2023

Towards Quantum Computing for Audio and Music Expression

Itaborai, Paulo Vitor

http://dx.doi.org/10.24382/5119
University of Plymouth

UNIVERSITY OF
PLYMOUTH

TOWARDS QUANTUM COMPUTING FOR AUDIO AND MUSIC EXPRESSION

by

PAULO VITOR ITABORAÍ

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

RESEARCH MASTER

School of Society and Culture

December 2023

Acknowledgements

First and foremost, I wish to thank Professor Eduardo Reck Miranda, my Director of Studies. He has provided me with a countless number of life-changing opportunities. Thanks to his pioneering vision, I was not only able to complete this research, but also engage in a plethora of fascinating projects that have had an impact beyond academia and into the European artistic scene. Thanks to him, I was able to perform alongside him and other highly esteemed artists across in the UK, Spain and Germany. I also thank my second supervisor, David Jenkins, for his essential support for completing this project.

I extend my gratitude to my Viva voce examiners, Prof. Bob Coecke and Dr. George Sioros, for their insightful discussions which enriched my academic journey.

Many thanks to everyone at the IKLECTIK Art Lab in London and in ICFO Bercelona. Their venues/events gave life to my first solo acousmatic piece, "*Rasgar, Saber*" settling a milestone in my career.

Dino Vicente deserves a special mention for our artistic collaboration that led to the other artistic output described here, *"Rever"*. Our virtual dialogues, which often lasted more than 5 hours per call, enriched the poetics, the critical thinking, and the human elements of quantum computing in music.

The same goes to Peter Thomas (and his alter-ego Cephas Teom), for our friendship and many collaborations have been invaluable. From technical implementations to live performances and academic publications, his presence has been a central reference to my endeavours. I will forever miss our Thursdays and his inherent contagious motivation.

Many thanks to the companies that provided resources that impacted this work, particularly, Moth Quantum and Quantinuum - for supporting Eduardo's research in the field and impacting my own; IBM Quantum for providing academic exclusive access to their quantum devices, some of which were used in real time in live musical performances.

My deepest gratitude to the ICCMR and the University of Plymouth for trusting me with a scholarship that viabilised this postgraduate research.

I am grateful to OCH - for our important academic collaborations; Alex Buck - for his inspiration; Tomas Mistrorigo - For giving voice to my quantum sounds and papers; and Finn Jackson - who has invented the origami mask techniques that I used to design my own for *Rasgar, Saber*.

I owe much to many, many friends that stayed in Brazil (particularly, friends from the Cherateria ensemble and those with whom I have shared Many lifetimes) and some few that shared part of the cotidian life here in Plymouth (special mentions to Hari Shaji). To Mariana Jó and Roberto Bodo, whom I have been travelling with into the Computer Music wilderness since the beginning.

Special gratitude to my family, particularly my parents, Silvia Carla and Olímpio. Their unconditional support and unwavering belief in my pursuits have been the bedrock of my overseas journey.

To my previous mentors and teachers, who gave me the essential knowledge to carry on this research.

Without Pedro Carvalho, Dante Lara, Fabiane Yoshinaga, Fabrizia de Marco, Gabriel Mardegan, and Matheus Rauh, it would have been impossible to overcome the most difficult moments of migrant life.

I dedicate this work to my godfather, Ivon Mendes de Barros. May this research be a small development of his legacy in my life.

Author's declaration

At no time during the registration for the degree of Research Master has the author been registered for any other University award without prior agreement of the Doctoral College Quality Sub-Committee.

Work submitted for this research degree at the University of Plymouth has not formed part of any other degree either at the University of Plymouth or at another establishment.

Publications:

Itaboraí, P.V., Miranda, E.R. (2022). Quantum Representations of Sound: From Mechanical Waves to Quantum Circuits. In: Miranda, E.R. (eds) Quantum Computer Music. Springer, Cham. `https://doi.org/10.1007/978-3-031-13909-3_10`. Abridged preprint version available at ArXiv: arXiv:2301.01595 **[quant-ph]**

Itaboraí, P. V. (2022). Quantumaudio Module (Version 0.0.2) [Computer software]. `https://github.com/iccmr-quantum/quantumaudio`

Hamido, O.C., Itaboraí, P.V. (2023). OSC-Qasm: Interfacing Music Software with Quantum Computing. In: Johnson, C., Rodríguez-Fernández, N., Rebelo, S.M. (eds) Artificial Intelligence in Music, Sound, Art and Design. EvoMUSART 2023. Lecture Notes in Computer Science, vol 13988. Springer, Cham. `https://doi.org/10.1007/978-3-031-29956-8_24`. Preprint version available at ArXiv: arXiv:2212.01615 **[cs.ET]**

Miranda, E. R., Thomas, P., & Itaboraí, P. V. (2023). Q1Synth: A Quantum Computer Musical Instrument. Applied Sciences, 13(4), 2386. `https://doi.org/10.3390/app13042386`

Itaboraí, P. V., Schwägerl, T., Yáñez, M. A., Crippa, A., Jansen, K., Miranda, E. R., & Thomas, P. (2023). Variational Quantum Harmonizer: Generating Chord Progressions and Other Sonification Methods with the VQE Algorithm. 2nd International Symposium on Quantum Computing and Musical Creativity (ISQCMC Berlin), Berlin. `https://doi.org/10.5281/zenodo.10206731`

Presentations and conferences attended:

Remote Lecture - São Paulo State University, on Music Technology and Quantum Computer Music, for the undergraduate multimedia course, ministered with prof. Laan Mendes de Barros (28 Jun 2022).

ICCMR Seminar series - University o Plymouth. Presentation of Quantum Computing Sound Synthesis (16 Nov 2022)

Seminars - A series of seminars on Quantum Representations of Audio presented in Germany with prof. Miranda. First, at the German Electron Synchrotron DESY intitute in Zeuthen, for the Applied Quantum Computing research group (08 Nov 2022). Second, in the Technical University of Berlin, for the Audio communication group (24 Jan 2023).

Remote Lectures - Two postgrad lectures ministered with prof. Miranda at TU Berlin, for the audio engineering and communication courses. The theme was on a detailed introduction on quantum representations of audio (28 Jun - 05 Jul 2023)

Internship - One week long academic visit at DESY Zeuthen for kickstarting a collaboration on Quantum Computer Algorithm Sonification, leading to the publication "Variational Quantum Harmoniser" cited above (23-27 Jan 2023)

Poster Paper - Remote poster presentation with Omar Costa Hamido (first author) at EvoMUSART 2023 conference (17 Apr 2023)

Conference Presentation - ICFO Quantum Sounds, Barcelona. Lecture/Concert on the approaches of quantum computer music research being developed at ICCMR. Ministered/performed with Peter Thomas and Eduardo Miranda (04-06 Jun 2023)

Musical Performance - Performance with the Q1Synth musical instrument, alongside Peter Thomas and Eduardo Miranda, on the occasion of the Quantum Computer Music book launch at Goethe Institut London (08 Dec 2022)

Musical Performance - ICCMR music performance on Quantum Computer Music at IKLECTIK Art Lab. Performances with Eduardo and Peter (Spinnings, Dependent Origination) Premiere of *Rasgar, Saber* (24 Apr 2023)

Musical Performance - Technical assistant and performer of fixed -electronics for the *Multiverse Symphony*, by Eduardo Miranda (for ensemble and quantum computer), with the Contemporary Chamber Orchestra Elbe; on the occasion of the inauguration of the Ligeti Centre in Hamburg (03 May 2023)

Musical Performance - Technical assistant and fixed electronics performance for the piece *Qubism* for quantum computer and ensemble, by Eduardo Miranda, with London Sinfonietta orchestra. Kings Place (24 Jun 2023)

Word count for the main body of this thesis: **19907**

Signed: Paulo Vitor Itaborai de Barros

Date: 27/08/2023

Abstract

Paulo Vitor Itaboraí

Towards Quantum Computing For Audio and Music Expression

This research delves into the initial approaches of the use of emerging Quantum Computing technologies in music practice, anchored on preliminary studies on Quantum Representation of Audio (QRA). The approach is interdisciplinary, and attempts to explore the object - QRA - both from a technical scope and an artistic scope. A systematic review unifies diverse QRA techniques, spotlighting the need to bring accessibility to these techniques and terminologies (dense in theoretical physics formalism) for a computer music and/or artistic audiences, in a didactic manner. In this review, some additional techniques initially only discussed for representing images are translated into audio. An improved classification system is proposed to better organise the QRA field. This led to the observation of a pair of strategies that can be converted into each other with the use of a single quantum gate, sparking ideas for future Quantum Audio Conversion algorithms. To bridge theory and practice, a Python package named `quantumaudio` was prototyped, enabling a flexible use of quantum audio encoding and decoding schemes for three QRA strategies. Demonstrations of simple workflows with `quantumaudio` underlined their potential to generate constituent sound materials for electronic music, on account of a simple digital musical interface prototype that maps quantum signals to wavetable oscillators. The journey culminated with one of the demonstrations - "Geiger-Counter Effect" (GCE) - being integrated into two artistic processes, which explored the juxtaposition of quantum measurements with early analog music aesthetics. The first became an acousmatic study, *"Rasgar, Saber"*, whereas the second materialised in the form of *Rever*, a design/improvisation process of crafting a hybrid quantum-digital-analog musical instrument, inspired by the work of Walter Smetak. This study hopes to lay foundations and examples for the dawning approaches in Quantum Computer Music, starting from the essence of organised sound.

Contents

List of Figures

List of Tables

List of Media

Note: All listed media below can be found annexed in the accompanying folder. See the thesis' directory for details.

Available Source Code:

The Source Code for the `quantumaudio` python package can be found on GitHub:
https://github.com/iccmr-quantum/quantumaudio.

There is also a rendered documentation page with examples on ReadTheDocs.io:
https://quantumaudio.readthedocs.io/en/latest/

Chapter 1

Introduction

"A 'hesitant advance': here's a method; [...] those who advance hesitate because they do not want to know where they are going - if they already knew, why would they go there? [...] Hesitating is an effect of the action of discovering; only those who have already discovered, who have already put an end to their research process, do not hesitate. "

— Gonçalo M. Tavares, Atlas of the Body and of Imagination, p. 22 (Translated from Portuguese)

The ability to imagine, understand and organise sound as music has always been closely related to the mathematical, scientific and technological knowledge of a historical period, geographical place and culture. These become a motivation for the academic study and experimentation of emerging technologies such as Quantum Computing (QC) in order to understand both their impact on music making - and, conversely, the impact that science and technology will receive from it.

1.1 Quantum Computer Music

In the first morning lights of the research field of Quantum Computer Music (QCM), foundations of a bridge between the worlds of quantum computing and music technology are being built. Early studies on both theoretical and practical aspects of this bridge - made by interdisciplinary researchers experimenting with QC in music creativity - were shared in a symposium (ISQCMC, 2021) and edited into two books (Miranda, 2022a,b).

"Quantum Music" is Not a new endeavour

There are several possible tracks of investigation when the intersecctions between physics and music are being considered. For instance, we could explore the influence of music in the development of mathematical models and scientific theories, as well as technology. Several examples of such influences can be found: From Pythagoras and the harmonic series; Kepler and the Harmony of the Spheres; Saveur and the birth of musical acoustics; Bernoulli and Euler and the vibrating string problem; The Fourier series, and many others (Maor, 2020).

Conversely, it is notable how particularly the music composition of the mid-20th century was itself influenced by mathematics and physics. It is necessary to acknowledge that the influence of the quantum theories in music is not recent nor insignificant and can be traced back to the early years of electronic music. It was inspired by Shrodinger's equation and Heisenberg's uncertainty principle that led Dennis Gabor to idealise, in 1947, the concept of Acoustic Quanta (Gabor, 1947), or time-frequency sound atoms. The primary idea is to decompose a sound into small localised sound grains. His formulations later seeded the development of both wavelet analysis (Mallat, 1999) and granular synthesis (Roads, 2002)[1].

Furthermore, physicist Meyer-Eppler was aware of Gabor's work (Cline, 2019). As one of the founders of the Westdeutscher Rundfunk (WDR) electronic music studio in Cologne, and a researcher in Information Theory himself, he exerted a critical influence on Stockhausen's Serialist music and also Xenakis' work (Barthel-Calvet, 2022; Xenakis, 1992). In analogy, the ability of manipulating Quantum Information using quantum devices may have its own stage of discussion in contemporary music composition.

The field of Quantum Computer Music is at a stage where fundamental questions are being asked, torwards incorporating state-of-the-art Noisy Intermediate-Scale Quantum Computig (NISQ) systems into music practice. NISQ devices are characterized by their small to medium capacity of around 5-100 quantum bits, or qubits; as the term "noisy" suggests, these devices suffer from quantum decoherence and other factors that might introduce noise and untraceable calculation errors into the system. NISQ machines provide fertile ground for theoretical study, proof-of-concept validation of models on the path to fault-tolerance (Gill et al., 2021); including Quantum Computer Music.

[1]Despite its undeniable importance, the Gabor atom *is not quantum sound in a physical sense*. The mathematical construction of the Gabor atom is *inspired* on a Gaussian particle (Mallat, 1999), but it is used to decompose classical signals into grains. In other words, altough quantum-inspired, acoustic quanta are grains of deterministic, *classical* sound information, as are its succeeding artistic developments. In contrast, this should not be confused with 'Sound Quanta' or Phonons, which do represent microscopic vibrations with quantised energy that behave as quasi-particles when propagating inside condensed matter (Srivastava, 2019). Phonons contain Quantum Information and are by nature non-deterministic. Phonons use position and momentum as their canonical coordinates, whereas the Gabor atom uses time and frequency.

