound composition with pulsars”. In.

Roads, Curtis and Max Mathews (1980). “Interview with max mathews”. In: *Computer Music Journal* 4.4, pp. 15–22.

Röbel, Axel (2003). “A new approach to transient processing in the phase vocoder”. In.

Rodgers, Tara (2010). “Pink Noises”. In: *Pink Noises*. Duke University Press.

Rohrhuber, Julian and Alberto de Campo (2009). “Improvising Formalisation: Conversational Programming and Live Coding”. In: *New Computational Paradigms for Computer Music. Delatour France/Ircam-Centre Pompidou*.

Rohrhuber, Julian, Alberto de Campo, and Renate Wieser (2005). “Algorithms today notes on language design for just in time programming”. In: *International Computer Music Conference*, p. 291.

Rohrhuber, Julian, RT Dean, and A McLean (2018). *Algorithmic Music and the Philosophy of Time.*

Rohruber, Julian (2010). *«> and <» : Two simple operators for composing processes at runtime*. Accessed on 20.08.2020.

Rosenboom, David (2021). “Illusions of Form”. In: *Sound Work: Composition as Critical Technical Practice*, p. 35.

Salzer, Felix (1962). *Structural hearing: Tonal coherence in music*. Vol. 1. Courier Corporation.

Sanfilippo, Dario (2020). “Complex musical behaviours via time-variant audio feedback networks and distributed adaptation: a study of autopoietic infrastructures for real-time performance systems”. PhD thesis. University of Edinburgh.

Sanfilippo, Dario and Andrea Valle (2013). “Feedback systems: An analytical framework”. In: *Computer Music Journal* 37.2, pp. 12–27.

Schaeffer, Pierre (2017). “Treatise on musical objects”. In: *Treatise on Musical Objects*. University of California Press.

Schloezer, Boris de (1979). *Introduction à J.-S. Bach: essai d’esthétique musicale*. Vol. 415. Gallimard.

Schmickler, Marcus (2007). *Altars of Science*. CD Editions Mego.

Schmid, Anne-Françoise and Armand Hatchuel (2014). “On generic epistemology”. In: *Angelaki* 19.2, pp. 131–144.

Schörkhuber, Christian and Anssi Klapuri (2010). “Constant-Q transform toolbox for music processing”. In: *7th sound and music computing conference, Barcelona, Spain*, pp. 3–64.

Schwab, Michael (2013). *Experimental systems: future knowledge in artistic research*. Leuven University Press.

— (2014). “The Exposition of Practice as Research as Experimental Systems”. In: *Artistic Experimentation in Music: An Anthology*, pp. 31–40.

Scoville, Thomas (1998). “The elements of style: UNIX as literature”. In: *Performance Computing. Tersedia di http://www. wenet. net/˜ scoville/PCarticle. html.*

Scrivener, Stephen et al. (2013). “Towards a Practice of Novel Epistemic Artefacts”. In: *Experimental Systems: Future Knowledge in Artistic Research*, pp. 135–150.

Serra, Marie-Hélène (1993). “Stochastic composition and stochastic timbre: Gendy3 by Iannis Xenakis”. In: *Perspectives of New Music*, pp. 236–257.

Siedenburg, Kai, Ichiro Fujinaga, and Stephen McAdams (2016). “A comparison of approaches to timbre descriptors in music information retrieval and music psychology”. In: *Journal of New Music Research* 45.1, pp. 27–41.

Simondon, Gilbert and Andrew Iliadis (2019). “Form, Information, and Potentials: February 27, 1960, Session of the Société française de philosophie”. In: *Philosophy Today* 63.3, pp. 571–583.

Smith, Hazel (2009). *Practice-led research, research-led practice in the creative arts*. Edinburgh University Press.

Smith III, Julius O (1991). “Viewpoints on the history of digital synthesis”. In: *Proceedings of the International Computer Music Conference*. INTERNATIONAL COMPUTER MUSIC ACCOCIATION, pp. 1–1.

Snyder, Bob (2000). *Music and memory*. MIT press Cambridge, MA.

Solomos, Makis (2019). *From music to sound: The emergence of sound in 20th-and 21st-century music*. Routledge.

Squibbs, Ronald (2002). "Some observations on pitch, texture, and from in Xenakis' Mists". In: *Contemporary Music Review* 21.2-3, pp. 91–108.

Squibbs, Ronald James (1996). *An analytical approach to the music of Iannis Xenakis: studies of recent works*. UMI Dissertation Services.

Stautner, John Paul (1983). "Analysis and synthesis of music using the auditory transform". PhD thesis. Massachusetts Institute of Technology.

Sterne, Jonathan (2003). *The audible past: Cultural origins of sound reproduction*. Duke University Press.

Stockhausen, Karlheinz (1958). "Structure and experiential time". In: *Die Reihe* 2, pp. 64–74.

— (1959). "How time passes". In: *die Reihe* 3, pp. 10–40.

— (1963a). "Texte zu eigenen Werken, zur Kunst anderer". In: *Band* 2, p. 102.

— (1963b). *Von Webern zu Debussy: Bemerkungen zur statistischen Form*. MD Schauberg.