1.2 Methodologies in Quantum Computer Music

1.2.1 Interdisciplinary Research: Two Ways of Holding a Magnifying Glass

Computer music research works, although interdisciplinary, usually can be put into either a more strongly technical-oriented or artistic-oriented research. In other words, if computer music happens to be a bridge between two fields, a researcher needs to start from solid ground, in one of its ends, and walk towards the middle. This utopic *middle* would be a singularity, where two inconciliable paradigms live in perfect and non-conflicting symbiosis.

The writer Gonçalo M. Tavares frames this contrast by dialoguing with texts of Bachelard in a stimulating way:

> *We are therefore faced with two states of attention, with two ways of holding a magnifying glass: the scientist who only looks for the new in order to eliminate it by repeating his gaze; they hold the magnifying glass in order to tame the new, to discover the formula of repetition, the formula which, by eliminating the unexpected in the world, soothes them. Meanwhile, the imaginer picks up the magnifying glass to discover the new, and when they discover it, they immediately set off for another place[...]. And here you are not looking for calmness, but for a shock. The imaginer does not want to be assured that they can return to the same place and see the same thing, but, on the contrary, that they can leave the place they know at any time.*(†) [2] (Tavares, 2021, p. 368).

This work will, to a certain extent, attempt to hold this magnifier in both ways, delimiting two scopes. The reader could then imagine that there are two *personas*, starting from different extremes, working their way closer to the interdisciplinary singularity. The object at the center is Quantum Computer Music. In one end, there will be an artist that sees quantum computers as a mysterious new technology and has to learn how it is operated. In the other, there will be a technician attempting to find aesthetical meaning amongst their tools.

While reading through the work of Alex Buck (2018) on Computer Assisted Composition and improvisation, I stumbled upon a definition reflected by Vilém Flusser and highlighted by him[3] - *fantasia*

[2]For the remainder of this text, direct citations and concepts that are followed by a dagger symbol '(†)' are translations made by the author from a source originally written in Portuguese.

[3]Ever since the fifteenth century, Occidental civilisation has suffered from the divorce into two cultures: science and

essata (Flusser, 1986, p.331) - which he used to signify part of his artistic process; It further solidifies and complements the motivation behind the dual approach applied in this text.

1.3 Scope: The Quantum Computer as a Musical Instrument

When involving the early stages of development of a new technology in Music, the researcher's activities might be no different from the practice of experimental luthery. That is, to imagine a universe of sonic possibilities outlined by a (yet to be) musical instrument. By means of musical instruments an artist can not only control and materialise sounds, but also reflect, abstract, generalise, or formalise musical structures. Thus, a suitable initial point for investigating the field could then be to ask:

Research Question 1: *How to use Quantum Computing as a Musical Instrument?* (**RQ1**)

This question is, in many ways, accompanied by the personal motivation to understand the *onthology of musical instruments*, in general. For that reason, throughout this introduction and later in chapters 5 and 6, there will be some effort to bring forth different understandings of this concept from different authors, conveying a larger picture in which this work could be ultimately interpreted as a work of musical instrument design.

1.3.1 Technical Scope - Representations of Sound

Musical Instrument: The Complex System

From a physical lens, it is possible to define traditional acoustic instruments as the collective result of simple constituents. As stated by Fletcher & Rossing (2012):

> *Mechanical, acoustical, or electrical vibrations are the sources of sound in musical instruments [...]. In most instruments, sound production depends upon the collective behavior of several vibrators, which may be weakly or strongly coupled together. This coupling, along with nonlinear feedback, may cause the instrument as a whole to behave as a complex vibrating system, even though the individual elements are relatively simple vibrators.*

its techniques — the "true" and the "good for something" — on the one hand; the arts — beauty — on the other. This is a pernicious distinction. Every scientific proposition and every technical gadget has an aesthetic quality, just as every work of art has an epistemological and political quality. More significantly, there is no basic distinction between scientific and artistic research: both are fictions in the quest of truth (scientific hypotheses being fictions). Electromagnetized images do away with this divorce because they are the result of science and are at the service of imagination. They are what Leonardo da Vinci used to call "fantasia essata"

Thus, it is by drawing from this complexity that rich, organised sound can be physically achieved. To paraphrase the sentence above to talk about quantum computing: *"In quantum computers, information depends upon the collective behaviour of several qubits, which may be weakly or strongly entangled together. This entanglement, along with external interference, may cause the computer as a whole to behave as a complex quantum system, even though the individual elements are relatively simple spin $\frac{1}{2}$ states".* Thus, the operations made to information in a quantum computer could be *metaphorically* interpreted as manipulated sound information. Moreover, if the information encoded in a quantum system happens to *represent* an audio signal, it could be said that this quantum computer is part of a complex musical instrument.

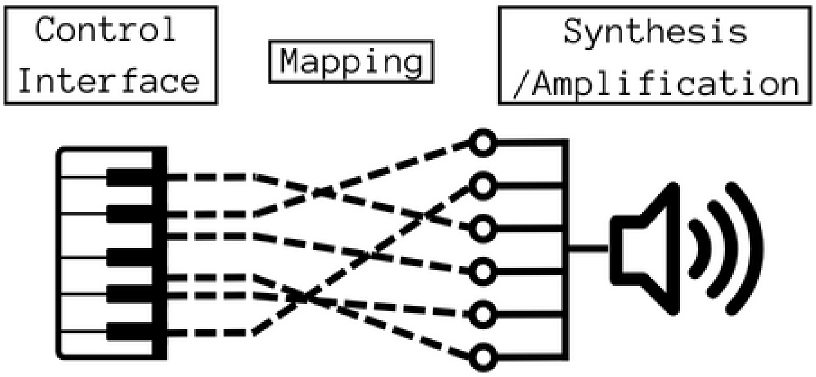


Figure 1.1: Constituents of a Digital Musical Instrument

Musical Instrument: Synthesis, Control and Interaction

Beyond the acoustic world, both Electronic and Digital medium have found their place as musical instruments. The adaptable nature of musical programming languages enables individuals to conceive and create hundreds of novel musical interfaces. Regarding this new musical interfacing ecosystem, Miranda and Wanderley (Miranda & Wanderley, 2006) have delineated succinct classification models to depict the core components that constitute a generic Digital Musical Instrument.

In simple terms, according to this model, there are three fundamental aspects of any digital musical instrument. The first is a gestural controller, the element where the performer engages with the parameters of the interface. On the other extreme, there is a mechanism responsible for the production, generation and amplification of sounds. Then, there is the parameter mapping between the controller and the synthesizer.

By reviewing the main topics discussed in both books (Miranda, 2022a,b), it can be noted that there is a stronger research focus on using quantum algorithms to generate musical structures at a higher symbolic (MIDI, musical scales, rhythmic structures, motives) and serial (pitch, tempo, intensity) levels. That is to include Quantum Random Walks (Miranda & Basak, 2022), Quantum Cellular

Automata (Miranda & Miller-Bakewell, 2022; Miranda & Shaji, 2023), Quantum Computer-Aided Composition (Hamido, 2022), Sonification of Variational Quantum Algorithms (Clemente et al., 2022), among others. To put it more simply, the literature above centers around quantum computing as a musical control interface or as a possible step in the mapping process.

Comparatively, there has been less attention to lower-level problems, such as audio and signal processing in the QCM field. However, this area of interest is crucial for musicians and music technologists, as sound waves are the building blocks of music and sound art. Understanding how to represent, store and retrieve sound in a quantum computer is a crucial step in developing new ways to understand, listen and produce sounds.

Hence, the more specific focus of this research will be to carry out an initial investigation of sound being represented in the quantum machine. More specifically, digital audio represented as quantum circuits.

Research Question 2: *How to encode Digital Audio in Quantum Circuits?* (**RQ2**)

To the extension of the known references, there is mention of a theoretical model to encode vocal-like features of a sound (Mannone & Rocchesso, 2022a), (Mannone & Rocchesso, 2022b), encoding rhythmic information as a Quantum Signal (Oshiro, 2022), and Quantum Fourier Transforms for Frequency detection (Mistry & Ortega, 2022).

Furthermore, **RQ2** ultimately results in an investigation on Quantum Representations of Audio (QRA, also appearing as QAR as in "Quantum Audio Representation"), which is a subfield of the area of Quantum Signal Processing (QSP). In the QSP literature, it is noticed that there is a strong prevalence in research on Quantum Representation of Images (QRI or QIR) compared to audio; having initial research on QRI appearing as early as 2003 (Venegas-Andraca & Bose, 2003). In contrast, the first mentions of QRA were officially proposed in 2016 (Wang, 2016); and even then it was based on a previously published QRI (Zhang et al., 2013) - a trend that would continue with later proposals ((Yan et al., 2018), (Şahin & Yilmaz, 2019) and others).

As a result, a research methodology is achieved: **To translate and adapt existing representation schemes from images (QRI) to the audio (QRA) context (M1).**

This precedence is also reflected in the number of proposed representations and designed signal processing algorithms exclusive to images - as well as subsequent literature review. At least two published surveys gathering different QRI strategies (Yan et al., 2016; Lisnichenko & Protasov, 2023b) were encountered. In contrast, a research gap is found, as there was no similar counterpart dedicated to Audio found in the literature before Itaboraí & Miranda (2022) (to the knowledge of the author), in the Quantum Computer Music Book. This book chapter, which presents an introductory survey on QRA, is a published partial result of this current work. As a result, the referred text will be revisited, updated and presented here - meaning that it will contain snippets and modified/adapted parts from its contents (chapter 2).

This leads to the succeeding methodology applied: **A Systematic Review on existing (or here first translated) Quantum Representation of Audio for the Computer Music Community (M2)**

The emphasis - "*for Computer Music*" - means that in chapter 2, there will be a careful choice of terminology and mathematical notation when writing definitions and showing equations. This work attempts to approach researchers aquaintant to DSP equations and jargons. Therefore, it integrates different proposed QRA under unified notation and a modified acronym system, simplified to the best of efforts. It aims at maximising readability and accessibility for the computer music community, bringing them closer to quantum technologies.

To realise this objective, a didactic framework was developed to present this survey, while also teaching newer audiences and highlighting the intersections between digital and quantum audio. Instead of immediately wearying the reader with a large list of methods and acronyms for audio (and their image counterparts), we begin with relevant traditional audio encoding concepts and then migrate them to a quantum context (as presented in Itaboraí & Miranda). In other words, from signals and digital arrays to quantum statevectors: A Quantum Audio Roadmap. Only then the more declarative shower begins - after giving an umbrella.

Later, to address **RQ1** technically, a selection of QRA was implemented in Python, with the aim of **Interfacing Quantum Computing Languages with Musical Software (M3)**.

With this in mind, the implementation of a simple Python package containing initial QRA implementations, providing a framework for audio QSP and artistic applications was proposed. A preliminary (beta) version (v0.0.2) was published on PyPI during the course of the program (Itaboraí, 2023) under the name `quantumaudio`.

Then, `quantumaudio` was interfaced with SuperCollider (SC) (McCartney, 2002) with the intention of designing a musical instrument using Wavetable Synthesis (WS). With that being said, we move from technical grounds towards the artistic realm.

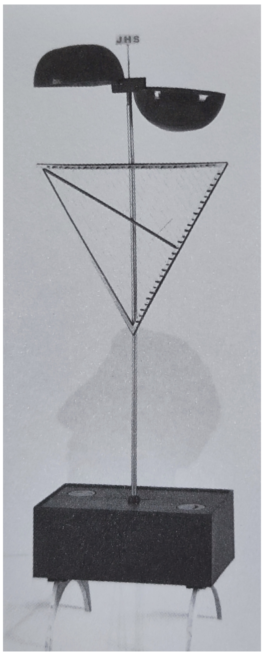


Figure 1.2: Vau, Sonic Plastic by Walter Smetak, 1969

1.3.2 Artistic Scope - The Vehicle of Sound

Now, the *imaginer* takes control of the magnifying glass.

What is a Musical Instrument? - Sound Plastics

Walter Smetak (1913-1984) was a Swiss-Brazilian composer, cellist, and instrument inventor. Trained in classical music, Smetak migrated to Brazil, where he became an influential figure in the country's avant-garde music scene. As a music professor at the Federal University of Bahia, he developed an innovative approach to musical instrument design, sistematically combining visual, architectural, improvisational and even mystical qualities to it. Smetak's dedication to sound experimentation led him to create over 150 unique instruments, which he referred to as *Sound Plastics* [†] (Plásticas Sonoras).

From the Instrument - Vehicle
From the doctrine - Content
Of improvisation - Application

These three elements constitute the integration of SOUND PLASTICS, interweaving three components in the future: Art, Science and Philosophy, initiating a SPIRITUAL ART.[...][†] (Smetak, 1967/2019, p. 26)

These sculptural instruments - crafted from various materials such as wood, bamboo, gourds, and metal - were marked by characteristic visual abstract forms (hence Plastics), reflecting Smetak's belief in a strong connection between the visual and sonic arts. They were meant to visually represent the sounds they produce. In a sense, this is also a representation of sound. His provocative style and design process often involved improvisation-as-creation methodologies and the deconstruction of concepts, often resembling the visual work of Marcel Ades et al. (2021) and the ideas of Derrida (1996). He even got as far as having said that he was himself a "music decomposer" (Scarassatti, 2008, p. 83).

Considering our ambitious position of claiming that "Quantum Computers are Musical Instruments", a "decomposed" and multi-faceted approach, guided by Smetakian Sound Plastics, might lead us to achieving music expression with "Quantum Musical Instruments".

Research Question 3: *How to put in practice the use of Quantum Computing in a Musical Artistic Process and/or Live Performance?* (**RQ3**)

In more practical terms, Scarassatti brings up the work by Arthur Ribeiro, who elencated/classified research practices of Marco Antônio Guimarães (a disciple of Smetak) on experimental luthery for the Uakti group (below), which, as pointed out by Scarassatti, also applies to Smetak. From that list, most practices will be applied in some form; with the exception of item 5.

- ***The discovery and the sonic investigation of new materials;***
- ***The research that originates from the artistic/musical impulse of a composer or group;***
- ***Research carried from a pre-selected sound result, through observation of physical phenomena;***
- ***The research that starts from the basic mechanisms of sound emission;***
- *The projection of a drawing, schematic, or floor plan;*
- ***Research that comes from the modification or adaptation of traditional instruments and their musical gesture techniques;***
- ***Research that recycle everyday materials into musical instruments;***[(†)] *(Scarassatti, 2008, p. 97)*

Given these points, a practice-as-research methodology based is outlined: **Investigating heuristic applications of Quantum Representation of Audio as *effects* for generating new sound materials for music composition (M4)**.

In this context, the approach was to apply the results from the technical scope (`quantumaudio`), using simple QRA manipulations that could be replicated by artists. Regardless of being optimal or not for QSP, this provided fertile ground for two artistic creations: First, an acousmatic piece that used mostly Musique Concrete techniques; Then, a Smetakian improvisation and artistic collaboration using Analog Synthesizers.

Using together the necessary technical and artistic processes, this investigation will lead us *towards quantum computing for audio and music expression*.

1.4 Structure of the Text

The body of this text is structured as follows:

- Chapter 1 - Introduction: The current chapter has focused on relevant philosophical concepts, motivations, research questions and interdisciplinary research methodologies used throughout the project.
- Chapter 2 - A Survey on Quantum Representations of Audio: Contains a systematic review of QRA, with a didactical presentation and proposed systematisation for classifying representations.
- Chapter 3 - The `quantumaudio` package: A technical description of the implementation structure as of version v0.0.2.
- Chapter 4 - Demonstrations: Early investigations of the package for processing audio and generating musical effects.
- Chapter 5 - Artistic Outputs: A description of the music pieces that resulted from using the effects of Chapter 4.
- Chapter 6 - Concluding Discussion: Personal Report on the artistic process and conclusions.

Chapter 2

A Survey on Quantum Representations of Audio

In the age of information, even our perceptual senses can be seen as machines that receive, process, and interpret incoming signals, such as sound and light. With the same angle, digital signal processing algorithms can interpret digital signals in various forms, providing tools for analysis (feature extraction) and synthesis (generation). The main distinction between signals (apart from their physical nature outside the digital realm) is with respect to dimensionality. For example, audio is a one-dimensional (1D) signal in the time domain, whereas images are two-dimensional (2D) signals in the displacement domain. Naturally, higher-dimensional signals should require more resources and often more complex mapping strategies to represent and process them. Therefore, a logical conclusion is that methodologies that are studied for simpler 1D (e.g., audio) signals can be later extrapolated to 2D (e.g., greyscale images), 3D (e.g., motion picture), and beyond. By and large, this is how textbooks on digital signal processing introduce the subject (Allen & Bryan, 2008).

While transitioning to a quantum-computing-driven research, it would seem reasonable to expect to find a vast literature on Quantum Representation of Audio (QRA) - and their unfoldings into potential signal processing algorithms - in comparison to more recent investigations on quantum images (QIR or QRI) and video. Surprisingly, this is not the case, and the very contrary is found. In fact, most quantum signal processing algorithms designed to date are for image applications. The first papers that describe how to represent an image on a quantum processor were published in 2003 (Venegas-Andraca & Bose, 2003) and 2005 (Venegas-Andraca, 2005). In comparison, the first papers specifying quantum representation methods for audio are from Wang (2016) and Yan et al. (2018); More importantly, they were based on existing methods for image representation, proposed by Zhang et al. (2013).

In other words, when the topic is Quantum Signal Processing, QRI (a 2D signal!) appears to lead with a decade-long headstart, leaving influences for Quantum Representation of Audio to catch up.

With this in mind, by drawing from literature review produced for QRI (Yan et al., 2016; Lisnichenko & Protasov, 2023b)), we propose a systematic presentation and classification focused on QRA for the adveturous computer musician.

2.1 The Quantum Audio Roadmap: From Mechanical to Quantum

In order to store a sound in a quantum computer (i.e., by the time of writing), one would need to record the sound (mechanical) as analog audio and make an analog-to-digital conversion. Then, the digital audio needs to be encoded into a quantum state that uniquely *represents* the signal (Figure 2.1).

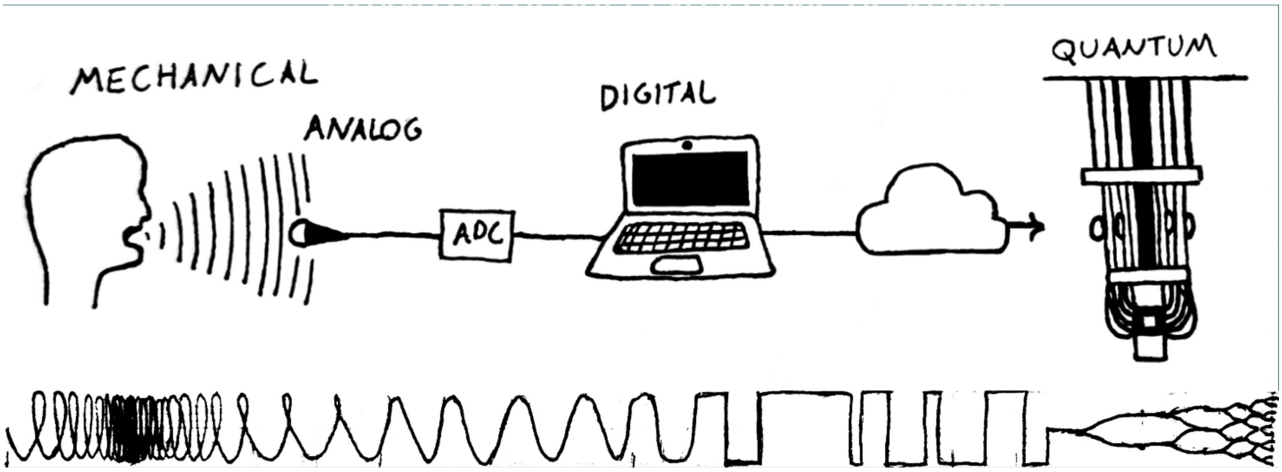


Figure 2.1: The Quantum Audio Roadmap

2.1.1 From Mechanical to Analog

In simple terms, sound is a mechanical wave that propagates mechanical disturbances through an acoustic medium. Tipically (and for our purposes), this manifests itself through longitudinal pressure variations in the air.

Upon reaching a dynamic microphone, the vibrating air pressure is *transduced* into varying electric voltage by a diaphragm coupled to an electromagnet (which induces the electric field). By definition, this is *analog audio*. This establishes a direct connection between mechanical and electrical media. Subsequently, we could say that analog audio *represents sound*.

The reason for including this section is to make it clear that analog audio is being used as a means of a *continuous, one-dimensional, and functional* portrayal of sound: Sound as a continuous function of time (eq. 2.1).

$$A = a(t) \tag{2.1}$$

Analog Audio Encoding

Attempts to reduce noise and errors during distant transmissions have long motivated more efficient audio encoding schemes. Therefore, analog audio is encoded by means of a measurable variation of a continuous *parameter* of another analog signal, such as the frequency, amplitude, or phase. Hence, the three most widely used analog encoding schemes are Amplitude Modulation (AM), Frequency Modulation (FM) and Phase Modulation (PM) (Ziemer & Tranter., 2014).

2.1.2 Analog to Digital

Contrarily, the digital medium reings and imposes the paradigm of discrete information and discontinuous signals. Thus, analog information needs to be digitised. This is done by means of Analog-to-Digital Conversion, which implies turning continuous time and amplitude information into discrete binary codes. This is realised through Sampling (discretization of the time domain) and Quantization (discretization of the signal amplitudes) [1].

Hence, the Digital Representation of Audio is a binary stream of associated amplitude/timestamp pairs (eq. 2.3). To bind respective pairs together, a dimensionless index n is introduced (eq.2.2) to enumerate steps in the sampled time progression. This leads to the introduction of the *sampling rate* S_R, which will determine the resolution of the sampling process, as well as the frequency range that can be accurately represented by this digital form (Allen & Bryan, 2008, p. 21).

$$n = S_R.t_n \tag{2.2}$$

$$A = (a_n, t_n);\;\; n \in \{0, 1, 2...N-1\} \tag{2.3}$$

Digital audio is commonly visualised as an array vector, as shown below (fig. 2.2). In other words, each slot contains an audio sample. The time information becomes an index that specifies a slot.

[1] Bear in mind that the notion of quantization here is nothing (and yet, everything) to do with quantum mechanics. Here, quantizing means *to restrict a continuous range to a prescribed set of values*.

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7

Figure 2.2: Audio array visualization.

Digital Encoding

Firstly, it is worth noting that the amplitudes a_n were represented as binary strings. There are a number of ways in which to represent and refine the resolution by using different binary forms (or types), namely, unsigned integers, signed (two's complement) integers, fixed and floating-point fractional numbers.

Secondly, to efficiently store and transmit digital data as a signal stream, a digital audio encoding framework called *Pulse Modulation* could be used. Similar to the analog version, it contains a carry signal - in this case, defined by a pulse train - controlled with its available parameters. For instance, there is Pulse Amplitude Modulation (PAM), Pulse Position Modulation (PPM), and Pulse Width Modulation (PWM)(Ziemer & Tranter., 2014, ch. 3-4). In a way, one can say that these strategies are structurally *hybrid*, as they are based on discrete time cycles (which satisfy time sampling), but the parameter mapping allows the encoding of *potentially* continuous amplitudes.

In contrast, Pulse Code Modulation (PCM) is a strictly digital protocol. As the name suggests, the binary string (a_n) is used to trigger or turn off pulses at each clock cycle.

2.1.3 From Digital to Quantum

A the end of the Roadmap, we discuss how a *quantum states* $|A\rangle$ can be interpreted as a signal and used as a representation space for digital audio.

As previously discussed, digital signals are arrays. Expressed differenly, they are essentially *a set of indexed information.* This index can also be seen as a piece of digital information used to point at the correct slots - a *pointer*. By analogy, the QRAs - and deductively quantum signals in general - will use a set of qubits (i.e. a qubit register) as pointers to index the amplitudes.

A pocket review of quantum information For those completely new to the concept of quantum computing that happen to fall into this chapter, a qubit is a two-state quantum system that can be measured into either 0 or 1. It can also be put in a superposition of states, written as shown in Eq. 2.4, where $|\alpha|^2$ and $|\beta|^2$ are the probabilities that a measurement of this state results in 0 or 1, respectively. Since the sum of the probabilities of all possible outcomes needs to be 1, it means that $|\alpha|^2 + |\beta|^2 = 1$.

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.4}$$

For multiple-qubit states, a similar equation can be written considering all possible outcomes, as shown in Eq. 2.5 for a 3-qubit system.

$$\begin{gathered} |\Psi\rangle = a\,|00\rangle + b\,|01\rangle + c\,|10\rangle + d\,|11\rangle + e\,|00\rangle + f\,|01\rangle + g\,|10\rangle + h\,|11\rangle\,; \\ |a|^2 + |b|^2 + |c|^2 + |d|^2 + |e|^2 + |f|^2 + |g|^2 + |h|^2 = 1 \end{gathered} \tag{2.5}$$

It is important to note that, in contrast to classical bits, the zeros and ones inside the 'kets' $|.\rangle$ *are not numbers*. The "Kets", are a simplified (Dirac) notation for writing *vectors*(2.6).

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad ; \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2.6}$$

Similarly, a multi-qubit state does not consist of a bit word, it is just a higher-dimensional vector, expressed in a tensor multiplication form (2.7).

$$|10\rangle = |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |2\rangle \tag{2.7}$$

Notice how anything written inside the ket, is just a conventionalised *label* that refers to a vector: $|\mathtt{label}\rangle$. We use numbers as labels to reduce the level of abstraction of those mathematical entities and provide some insight into their use inside a system. In other words, the interpretation above is not changing the state in mathematical terms, but provides a convenient visualisation.

Thus, by interpreting multi-qubit states as a binary expansion, we can write a 3-qubit state as the one shown in Eq. 2.8.

$$|\Psi\rangle_{3-\mathtt{qubit}} = \frac{1}{\sqrt{8}}\big[\,|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + |6\rangle + |7\rangle\,\big] \tag{2.8}$$

For a more robust introduction to quantum information, refer to (Nielsen & Chuang, 2002, p. 216-221),

Indexed Quantum Information

In summary, there would generally be two direct approaches to quantum indexing while in possession of a qubit register, discussed below. First, each individual qubit could be seen as it's own individual and independent slot that will point to an individual amplitude (Fig. 2.3a); this is often referred to as a Qubit Lattice Model (QLM) and is frequently found in Variational Quantum algorithms. Second, we could focus on simple quantum superpositions, as in (2.8), meaning that the qubit configuration will be interpreted as the index (Fig. 2.3b).

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7

$$\downarrow$$

$$|t_0\rangle \otimes |t_1\rangle \otimes |t_2\rangle \otimes |t_3\rangle \otimes |t_4\rangle \otimes |t_5\rangle \otimes |t_6\rangle \otimes |t_7\rangle$$

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7

(a) Quantum lattice indexing

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7

$$\downarrow$$

$$|t_0\rangle + |t_1\rangle + |t_2\rangle + |t_3\rangle + |t_4\rangle + |t_5\rangle + |t_6\rangle + |t_7\rangle$$

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7

(b) Simple superposition indexing

Figure 2.3: Digital to Quantum

By and large, the QRA and QRI share the same indexing strategy - Superposition; although an example of a Qubit Lattice image also exists in the literature (Venegas-Andraca & Bose, 2003; Venegas-Andraca, 2005). The method is to use a quantum register to create a superposition of all possible indices associated with each audio sample. This will be referred to as *time register* (t_{reg}) at most times in this text. Each state will be an index, indicating a position in time. This will be the focus of this survey.