— (1989). "Four criteria of electronic music". In: *Stockhausen on Music: Lectures and Interviews. New York: Marion Boyars*.

Stockhausen, Karlheinz and Elaine Barkin (1962). "The concept of unity in electronic music". In: *Perspectives of New Music*, pp. 39–48.

Supper, Martin (1997). *Elektroakustische Musik und Computermusik: Geschichte, Ästhetik, Methoden, Systeme*. Wolke.

Tenney, James (2014). *From Scratch: Writings in Music Theory*. University of Illinois Press.

Thomson, James (1879). "XVIII. On the Jointed Prismatic Structure in Basaltic Rocks". In: *Transactions of the Geological Society of Glasgow* 6.1, pp. 95–110.

— (2016). *Collected papers in physics and engineering*. Cambridge University Press.

Thomson, Phil (2004). "Atoms and errors: towards a history and aesthetics of microsound". In: *Organised Sound* 9.2, pp. 207–218.

Tipei, Sever (1975). *MP1: a computer program for music composition*. Ann Arbor, MI: Michigan Publishing, University of Michigan Library.

Toop, Richard (1979). "Stockhausen and the Sine-Wave: The Story of an Ambiguous Relationship". In: *The Musical Quarterly* 65.3, pp. 379–391.

Truax, Barry (1986). "Computer music language design and the composing process". In: *The language of electroacoustic music*. Springer, pp. 155–173.

— (1988). "Real-time granular synthesis with a digital signal processor". In: *Computer Music Journal* 12.2, pp. 14–26.

Ungeheuer, Elena (1992). *Wie die elektronische Musik" erfunden" wurde...: Quellenstudie zu Werner Meyer-Epplers musikalischem Entwurf zwischen 1949 und 1953*. Schott.

Vaggione, Horacio (1991). "A note on object-based composition". In.

— (1993). "Determinism and the false collective about models of time in early computer-aided composition". In: *Contemporary Music Review* 7.2, pp. 91–104.

— (1994). "Timbre as syntax: a spectral modeling approach". In: *Contemporary Music Review* 10.2, pp. 73–83.

Valle, Andrea and Dario Sanfilippo (2012). "Towards a Typology of feedback Systems." In: *ICMC*.

Varèse, Edgard (1967). "Rhythm, form and content". In: *Contemporary Composers on Contemporary Music. New York: Norton*.

Varèse, Edgard and Chou Wen-Chung (1966). "The liberation of sound". In: *Perspectives of new music* 5.1, pp. 11–19.

Vetterli, Martin and Cormac Herley (1992). “Wavelets and filter banks: Theory and design”. In: *IEEE transactions on signal processing* 40.ARTICLE, pp. 2207–2232.

Wang, Ge (2007). “A history of programming and music”. In: *The Cambridge Companion to Electronic Music*, pp. 55–71.

Wardrip-Fruin, Noah (2011). “14. Digital Media Archaeology: Interpreting Computational Processes”. In: *Media Archaeology*. University of California Press, pp. 302–322.

Wason, Robert W, Elizabeth West Marvin, and Hugo Riemann (1992). “Riemann’s" Ideen zu einer’Lehre von den Tonvorstellungen’": An annotated translation”. In: *Journal of Music Theory* 36.1, pp. 69–79.

Wiener, Norbert (1976). “Spatio-temporal continuity, quantum theory and music”. In: *The concepts of space and time*. Springer, pp. 539–546.

Williams, Sean (2015). “Technical influence and physical constraint in the realisation of Gesang der Jünglinge”. In.

— (2016a). “Interpretation and performance practice in realizing Stockhausen’s Studie II”. In: *Journal of the Royal Musical Association* 141.2, pp. 445–481.

— (2016b). “The Hohner Electronium: a 1950s portable monophonic valve synthesizer”. In.

Wright, Matthew (2005). “Open Sound Control: an enabling technology for musical networking”. In: *Organised Sound* 10.3, pp. 193–200.

Wright, Matthew et al. (2001). “Managing complexity with explicit mapping of gestures to sound control with OSC”. In: *ICMC*. Citeseer.

Xenakis, Iannis (1965). “La voie de la recherche et de la question”. In: *Preuves* 177, pp. 33–36.

— (1968). *Vers une philosophie de la musique*. Klincksieck.

— (1988). “Sur le temps in Redécouvrir le temps”. In: *Revue de l’Université de Bruxelles* 1-2, pp. 193–200.

— (1992). *Formalized music: thought and mathematics in composition*. 6. Pendragon Press.

— (1996). “Determinacy and indeterminacy”. In: *Organised Sound* 1.3, pp. 143–155.

Xenakis, Iannis and John Rahn (1990). “Sieves”. In: *Perspectives of New Music*, pp. 58–78.

Xenakis, Iōannēs and Bálint András Varga (1996). *Conversations with Iannis Xenakis*. Faber & Faber.

Zaplitny, Michael and Iannis Xenakis (1975). “Conversation with Iannis Xenakis”. In: *Perspectives of New Music*, pp. 86–103.

Zdebik, Jakub (2012). *Deleuze and the diagram: Aesthetic threads in visual organization*. Bloomsbury Publishing.

Zielinski, Siegfried (1996). “Media archaeology”. In: *CTheory*, pp. 7–11.