2.2 Quantum Representations of Audio

Given these points presented in the last section, an initial understanding of Quantum Representation of Audio is straightforward, in the sense that Dirac's notation can clearly show how the audio information (time and amplitude) could be translated into a quantum state (Figure 2.3). There is a piece of information that is correlated to this (time) state that could potentially encode its respective sample. In summary, QRAs will differentiate themselves by using different strategies to make this binding between amplitude and index (Eq. 2.9).

$$\boxed{\text{Amplitude}}\,|t_{reg}\rangle \tag{2.9}$$

2.2.1 The anatomy of Quantum Audio Encoding/Decoding

In practical terms, having a multi-qubit quantum state that somehow describes an audio signal should also mean there is a protocol, containing a quantum circuit able to *prepare* (encode) that state. Moreover, this protocol should also provide a strategy to *reconstruct* (decode) the results of a quantum experiment back to audio.

It is worth mentioning that, in order to retrieve the information stored in the quantum machine,

To better organise and distinguish QRAs from each other, a three-layer category scheme is proposed:

Types: Coefficient vs State (Continuous vs Discrete)

In terms of encoding digital amplitude information (while sharing this common root for time encoding), Yan et al. (2016) groups different QRIs into two different classes. Their difference lies on the general *type* of parameter used to encode the information (in other words, the encoding strategy), resulting in distinct properties of the retrieval/decoding protocols. In this text, they are named as Coefficient-Based (CB) and State-Based (SB)[2]. CB methods are probabilistic in nature, which means that they focus on the probability amplitudes of quantum states to encode audio information. In contrast, SB methods use multi-qubit states to represent audio information as binary strings correlated with the time register. This distinction will become clearer along the text, as the different strategies are presented.

Protocols/Models/Schemes

At this level, the representations are further divided into the general model (or models) used as the base mapping strategy. This specifies *which* parameter is used (e.g. angles, amplitudes, qubit register configurations, phases, etc.).

Variants

This category is useful when there are different *interpretations* or slight adaptations of a given parameter. For instance, binary strings might be seen as integers or floating-point numbers, or a number might be presented in cartesian or polar form. Nevertheless, the main structure is virtually preserved.

[2]In other words, one class looks at the *eigenvalues* of the quantum vector, whereas the other uses the *eigenstates*.

2.2.2 Parenthesis: Terminology Index

This survey proposes a new naming system for QRA, based on the word 'Modulation', bringing together paralells with their classical counterparts. There are a number of motivations for taking this action (even if it only ends up bringing additional definition fog). This proposal is motivated by the intention of a more unified integration and mathematical notation between quantum and classical representations/encoding strategies.

Following this logic, the remaining part of the acronym would contain useful information on where in the system the amplitude sample was encoded. For instance, a State-Based representation like FRQA (Yan et al., 2018) would be naturally translated to "Quantum State Modulation", (or QSM) in hope of a clearer definition. For a reference, see Table 2.1.

Table 2.1: Quantum audio terminology index

In this Text	QRA as found in the literature	QRI as found in the literature
QPAM	-	Real-Ket
SQPAM	-	FRQI
QSM	FRQA,QRDA,QRDS	NEQR
MQSM	QRMA,CQRDS	MCQI
MSQPAM	PMQA	-
$SQPAM^2$	-	OQIM
TQPAM	-	QSMC & QSNC
PSQPAM	-	IFRQI
QPPM	-	FTQR
SQPPM	-	-

2.3 Coefficient-Based Representations

When exposed to the superposition of Eq. 2.8, one could induce that a natural audio encoding choice would be to create a superposition of all possible states t (encoding time). Therefore, each time state would be weighted by a respective probability amplitude (encoding the sample value) in a way that resembles an array (Eq. 2.10).

$$\boxed{a_n}\; t_n \longrightarrow \alpha_i \left|t_i\right\rangle \tag{2.10}$$

2.3.1 A simple representation: Quantum Probability Amplitude Modulation: QPAM

More generally, for an arbitrary audio of size N, with n (or $\lceil \log N \rceil$) qubits, we build an encoding scheme where each possible amplitude value of the audio is mapped onto a probability amplitude

(Eq. 2.11). Equation 2.12 shows an expanded example for an audio signal of size 8.

$$|A_{QPAM}\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle \tag{2.11}$$

$$|A\rangle_{(3-qubit)} = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \alpha_2 |2\rangle + \alpha_3 |3\rangle + \alpha_4 |4\rangle + \alpha_5 |5\rangle + \alpha_6 |6\rangle + \alpha_7 |7\rangle \tag{2.12}$$

In other words, the audio amplitudes a_i, in this case, are directly encoded as (real-valued) probability amplitudes α_i associated to the measurement of the state $|i\rangle$, which is interpreted as an index written as a binary number. The conversion between a_i and α_i is shown in Eq. 2.13.

$$\alpha_i = \frac{1}{\sum_k \frac{(a_k+1)}{2}} \sqrt{\frac{(a_i+1)}{2}} \tag{2.13}$$

Put differently, Eq. 2.13 is *normalizing* the signal, i.e., making the squared sum of all samples equal to 1.

QPAM Preparation

The QPAM representation has a practical preparation strategy. As the information is stored in the probability amplitudes, we just need to bring the qubits - initially at a resting ground position |000...⟩ - to the desired quantum superposition by providing the coefficients α_i. Most available quantum programming tools provide commands to initialise arbitrary qubit states. Under the hood, this is realised by using U and Toffolli gates (Plesch & Brukner, 2011) to rotate these qubits around the valid configurations. So, it suffices here to assume that any arbitrary superposition of states can be prepared by providing a set of probability amplitudes to a *initialise gate* (Figure 2.4). Consequently, for preparing a QPAM quantum audio, we would only need to convert the digital samples into probability amplitudes.

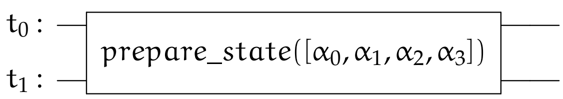


Figure 2.4: Initializing qubit states for QPAM.

Using the quantum state preparation algorithm to reach an arbitrary n-qubit state usually requires $\mathcal{O}(n)$ simple operations (Zhang et al., 2022), which implies that QPAM preparation needs $\mathcal{O}(\lceil \log N \rceil)$

operations for a signal with N samples.

QPAM Retrieval

The Coefficient-Based representations share a common property: Their retrieval strategy is inherently *Probabilistic*. Retrieving audio requires that we prepare many identical versions of the quantum audio state and perform a statistical analysis of the result over several measurements. The analysis will return a histogram of the experiment (e.g., Fig. 2.5), from which we can assess an approximate probability of measuring each state. When considering how the system was prepared in the first place, it is noted that the histogram itself is already a scaled and shifted version of a digital audio. To rescale the audio back to its original range, we need to use something close to the inverted version of Eq. (2.13).

Retrieving information from the quantum domain means taking measurements from our quantum system. However, measuring a quantum system has its own limitations. For visualisation/analogy purposes, we can imagine a photographer (hello again, Flusser) that can only take 2D "pictures" of a moving 3D object from a fixed position (in physics terminology, this is called *projective measurement*). The more photographs are taken, the better the reconstruction of the shape of the 3D object will be. This notion was highlighted here because it will be useful when discussing artistic applications later (Section 4.3.1).

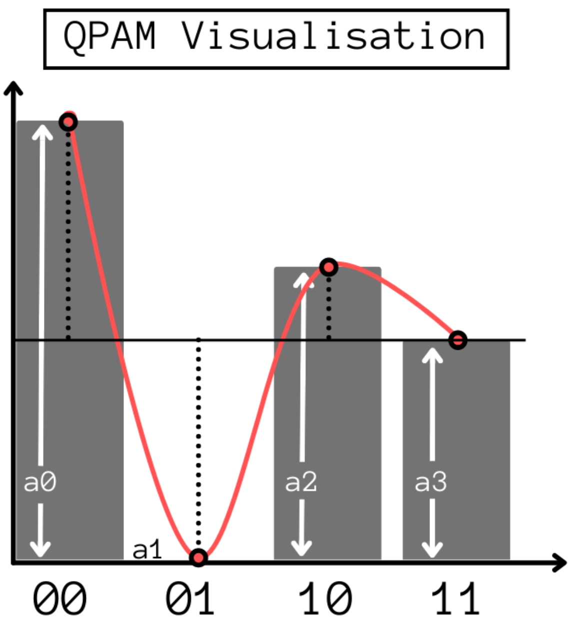


Figure 2.5: A hypothetical QPAM representation.

Attention to the Normalisation

There is a caveat with the normalisation in eq. 2.13 and its proposed inverse. First, when the encoded signal is -1. Second, when the amplitude of the signal is low.

As a result, retrieval normalisation becomes a more arbitrary choice. Depending on the application, the it needs to be clear that the amplitude range of the signal is compressed, an any information on the ampitude range of the original signal is *lost*; unless this information is stored classically.

2.3.2 Single-Qubit Probability Amplitude Modulation: SQPAM

Another coefficient-based QRA Model can be adapted from an image representation scheme known as Flexible Representation of Quantum Images (FRQI) (Le et al., 2011; Yan et al., 2016). The audio translation for FRQI is given in equation 2.14, under the name Single-Qubit Probability Amplitude Modulation (SQPAM).

$$|A_{SQPAM}\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (\cos\theta_i |0\rangle + \sin\theta_i |1\rangle) \otimes |i\rangle \tag{2.14}$$

SQPAM also works with real-valued probability amplitudes. However, instead of using the coefficients of t_{reg}, it improves the logic to encode the samples in the probability amplitudes of one extra, *dedicated* qubit, added as a new register, which will be baptised as amplitude register (a_{reg}). As a consequence, it uses $\lceil \log N \rceil + 1$ qubits. Hence, the name assigned to this representation was "Single Qubit Probability Amplitude Modulation", or SQPAM.

An expanded version of Eq. 2.14 for an arbitrary signal with length $N = 8$ is shown in 2.15.

$$\begin{aligned} |A\rangle_{3-qubit} = \frac{1}{\sqrt{8}} \Big[&(\cos\theta_0 |0\rangle + \sin\theta_0 |1\rangle) \otimes |0\rangle + (\cos\theta_1 |0\rangle + \sin\theta_1 |1\rangle) \otimes |1\rangle + \\ &(\cos\theta_2 |0\rangle + \sin\theta_2 |1\rangle) \otimes |2\rangle + (\cos\theta_3 |0\rangle + \sin\theta_3 |1\rangle) \otimes |3\rangle + \\ &(\cos\theta_4 |0\rangle + \sin\theta_4 |1\rangle) \otimes |4\rangle + (\cos\theta_5 |0\rangle + \sin\theta_5 |1\rangle) \otimes |5\rangle + \\ &(\cos\theta_6 |0\rangle + \sin\theta_6 |1\rangle) \otimes |6\rangle + (\cos\theta_7 |0\rangle + \sin\theta_7 |1\rangle) \otimes |7\rangle \Big] \end{aligned} \tag{2.15}$$

Mapping the SQPAM Audio Amplitudes

The probability coefficients of a_{reg} on this representation are written in trigonometric form; this suggests that its preparation strategy will contain some type of rotation [3].

The function that maps the audio amplitudes a_i to the SQPAM angles θ_i is displayed in eq. 2.16. Note that θ_i is duplicated in their respective trigonometric pairs (cosine and sine) of each term in Eq. 2.15. Subsequently, mappings using cos^{-1} or sin^{-1} are equivallent, as long as the decoding process follows the same convention. In this text, the audio will be encoded using the $sine$ convention.

$$\theta_i = \sin^{-1}\left(\sqrt{\frac{a_i + 1}{2}}\right) \tag{2.16}$$

SQPAM Preparation

The SQPAM preparation algorithm will be a Value-Setting Operation (VSO) (see the next section for details - 2.3.3) that encodes θ_i by rotating the a_{reg} qubit to the desired angle. The gate used to realise this rotation is $R_Y(2\theta)$; after inspecting its matricial form, R_Y turns out to be, quite literally, a 2D rotation matrix (Eq. 2.17) A typical quantum circuit for preparing an SQPAM state is shown in Fig. 2.6.

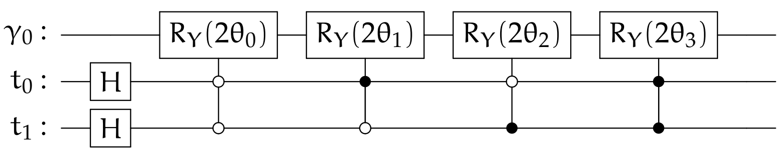


Figure 2.6: Preparation of a SQPAM representation using Value-Setting operations with R_Y gates

The operator being controlled, and is responsible for encoding the amplitudes according to the SQPAM scheme. By looping through all time indexes ('`for`' loop), the SQPAM state is prepared.

$$R_Y(2\theta) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad ; \quad R_Y(2\theta_i)|0\rangle = \cos\theta_i|0\rangle + \sin\theta_i|1\rangle \tag{2.17}$$

Compared (and adapted) to its image counterpart, this preparation process requires $\mathcal{O}(N^2)$ operations for a N-sized audio. The current complexity of this algorithm is caused by the multi-controlled operations, which are computationally expensive (as explained in sec. 2.3.3).

[3]Trigonometric functions are often used to represent probability amplitudes since their absolute value ranges from 0 to 1, and the relation between the cosine and sine functions satisfy the normalization requirement ($\cos(\theta)^2 + \sin(\theta)^2 = 1$).

Retrieval

Figure 2.7 shows what to expect from an SQPAM retrieval histogram. It contains twice as many bins compared to QPAM. Each consecutive pair contains information of one sample. This represents the $sine$ and $cosine$ terms of a specific index. In other words, when all qubits are measured, the probabilities of a_{reg} will be *complementary* (2.18). In our convention, it is possible to notice the original signal profile by looking at the even bins of the histogram, marked in red. Eqs. 2.18 and 2.19 are used to reconstruct the signal (γ_i denotes a pair of bins used to encode the ith sample, and p_{γ_i} refers to the probabilities of the amplitude qubit at a given index).

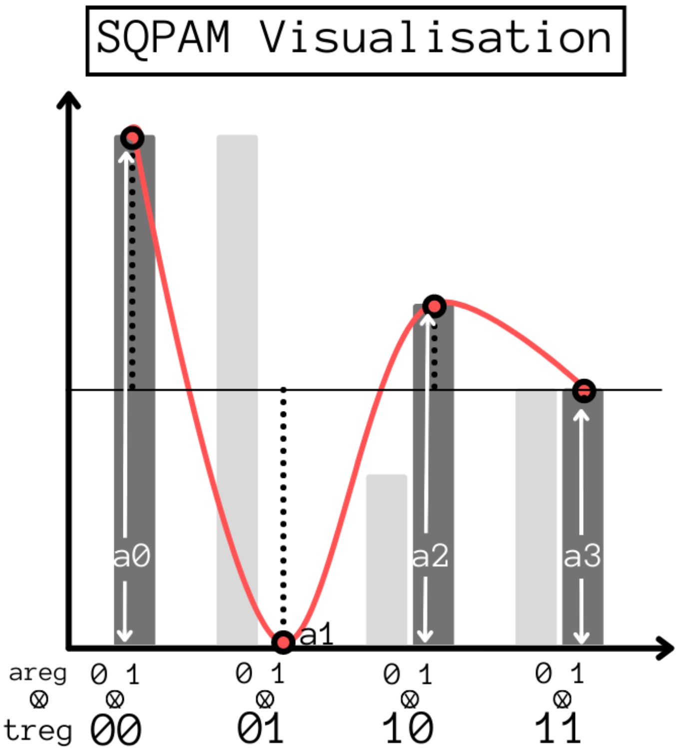


Figure 2.7: Hypothetical SQPAM representation

$$p_{\gamma_i}(|0\rangle) = (\cos(\theta_i))^2 \qquad ; \qquad p_{\gamma_i}(|1\rangle) = (\sin(\theta_i))^2 \tag{2.18}$$

$$a_i = \frac{2p_{\gamma_i}(|1\rangle)}{p_{\gamma_i}(|0\rangle) + p_{\gamma_i}(|1\rangle)} - 1 \tag{2.19}$$

According to Yan et al. (2016), the introduction of an extra qubit $|a_{reg}\rangle$ significantly improves the *flexibility* of the representation in terms of both encoding and retrieval of the audio. The first improvement is that the encoding scheme no longer relies on the entirety of the audio size content - the total number of states on the superposition and their relative amplitudes - anymore. It is encoded "locally" in the new qubit and, therefore, could be independently manipulated, by rotating the desired sample using a multi-controlled gate strategy. This opens up a new range of possibilities for sample-specific operations 2.10.

2.3.3 Value-Setting Operation

With the addition of a_{reg}, the structure behind the preparation algorithm will mainly consist of a series of Value-Setting Operations. In summary, a Value-Setting operation will correlate (or 'entangle') a specific state of t_{reg} with a specific configuration of a_{reg} - representing the respective sample. In fact, most representations that contain indexed information (i.e., using the t_{reg} as defined in Sec. 2.1.3) will be prepared with some form of VSO.

A Value-Setting Operation involves a *multi-controlled gate*. The control part is used to select a specific configuration(index) of t_{reg}. Only when this condition is met is the corresponding gate carried by the encoding part employed. The architecture of a VSO is shown below in Fig. 2.8, while a case for SQPAM was seen in Fig. 2.6. The control condition is denoted by either a black (for testing $t_{reg}^{(n)} == |1\rangle$) or a white (for $t_{reg}^{(n)} == |0\rangle$) dot. The "$Set$ gates in Fig. 2.8 indicates where different QRA employ their own gates or algorithms to prepare the quantum audio state.

By using a VSO for each desired index, there will be a direct correlation between a time (in t_{reg}) and an amplitude (in a_{reg}). For readers familiar with computer science, this algorithm could be framed as a 'quantum' version of a sequence of *if-else* statements with variable assignments. The only difference will be that the multi-controlled operation would - in principle - perform the conditioning test for all t_{reg} qubits simultanously, whereas the classical counterpart will need to verify each bit individually. In practice, however, there will be additional circuit complexity involved, since larger multi-controlled gates will need to be decomposed in terms of valid QPU operators, such as CNOTs, and Toffoli gates (Fig. 2.9, (Nielsen & Chuang, 2002, ch. 4)).

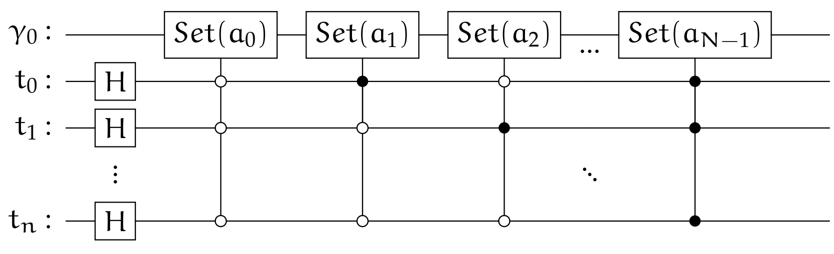


Figure 2.8: Value-Setting Operation Visualisation

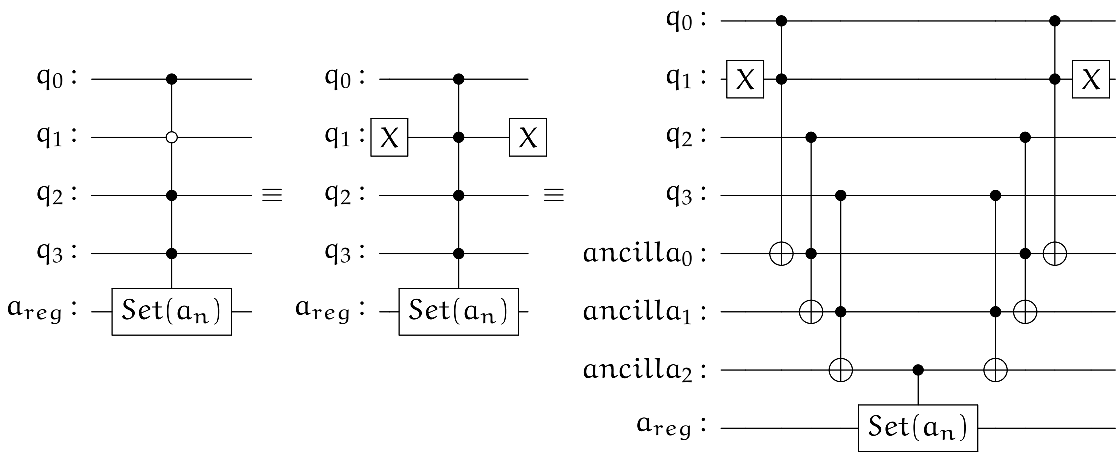


Figure 2.9: CCCC-Set(a_n) example and its decomposition

2.3.4 SQPAM - Different Variants

In summary, in sec. 2.3.2, the SQPAM model (e.g. encoding amplitude in a single qubit) has used a trigonometric form to encode the audio into angles θ_i. The retrieval process leads to a pair of complementary probabilities related to the binary form of a_{reg}. However, this is not the *only* possible way of defining SQPAM. In Lisnichenko & Protasov (2023b), it is possible to identify different single-qubit encoding approaches that are effectively analogous to the definition in (2.14).

For that reason, the proposed classification structure for QRAs puts all single-qubit encodings grouped together inside SQPAM. Then, the representation exposed in the last section would be a *trigonometric variant* of SQPAM.

NOTE: From this point owards, if a *Variant* needs to be specified or made clear in the text, there will be a hyphen and a set of up to two letters added to the acronym, as in SQPAM-t (Trigonometric) and SQPAM-c (Cartesian).

SQPAM-c

Inspired by Liu et al. (2018), it is possible to describe a SQPAM in a "rectangular" or "cartesian" form (2.20); although the result is mathematically equivalent, having multiple forms could lead to potential algorithms that benefit from combining both forms, as the one studied in Dang et al. (2018).

$$|A_{SQPAM-c}\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (\sqrt{1-c_i^2}\,|0\rangle + c_i\,|1\rangle) \otimes |i\rangle \tag{2.20}$$

$$c_i = \sqrt{\frac{a_i+1}{2}} \;; \qquad \text{(VSO)} \rightarrow R_Y(2\sin^{-1} c_i) \tag{2.21}$$

2.4 State-Oriented Representations

In the previous section, the quantum audio representation schemes stored the time domain into a superposition of multi-qubit states and the amplitudes parametrised as a coefficient. In this section, the main difference is that the audio amplitudes will also be stored in a multi-qubit state. This updated $\mathtt{a}_{\mathtt{reg}}$ will produce a set of time-amplitude pairs of *entangled configurations* when combined with $\mathtt{t}_{\mathtt{reg}}$.

This paradigm was presented at a later stage in the QRI literature. For Zhang et al. (2013), it can be seen as an *improvement* from from the previous representations due to its more computationally efficient state preparation process. As the reader may conclude from this section, there is a strong appeal to state-based representations, since they seem intuitively related to their digital counterparts. As a result, there are more straightforward ways to translate and benchmark the known DSP with state-based Quantum Signal Processing (Yan et al., 2016) - which supports Zhang et al.'s argument. Ultimately, each application could benefit more from a different QRA, and both should be equally preserved.

Additionally, state-based can *potentially* be manipulated in more detail (accessing amplitudes as binary information). A consequence of these "finer" capabilities is that they might increase the amount of space required to represent and manipulate the audio. In other words, there seems to be some kind of trade-off between the number of qubits required to store a signal and the amount of potential applications (Yan et al., 2023).

2.4.1 Quantum State Modulation

The following quantum audio representations are derivations of the Novel Enhanced Quantum Representation for digital images (NEQR) (Zhang et al., 2013). As mentioned, the information is encoded whithin the multi-qubit states, instead of using other qubit parameters; this motivates its name: Quantum State Modulation (QSM).

Recalling the QRA literature, the first proposed representations (Wang, 2016; Yan et al., 2018) were both based on NEQR [4]. The earlier, Quantum REpresentation of Digital Audio (QRDA) has a dedicated quantum register that encodes the sample using strictly positive integers (directly inherited from the NEQR Image that encoded grey colour information ranging from 0 to 1). The Flexible Representation of Quantum Audio (FRQA) on the other hand, corrects this and uses the Two's Complement scheme

[4] An unsuspecting reader might assume that FRQA was based from the previously discussed FRQI, which is, unfortunately, **not** the case.

to interpret the amplitudes. The signed integer makes audio mixing extremely convenient since only a regular bit-wise adder algorithm is needed (See sec. 2.10.1). This indicates that the FRQA's interpretation of the states is more suitable for quantum audio than QRDA.

In other words, both QRDA and FRQA are practically identical from a mathematical standpoint, but there is a slight difference in the interpretation of the qubit states. Respectively, they are read as either *unsigned* or *signed* integers. As expected, this perfectly fits our classification as two different *variants*.

Later, Sahin (Li et al., 2018) has generalised the concept of NEQR as a Quantum Representation of Digital Signals (QRDS), which includes audio and higher-dimensional signals. This is a similar attempt of grouping multiple proposed representations together. For all purposes, the reader can consider that Li et al.'s QRDS and "Quantum State Modulation" defined here carry the same meaning; just compiled within a larger classification structure and audio-biased terminology.

For convention and commonality, signed integers are more commonly used to encode audio information in the classical domain (PCM audio). As a result, the main variant discussed will be the one proposed by Yan - FRQA - and denoted as both QSM and QSM-s. Whenever a specific application explicitly requires unsigned integers (QRDA), this could be indicated with a lowercase "u" (QSM-u). Other variants found in the literature interpret a_{reg} as fixed-point QSM-fp (Li et al., 2018) and floating-point QSM-fl (Chaharlang et al., 2020b) numbers.

2.4.2 QSM

Unlike its coefficient-based counterparts, the QSM encodes amplitudes similarly to digital audio - as binary words. Therefore, the quantization depth (bit depth) 'q' of the audio will determine the space requirements of the amplitude register.

Consider a digital audio file with $N = 8$ samples and amplitude values quantized in a 3-bit depth two's complement scheme. Imagine that the seventh sample of the signal was represented by a given integer bit string "001". In possession of this string, a quantum circuit would be designed to write this information on a 3-qubit register $|001\rangle$. This quantum state will later be entangled with a respective index of t_{reg} - in this case, t_7. as described in Eq. 2.22.

$$(a_7, t_7) \equiv (''001'', ''110'') \longrightarrow |001\rangle\,|110\rangle \equiv |a_7\rangle \otimes |t_7\rangle \tag{2.22}$$

Applying this structure to all the samples would result in the superposition shown in equation 2.23. More generally, QSM could be written as in Eq. 2.24.

$$|A_{QSM}\rangle = \frac{1}{\sqrt{8}}\Big[|000\rangle \otimes |000\rangle \ + \ |111\rangle \otimes |001\rangle \ + \ |010\rangle \otimes |010\rangle \ + \\ |011\rangle \otimes |011\rangle \ + \ |110\rangle \otimes |100\rangle \ + \ |101\rangle \otimes |101\rangle \ + \\ |001\rangle \otimes |110\rangle \ + \ |000\rangle \otimes |111\rangle\Big] \tag{2.23}$$

$$|A\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |S_i\rangle \otimes |i\rangle \tag{2.24}$$

Preparation

The same strategy mentioned in the SQPAM case will be used here: a Value-Setting Operation (VSO). In this case, instead of using $R_Y(2\theta_i)$, the operation will use multi-controlled CNOT gates to flip the qubit states of the amplitude register. As a result, the algorithm will access and entangle specific amplitude states with their respective time states.

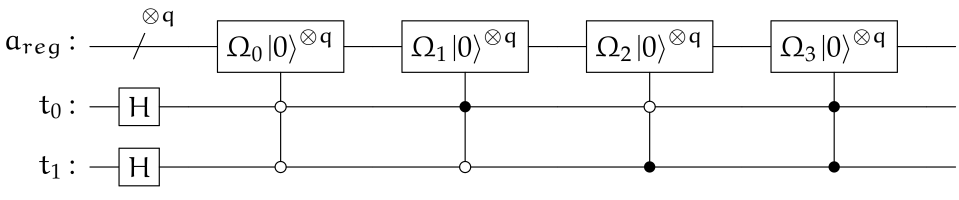


Figure 2.10: Preparation of a QSM representation using Value-Setting operations

This VSO controlled gate consists of a *quantum oracle*. In other words, the gate "knows" every bit word that is going to be encoded for a given index. The CNOT gate is only applied when the respective classical bit is 1 (Yan et al., 2018). This is indicated in (2.25).

$$\Omega_k |0\rangle = \bigotimes_{\ell=0}^{q-1} \left(\Omega_k^{\ell} |0\rangle\right) = \bigotimes_{\ell=0}^{q-1} |0 \oplus B_k^{\ell}\rangle \tag{2.25}$$

Where B_k^{ℓ} denotes the ℓ-th bit in the binary sequence at time k

Compared to SQPAM ($R_Y(2\theta_i)$), the preparing circuit requires $\mathcal{O}(q + \log N)$ basic instructions. In conclusion, the preparation circuit for the QSM is exponentially faster. According to Yan, it requires $\mathcal{O}(qN \log N)$ simple operations .

Retrieval

The interesting part of retrieving state-based quantum audio is that it can be seen as *deterministic*. It can, in principle, guarantee a *perfect* reconstruction of the signal. This is guaranteed since the sample is represented as a binary state.

In the Coefficient-Based version, the samples were statistical quantities that needed to be approximated through many measurements and statistical analysis. There will always be an intrinsic statistical error whenever the amount of measurements is finite. It is a trade-off between lying an error inside some tolerance range (depending on musical, psychoacoustic, perceptual factors) and time consumption.

In the case of QSM retrieval, we would only need to ensure that all *valid* time-amplitude pairs are measured to retrieve the audio. For instance, in eq. 2.24, each t_{reg} state is entangled to an a_{reg} state. Thus, a given time state (e.g. from eq. 2.24) would certainly be measured with its correlated amplitude; without the presence of noise, there would be no other possible states to be measured. Put another way, in an ideal situation, a state-modulated representation will have a balanced superposition of *only* the states containing audio information (Fig. 2.11). Moreover, the measured states are then decoded into the correspondent digital amplitudes using the Two's Complement scheme.

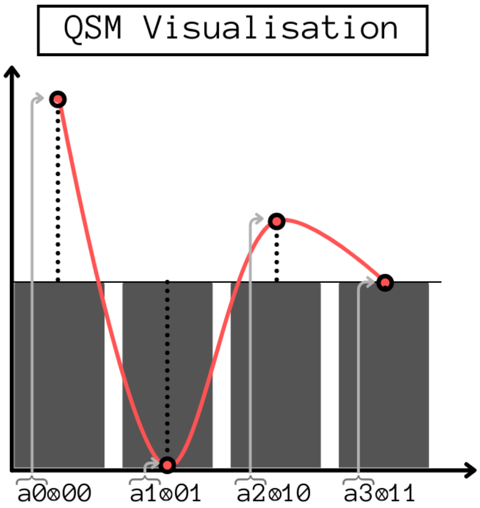


Figure 2.11: Hypothetical QSM histogram

2.5 Revisiting Time Encoding

At this point, all proposed (single-channel) representations of audio have been exposed. Now it is important to once again involve Lisnichenko & Protasov (2023b) in the discussion, drawing more intuition around audio representations and provide pathways for new ones. While listing a vast amount of proposed QRIs, they also arrive at a possible (and useful) classification scheme to separate them; it accounts for how both amplitude *and time* are being encoded. But why? In the QRI literature, due to its 2-dimensional features, it is important to keep track of two variables (x, y) that form the coordinates of a pixel. In that regard, there are varied efforts to define mappings that contain one index that will point to a location in the image matrix. Additionally, there is a desire for encoding images of arbitrary sizes (so far, the superposition indexing has restricted us to signals from which sizes are powers of two). This leads to a discussion framework that *separates* the index from the coordinate. Although this mapping can be performed classically (e.g. a dictionary with index-location pairs), some representations explicitly allocate dedicated registers to encode the location of the qubit (Xu et al., 2019). Further, these QRI often use coefficient-based encoding to do so. This motivates a naming system that separates these cases from, say, FRQI.

First, their approach will be briefly outlined, and then it will be showed where our definitions might diverge and how this project addresses the categorization.

In simple terms, Lisnichenko & Protasov (2023b) describe their representation classes not by focusing on the nature of the paramenter used (e.g., a coefficient, a quantum state, a configuration) but rather by expressing the the parameter's domain type (e.g., continuous or discrete). As a result, the QRI were classified as listed below:

- Pure Discrete Representations - Uses state-based strategies to encode both pixel intensity and coordinates (NEQR, QUALPI, GQIR, etc.).
- Pure Continuous Representations - Uses coefficient-based strategies to encode both intensity and coordinates (QSMC & QSNC, NAQSS, OQIM).
- Mixed - When there is a combination of both, or when coordinate encoding is not clearly specified. (FRQI, Qubit Lattice, SQR, Real Ket, etc.)

This system proves to be an efficient way to distinguish between representations. Nevertheless, it presents some limitations and rough corners that could be improved and sanded down.

For example, while the three strategies under the Pure Continuous catagory use single-qubit amplitude encoding, they seem to approach this in different ways, requiring a better specific distinction. Similarly, looking at the explored representations, it is possible to argue that FRQI (our SQPAM), and Real Ket (our QPAM) are radically different from SQR and Qubit Lattice when it comes to preparation circuits, retrieval, and potential applications. A possible conclusion would be that grouping strategies that are mixed or unspecified coordinate encoding could lead to a group that is *too large*. Moreover, there is a cited representation, FTQR, which was given its own individual class, even though it arguably shares affinities and proximities to the Real-Ket representation (see Sec. 2.8.1).

By introducing a spotlight at the discussion around Indexed Information (sec. 2.1.3), it might be possible to rearrange the mentioned conflicts into a new organisation, elencated below.

1. A Quantum Signal is a set of indexed information.
2. The main information elements to be indexed are part of the *image* $f(x)$ of the signal (e.g. amplitudes, intensities, voltages)
3. Optionally, the *domain* x of the signal (e.g. time, coordinate, displacement) are also information elements that could be indexed.
4. What globally distinguishes quantum signals apart is their indexing strategy, such as QLM, Simple Superposition, and others.
5. What locally distinguishes quantum signals apart is their encoding strategies for each element of information, either continuous (coefficient-based) or discrete (state/configuration based).
6. *Continuous/Pure Continuous* representations use only coefficent-based models to encode all elements, regardless of the indexing strategy.
7. *Discrete/Pure Discrete* representations use only state-based models to encode all elements, regardless of the indexing strategy.
8. *Mixed* representations use a combination of discrete and continuous approaches for different elements.
9. The *(n-)fold* of a quantum representation is defined by the number *n* of distinct information elements (i.e., quantum registers) that were indexed.
10. The *dimensions* of the quantum signal will depend directly on the interpretation of the types of each information element (either domain or image).
11. Different elements of information could be indexed differently, leading to *hybrid* representation models.

Now, let us analyse the meaning and consequences of defining a classification system this way. Firstly, there is a separation between 'the information' and 'the index'. There is a suggested analogy with the classical case: The indexing would deteremine the *data structure*[5] of the system. In other words, what are the data slots are like, and how they are organised. The encoding strategies discussed (QPAM, SQPAM, QSM) deal with the information inside the slot.

[5]This does NOT *necessarily* mean that it carries the same semantic importance or optimization properties observed in (classical) computer science

As a result, taking the QRI space to exemplify, the FRQI, and Real-Ket - which use superposition indexing - would be separated from Qubit Lattice (sic) and SQM (A Qubit Lattice Image Variant) - which use lattice model indexing. In similar terms, NAQSS is discriminated from the other two.

The other important outcome emerges from item 9: *fold*. This creates a parallel hierarchy that might help to separate representations for future comparative analysis. The more elements (meaning quantum registers), the higher the n-fold. This implies that representations that allocate dedicated registers for signal domain information (such as coordinate location or time) are put in a different category than those that do not.

Looking back, sections 2.3 and 2.4 describe models of 0-fold (QPAM) and 1-fold (SQPAM, QSM). In more detail, QPAM can be seen as the very definition of quantum superposition indexing, with nothing else. SQPAM and QSM, on the other hand, do have a distinction between a_{reg} (the information element) and t_{reg} (the index/pointer).

Next, in sections 2.6 and 2.7, we will explore 2-fold representations.

2.6 Multichannel Audio

Needless to say, the music industry is embedded in multichannel audio signals, supported by sample packs, sound and filter banks, and stereo music. For encoding multiple signals, we need to introduce a new information element - the audio channel - leading to a new quantum register: the *channel register* *(c_{reg})*.

Building up from the standardized superposition indexing, we are left with a 2-fold anatomy (2.26)

$$\boxed{\text{Channel}}\boxed{\text{Amplitude}}\,|t_{reg}\rangle \tag{2.26}$$

2.6.1 MQSM

From the QRA literature, two multichannel representations for audio that were encountered - QRMA (Şahin & Yilmaz, 2019)&CQRDS(Chaharlang et al., 2020b)) - are discrete, which makes them fit as two separate variants of a Multichannel Quantum state Modulation (MQSM) model(2.28). In other words, both c_{reg} and a_{reg} are encoded using a QSM-like strategy, using t_{reg} as their index (2.27).

$$(a_{n,m}, t_{n,m}, c_m) \longrightarrow |c_j\rangle \otimes |a_{i,j}\rangle \otimes |t_{i,j}\rangle \tag{2.27}$$

$$|A_{MQSM}\rangle = \frac{1}{\sqrt{NC}} \sum_{i=0}^{N-1} \sum_{j=0}^{C-1} |c_j\rangle \otimes |a_{i,j}\rangle \otimes |t_{i,j}\rangle \tag{2.28}$$

Stereo audio would be encoded as follows:

$$|A_{Stereo}\rangle = \frac{1}{\sqrt{2N}} \begin{bmatrix} |L\rangle \otimes \left(|a_{0,L}\rangle |0\rangle + |a_{1,L}\rangle |1\rangle + |a_{2,L}\rangle |2\rangle + |a_{3,L}\rangle |3\rangle + ...\right) \\ |R\rangle \otimes \left(|a_{0,R}\rangle |0\rangle + |a_{1,R}\rangle |1\rangle + |a_{2,R}\rangle |2\rangle + |a_{3,R}\rangle |3\rangle + ...\right) \end{bmatrix} \tag{2.29}$$

2.6.2 MSQPAM

Correspondingly, a channel register could be added to Coefficient-Based representations. In Itaboraí & Miranda (2022), a possible multichannel version of SQPAM was mentioned (MSQPAM), using c_{reg} with state-based encoding (2.30). After the book chapter was submitted for review, I found that this representation was formally proposed and analysed by Yan et al. (2022), under the name Probability amplitude-encoded Multichannel representation for Quantum Audio signals (PMQA). This citation did not appear in the book, but is now recognised here.

2.30.

$$|A_{MSQPAM}\rangle = \frac{1}{\sqrt{NC}} \sum_{i=0}^{N-1} \sum_{j=0}^{C-1} |c_j\rangle \otimes (\cos\theta_i |0\rangle + \sin\theta_i |1\rangle) \otimes |i\rangle \tag{2.30}$$

As exemplified in section 2.10.2, MSQPAM this representation can become a valuable tool for quantum-paralellised feature extraction algorithms.

2.7 Other 2-fold Models

Two notable 2-fold representations for images found in the literature mention coordinate encoding in a separate register, using coeficient-based strategies - QSMC & QSNC (Li et al., 2013) and OQIM (Xu et al., 2019). Their translated audio counterparts are enunciated below.

2.7.1 TQPAM

This 2-fold variant of SQPAM has time information encoded into probability amplitudes of an additional dedicated qubit with a segmented range (2.31). As a result, there is an indexed qubit pair comprising amplitude and time information. Hence, Two-Qubit Probability Amplitude Modulation

(TQPAM)

$$|A_{TQPAM}\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} (\cos\theta_k |0\rangle + \sin\theta_k |1\rangle) \otimes (\cos\tau_k |0\rangle + \sin\tau_k |1\rangle) \otimes |k\rangle \tag{2.31}$$

$$\theta_k = \sin^{-1}\left(\sqrt{\frac{a_k+1}{2}}\right) \tag{2.32}$$

$$\tau_k = \sin^{-1}\left(\frac{\pi k}{2(N-1)}\right) \ ; \ k \in \{0,1,2,...,N-1\} \tag{2.33}$$

2.7.2 SQPAM^2

Another 2-fold representation based on SQPAM takes a different turn. It genuinely uses a *single* qubit to encode both time and amplitude. In that case, it distingushes between time and amplitude by using an additional *pointer* (2.34).

$$|A_{SQPAM^2}\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \big[(\cos\theta_k |0\rangle + \sin\theta_k |1\rangle) |0\rangle + (\cos\tau_k |0\rangle + \sin\tau_k |1\rangle) |1\rangle \big] \otimes |k\rangle \tag{2.34}$$

Notice that, effectively, this representation is a variant of MSQPAM. The difference is in the state interpretation, and hence the signal dimension. In other words, one of the channels is used for the domain of the signal. Consequently, if a user intends to work with multichannel audio, a new register serving the function of c_{reg} would be required.

2.7.3 PSQPAM

Moreover, Khan (2019) has proposed a QRI called IFRQI (as in Improved FRQI). In this paper, the authors propose a novel intricate approach to coefficient-based encoding. Lisnichenko & Protasov (2023b) would define its amplitude encoding as being discrete, even though the information is encoded into probability amplitude. The authors claim that the preparation complexity of IFRQI is better than FRQI (depicted in Table 2.3).

As a result, the authors achieved a novel technique for encoding digital information, using coefficient-based representations: Partitioned Single-Qubit Probability Amplitude Modulation (PSQPAM).

Consider a PCM audio, e.g., with quantised amplitudes represented by even-sized binary words $P^{(2M-1)}P^{(2M-2)}...P^1P^0$. The method is to break this binary sequence into pairs $P^{(2m+1)}P^{(2m)}$. Each

$p^{2m+1}p^{2m}$	$\|00\rangle$	$\|01\rangle$	$\|10\rangle$	$\|11\rangle$
β_m	0	$\frac{\pi}{5}$	$\frac{\pi}{5}-\frac{\pi}{2}$	$\frac{\pi}{2}$

Table 2.2: PSQPAM binary pair encoding

binary pair is then encoded as an amplitude of a separate qubit $|B_{k,m}\rangle = \cos\beta_{k,m}|0\rangle + \sin\beta_{k,m}|1\rangle$, using a pre-defined truth table 2.2.

$$|A_{PSQPAM}\rangle = \frac{1}{N}\sum_{k=0}^{N-1}\left(\bigotimes_{m=0}^{M-1}(\cos\beta_{k,m}|0\rangle + \sin\beta_{k,m}|1\rangle)\right)\otimes|k\rangle \tag{2.35}$$

The striking feature of this representation is that the amplitude information is split into binary pairs, which are then indexed using a Qubit Lattice Model. This is more clearly visualised if equation (2.35) is written differently (2.36).

$$|A_{PSQPAM}\rangle = \frac{1}{N}\sum_{k=0}^{N-1}\big(\overbrace{|B_{k,M-1}\rangle|B_{k,M-2}\rangle\underbrace{|B_{k,M-3}\rangle}_{\text{SQPAM}}\otimes\ldots\otimes|B_{k,1}\rangle|B_{k,0}\rangle}^{\text{Qubit Lattice Indexing}}\big) \quad\otimes\quad \underbrace{|k\rangle}_{\text{Superpos. }(t_{reg})} \tag{2.36}$$

2.8 Towards Phase Modulation

Finally, an altertate 0-fold representation will be mentioned, proposed by Grigoryan & Agaian (2020). As a 0-fold representation, the information is encoded directly into the indexing register. However, instead of using real-valued probability amplitudes, the authors interestingly chose to use the polar *phase* of each probability amplitude, instead of its magnitude. Hence, Quantum Probability Phase Modulation (QPPM).

2.8.1 QPPM

In summary, assume that t_{reg} is in a balanced superposition. It is possible to write the probability amplitudes α_k in polar form ($re^{i\phi}$). Finally, the signal is encoded into the phase parameter ϕ (2.37). As a 0-fold technique with superposition indexing, there is no direct Value-Setting Operation. In the original paper, the preparation stage is shown to be a unitary operator (2.38). Although not easily written in terms of Clifford gates, it can be directly applied as a custom unitary gate.

$$|A_{QPPM}\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1} e^{i\phi_k}|k\rangle \tag{2.37}$$

$$U_{(2-\text{qubit QPPM})} = \begin{bmatrix} e^{i\phi_0} & 0 & 0 & 0 \\ 0 & e^{i\phi_1} & 0 & 0 \\ 0 & 0 & e^{i\phi_2} & 0 \\ 0 & 0 & 0 & e^{i\phi_3} \end{bmatrix} \tag{2.38}$$

There are a few challenges and difficulties that can be identified with this representation. As pointed out by Lisnichenko & Protasov (2023b), retrieving this state requires a pre-processing stage of transforming the phases into probability amplitudes, through Quantum Phase Estimation (QPE), which is computationally expensive.

On a smaller note, the terminology used in the original paper (Fourier Transform Qubit Representation) might end up being more misleading than helpful. Apart from a clear visual resemblance to a classical Disctete Fourier Transform (DFT) due to complex exponentials, the QPPM state is not a DFT nor aQuantum Fourier Transform (QFT). In Fourier Transforms, each coefficient related to a state will be the result of a sum containing all components of a signal. In QPPM, each state holds information of a single amplitude. Hence, it is concluded that FTQR is, actually, using only complex phases, without a stronger similarity to the phases seen in Fourier Transforms.

Nevertheless, the paper does explore potential applications of the representation, which may result in novel applications.

2.8.2 SQPPM

A natural suggestion for improving QPPM would be to turn it into a 1-fold representation, using a dedicated index, and a VSO to change the phases of a dedicated a_{reg}. Thus, the Single-Qubit Probability Phase Modulation (SQPPM) is proposed. The definition of SQPPM could be written as below (2.39)

$$|A_{SQPPM}\rangle = \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1} (|0\rangle + e^{i\lambda_k}|1\rangle) \otimes |k\rangle \tag{2.39}$$

$$\lambda_k = \frac{\pi}{2} a_k \tag{2.40}$$

In this case, theValue-Setting Operation controls a Phase Gate (2.41) to encode the sample, instead of R_Y. Furthermore, the encoding stage requires a_{reg} to start in a $|+\rangle$ state, which can be simply achieved by applying a Hadamard gate at the beginning.

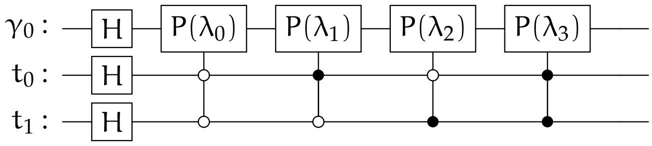


Figure 2.12: Preparation of a SQPAM representation using Value-Setting operations with R_Y gates

$$P(\lambda_k) = e^{i\frac{\lambda_k}{2}} R_z(\lambda_k) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix} \quad ; \quad P(\lambda_k)\left|+\right\rangle = \frac{1}{\sqrt{2}}[\left|0\right\rangle + e^{i\lambda}\left|1\right\rangle] \tag{2.41}$$

A Quantum Audio Conversion

To retrieve SQPPM, it is also necessary to bring the phases to a measurable amplitude domain. However, for this case, a QPE algorithm might be unnecessary. Since the probability amplitudes are stored in a single qubit, there might be a single-qubit gate that coud perform this transformation.

Notice the effect of applying a $\sqrt{X}^\dagger$ gate to a_{reg} (see Appendix A):

$$\sqrt{X}^\dagger \otimes I^{\otimes n}\left|A_{SQPPM}\right\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1} e^{i\lambda_k/2}\left[\cos\left(\frac{\lambda_k}{2}+\frac{\pi}{4}\right)\left|0\right\rangle + \sin\left(\frac{\lambda_k}{2}+\frac{\pi}{4}\right)\left|1\right\rangle\right] \otimes \left|k\right\rangle \tag{2.42}$$

Apart from a sample-wise phase ($e^{i\lambda_k/2}$), and a constant shift/scale ($\frac{\lambda_k}{2}+\frac{\pi}{4}$), this form closely approaches equation (2.14), the definition of SQPAM (!).

This is a noticeable result, as it might indicate a path toward a *Quantum Audio Conversion Algorithm*, from SQPPM to a variant of SQPAM. If the community wants to make use of QRA, a new potential investigation path would be on *how to convert between representations once they have been encoded in the system*. This would allow a given quantum signal to benefit from potential applications developed in a different representation. As seen here, SQPPM, which contains an unclear retrieval scheme, was to converted it to an SQPAM for a viable decoding.

Notice how this conversion dialogues with the definition given in (2.40), restricting λ_k to be within a range $[-\pi/2, \pi/2]$. When the signal is retrieved using the SQPAM strategy (eq. 2.18), there needs to be a different reconstruction back to the signal range (2.43).

$$a_k = 2\sqrt{p_{\gamma_k}(\left|1\right\rangle)} - 1 = 2\left|\sin\left(\frac{\lambda_k}{2}+\frac{\pi}{4}\right)\right| - 1 \tag{2.43}$$

2.9 Summary: The QRA Map

An overview of the discussed representations is gathered in table 2.3. Additionally, the observations in section 2.5, which led to new ways of systematising the area, could be organised into a Venn diagram, grouping different representations according to the definitions that have been solidified (Fig. 2.13).

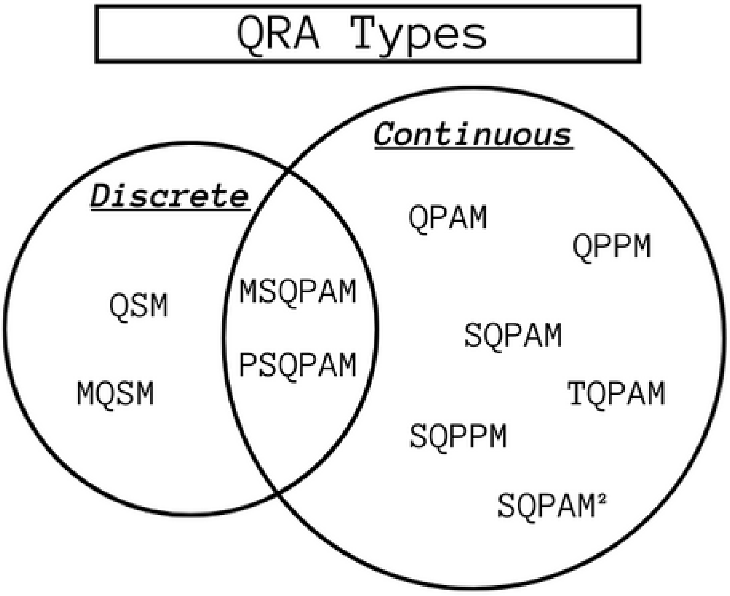


Figure 2.13: QRA Types

2.10 Quantum Audio Signal Processing: Proposed Ideas

Current digital audio signal processing (DSP) algorithms can perform a wide range of applications. Among them, recognising and categorising auditory elements (Music Information Retrieval), transforming text into speech, and implementing artistic modifications in audio during live performances.

As hinted in the introduction, the main emphasis of this work is on the latter (artistic audio modifications), which will be evident from Chapter 3 and beyond. However, a brief overview of the QSP field will be provided, looking at the potential strengths and weaknesses of particular approaches.

The content for this section is drawn from three types of sources:

- The authors that initially proposed QRA strategies also suggested initial QSP operations using their own proposed techniques.
- Other authors have also investigated more complex QSP algorithms applied to both images and audio.

- The surveys on QRIs being used so far also compile a set of algorithms for each strategy, with the inclusion of works dedicated on reviewing solely Quantum Image Processing (Yan et al., 2023; Wang et al., 2022). For a more detailed exposition, refer to these references.

QRI and QRA are are still in their infancy, and there is much to be studied from QSP. It is possible to separate different QSP algorithms into two approaches. On one hand, there are attempts of translating known algorithms to the quantum domain - which is useful both for validation, comparison and benchmarking. In this case, based on the literature reviews on signal processing for images, there may be no application yet that leans toward a full *quantum advantage* over its classical counterpart[6]. On the other hand, it is also possible to explore the structure of the quantum system to reach new algorithms that are exclusive to quantum signals.

As with the case of the literature on representations of digital signals, current literature on QSP is more focused on formalization and quantum circuit design. This is due to technical difficulties of both simulating these circuits or attempting to run them in NISQ hardware. For example, a basic signal addition method requires enough qubits for storing two disticnt representations of audio, allocate memory to ancilla qubits for carrying out necessary operations, and robustness to handle the resulting circuit depth without losing qubit coherence.

To start this brief overview, some of the applications found both for QRA can be clustered into different scopes, separated below. Naturally, the algorithms found gravitate toward the prevalent QSM representation, which is both the oldest QRA and the closest to digital audio due to its *discrete* characteristics.

2.10.1 Quantum Transformation

In Itaboraí & Miranda (2022), there is a seccion introducing the general framework behind designing QSP algorithms for simple signal arithmetic operations - the most essential toolset of any signal processing ecossystem. To exemplify, the chapter didactically dissects how quantum signal addition for QSM is constructed and how it operates on the signals, *in excruciating detail*. This explanation will be refrained here. It suffices to the large scope of this work to present the general sketches behind simple arithmetics.

[6] A promising exception may be on Quantum Fourier Transform of QPAM representations, as explored by Mistry & Ortega (2022). Even then, the preparation and retrieval complexities should also be accounted for

Syncronization

Most of these applications depend on the ability to sync signals together, so that sample-by-sample operations can be performed. In QSM (and more generally in any superposition indexed signal) This is usually achieved by means of a Serial-Based Quantum Comparator (fig. 2.14). It allocates 2q ancillary qubits (where q is the size oft_{reg}), two of which - (e_0, e_1) - are used as the output of the operation. They compute whether a multi-qubit state is smaller (1, 0), bigger (0, 1) or equal (0, 0). A 2-qubit example is shown in Fig. 2.14

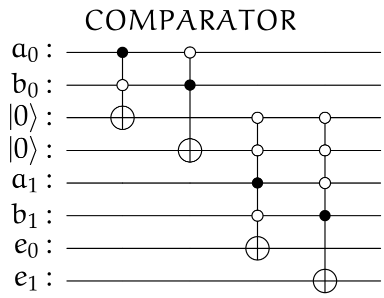


Figure 2.14: 2-qubit example of a serial-based quantum comparator circuit.

Notice that when $|a\rangle$ and $|b\rangle$ are equal, none of the control conditions are satisfied and $|e_1 e_0\rangle = |00\rangle$.

Then, this comparator could be used to make sample-wise operations in parallel (fig. 2.15).

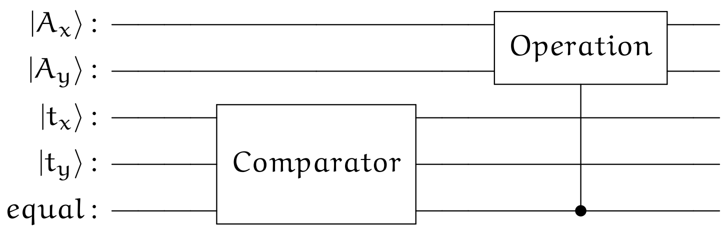


Figure 2.15: Sketch of a signal addition circuit.

Addition

For QSM-s addition, Yan et al. (2018) uses a bit-wise adder algorithm, taking into account that the Most Significant Qubit is used as a signing bit with a Two's Complement scheme.

Other Quantum Transformations

It is noted that the applications in signal transformations have been predominated by *discrete* representations, namely QSM and MQSM and their subsequent variants.

There were mentions of quantum signal transformation algorithms in four main papers:

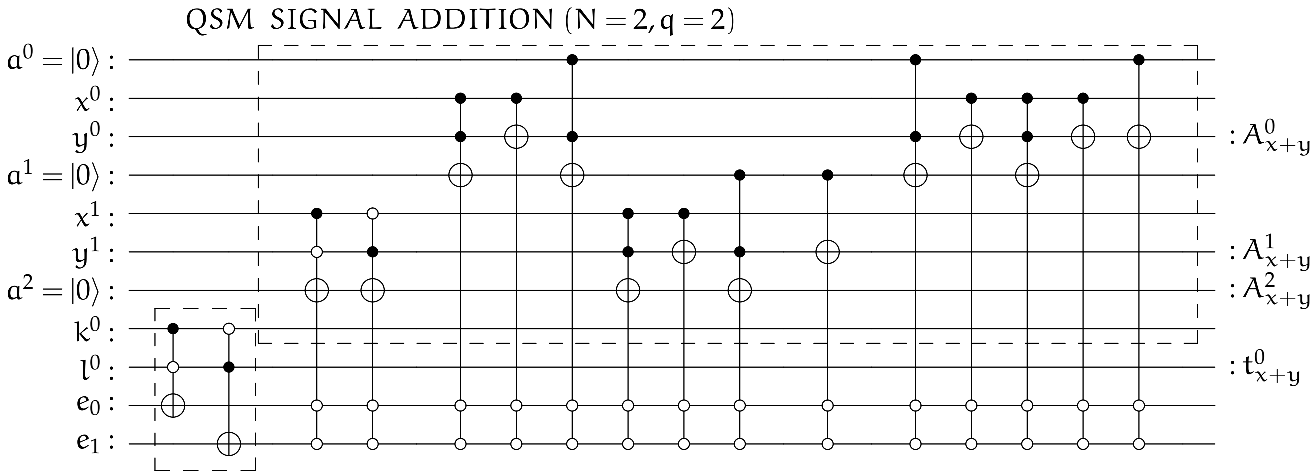


Figure 2.16: QSM signal addition operation (2-qubit long, 2-qubit amplitude depth).

- Addition/Mixing - (Wang, 2016; Yan et al., 2018; Şahin & Yilmaz, 2019; Li et al., 2018)
- Concatenation - (Wang, 2016)
- Inversion - (Yan et al., 2018; Şahin & Yilmaz, 2019)
- Channel Merging - (Şahin & Yilmaz, 2019)
- Multiplication - (Li et al., 2018)
- Division - (Li et al., 2018)
- Circular Convolution - (Li et al., 2018)
- Delay - (Yan et al., 2018)
- Reversal - (Yan et al., 2018; Şahin & Yilmaz, 2019)

It is noticed that Delay and Reversal are transformations that are operating on the indexing register of the signal.

2.10.2 Information Retrieval/Feature Extraction

In contrast, for discussions on the extraction of features for classification ot pattern recognition, there are slightly more papers that employ coefficient-based representations on the QRI literature. They can be later translated to the audio case.

In the domain of FRQI, there is an interesting investigation on the use of simple operations over a quantum dataset simultaneously, which could lead to parallel signal comparison (Yan et al., 2013). This example was translated in Itaboraí & Miranda (2022), as a form of MSQPAM processing. On a different approach, matches similar quantum images by means of *swap tests* (Li et al., 2019). Moreover, two references study potential MSQPAM image classification schemes based on classical algorithms, e.g., K-Nearest-Neighbour (Dang et al., 2018) and Relief Algorithm (Liu et al., 2018).

For NEQR (QSM), there is a translation of an image edge detection algorithm found, which could be directly applied to audio onset detection (Li, 2022).

Surprisingly, there were some remarkable publications on QRA using a 0-fold, Qubit Lattice Model indexing strategy to train a Quantum Machine Learning circuit, with envisioned applications in Speech Recognition (Qi & Tejedor, 2021) and disease diagnosis) (Esposito et al., 2022). On that same note, QLM processing is also at the core of the proposed Quantum Fast Fourier Transform (Asaka et al., 2020).

In the QCM book, Mistry & Ortega (2022) investigates frequency detection by applying a Quantum Fourier Transform in a QPAM state.

2.10.3 Watermarking and Security

Another path comparatively well explored in QSP is related to securty solutions.

Two outlined papers use Quantum Discrete Cosine Transform (qDCT) (Klappenecker & Rotteler, 2001) to propose watermarking schemes for audio, using QSM (Chen et al., 2019) and QSM-fp (Nejad et al., 2020)n. A third reference presents a criptography scheme for signature images (Yilmaz & Şahin, 2018) and a fourth applies Quantum Steganography to QSM (Chaharlang et al., 2020a).

2.10.4 Quantum Signal Compression

Lastly, there were concerns towards the optimisation of the QSM preparation circuit, improving circuit depth and simulation sizes. In the image realm, there is a compression algorithm that applies qDCT for NEQR circuit compression, which also works in QSM audio (Pang et al., 2019). Wang also traces two QRA compression algorithms for QSM-u: Minimization of Boolean Expressions and Differential Pulse Code Modulation (Wang, 2016).

Taking into account the above, we have glided into the realm of Quantum Representation of Audio, attempting to bring a different structure and presentation for the literature review, preparing the grounds for introducing music into the mix.

Representation Name	**Acronyms**	**Basic Structure**	**Amplitude Mapping**	**Space Required**	**Preparation Complexity**	**Retrieval**	**n-fold**
Quantum Probability Amplitude Modulation	QPAM (Real Ket)	$\alpha_n \lvert n\rangle$	Normalized, strictly positive vector	$\lceil \log N \rceil$	$\mathcal{O}(N)$	Probabilistic	0
Single-Qubit Probability Amplitude Modulation	SQPAM	$(\cos\theta_n \lvert 0\rangle + \sin\theta_n \lvert 1\rangle)\lvert n\rangle$	Angle Vector	$\lceil \log N \rceil + 1$	$\mathcal{O}(N^2)$	Probabilistic	1
Quantum State Modulation	QSM (FRQA)	$\lvert a_n\rangle \lvert n\rangle$	Signed Integer Vector	$\lceil \log N \rceil + q$	$\mathcal{O}(qN\lceil \log N \rceil)$	Deterministic	1
	QSM-u (QRDA)		Unsigned Integer Vector				
	QSM-fp (QRDS)		Fixed-Point Decimal Vector				
	QSM-fl ()		Floating-Point Vector				
Multichannel Quantum State Modulation	MQSM (QRMA)	$\lvert c_n\rangle \lvert a_n\rangle \lvert n\rangle$	Signed Integer Vectors	$\lceil \log N \rceil + q + \lceil \log C \rceil$	$\mathcal{O}(CqN\lceil \log N \rceil)$	Deterministic	2
	MQSM-fl (CQRDS)		Floating-Point Vectors				
Multichannel Single-Qubit Probability Amplitude Modulation	MSQPAM (PMQA)	$\lvert c_n\rangle (\cos\theta_n \lvert 0\rangle + \sin\theta_n \lvert 1\rangle)\lvert n\rangle$	Angle Vectors		$\mathcal{O}(CqN^2)$	Probabilistic	2
	SQPAM^2 (OQIM)	$\{\lvert 0\rangle(\cos\theta_n \lvert 0\rangle + \sin\theta_n \lvert 1\rangle) + \lvert 1\rangle(\cos\tau_n \lvert 0\rangle + \sin\tau_n \lvert 1\rangle)\} \otimes \lvert n\rangle$	Angle Vector + Displacement Vector	$\lceil \log 2N \rceil + q + 1$	$\mathcal{O}(N^2)$	Probabilistic	
Two-Qubit Probability Amplitude Modulation	TQPAM (QSMC & QSNC)	$(\cos\theta_k \lvert 0\rangle + \sin\theta_k \lvert 1\rangle) \otimes (\cos\tau_k \lvert 0\rangle + \sin\tau_k \lvert 1\rangle) \otimes \lvert n\rangle$	Angle Vector + Displacement Angle Vector	$\lceil \log N \rceil + q$	$\mathcal{O}(N^2)$	Probabilistic	2
Partitioned Probability Amplitude Modulation	PSQPAM (IFRQI)	$\bigotimes_{m=0}^{M-1} \{\lvert B_{k,m}\rangle\} \otimes \lvert k\rangle$	Signed Integer Vector	$\lceil \log N \rceil + q/2$	$\mathcal{O}(qN\lceil \log N \rceil)$	Probabilistic	1
Quantum Probability Phase Modulation	QPPM (FTQR)	$e^{i\lambda_n} \lvert n\rangle$	Unsigned Integer Vector	$\lceil \log N \rceil$	$\mathcal{O}(N)$	Probabilistic	0
Single-Qubit Probability Phase Modulation	SQPPM	$(\lvert 0\rangle + e^{i\lambda_n} \lvert 1\rangle)\lvert n\rangle$	Phase Vector	$\lceil \log N \rceil + 1$	$\mathcal{O}(N^2)$	Probabilistic	1

Table 2.3: Comparison between quantum representations of audio

Chapter 3

The `quantumaudio` package

During the literature review and the development of the published survey Itaboraí & Miranda (2022), it became clear that the scope of the research (at the time of writing) leaned towards theoretical proposals. Whenever proof-of-concept simulations were displayed, most failed to provide access to open-source implementations. Furthermore, it seems that those studies (both for image and audio) while focussing on signal processing applications have mostly overlooked the potential for artistic practice of this field.

A possible explanation would be that QSP is a novel field, and it remains unclear whether there is any practical advantage in encoding quantum signals this way (even more so for NISQ devices), compared to the ubiquitous DSP algorithms.

However, in artistic research, efficiency and advantage are not aways required. Artists produce their work with any instrument and methodology available in their toolbox - as long as it serves to convey their poetic and aesthetic message.

However, it seems that both QSP and artistic research can benefit from having open-source code and APIs to carry out and standardise investigations. In this work, although we focus on potential applications for music (chapters 4 and 5), we reinforce that these implementations could also provide foundations and usefulness for detailed comparative analysis between different QRA and QSP.

This has motivated the development of the `quantumaudio` package, a seed for a future library; aggregating numerous QRA strategies and spanning potential applications for audio and music expression.

Interestingly, this package was published a few days after the most recent QRI survey (Lisnichenko & Protasov, 2023b) . They also made available implementations of different QRIs in a github repository (Lisnichenko & Protasov, 2023a) - which indicates that there is also a similar research gap in QRI. Their implementation was made using Jupyter Notebooks and extends over various representations and applications.

Although quantumaudio (as of version v0.0.2), includes a comparatively modest range of representations and applications at the moment, it has adopted a different approach from Lisnichenko's emphasis. The strength of quantumaudio lies in its development structure as an *API framework*, resulting in a package aimed at more flexible and end-user (e.g. artistic) applications, as discussed in section 3.1.1.

Further, it enabled the investigation of speculative methods on how to interface simulated results from quantum representations of audio with real-time sound synthesis engines (e.g. SuperCollider) (sec. 4.1.1). In this section, details of the package and its preliminary results are analysed. The published version of the code can be found on github (Itaboraí, 2023).

3.1 Structure of the `quantumaudio` package - v0.0.2

In version v0.0.2, the quantumaudio package provides non-optimised basic encoding/decoding functionality for three of the QRA protocols discussed in chapter (2): QPAM(2.3.1), SQPAM-t(2.3.2) and QSM-s(2.4.2). The package uses the Qiskit (Qiskit Contributors, 2023) framework to handle the quantum circuit generation and simulation as presented on Itaboraí & Miranda (2022). The structure of the code consists of four classes. The content tree of the package is shown in fig. 3.1 and some specifics of the implementation are discussed:

The main class is called QuantumAudio. It is the object with which the user interacts. By initialising a QuantumAudio object to a variable, the user defines which QAR scheme to use. The first lines from the class declaration give some insight into the structure (Code 3.1):

```
class QuantumAudio():
    def __init__(self, encoder_name: str):
        self.encoder = EncodingScheme().get_encoder(encoder_name)
        self.encoder_name = encoder_name
                ...
```

Code 3.1: QuantumAudio() class consrtuctor

When an object is initialised (for instance, `QuantumAudio("qpam")`), a subclass called `EncodingScheme()` is called and returns a QRA Python class that matches the string. In this case, the variable `self.encoder` will load a `QPAM` class. Furthermore, it is noted from the content tree (fig. 3.1 that QPAM, SQPAM and QSM share common methods, such as "`.prepare()`" and "`.measure()`". This mimics the behaviour

- quantumaudio package
  - quantumaudio module contents
    - `EncodingScheme`
      - `EncodingScheme.get_encoder()`
    - `QPAM`
      - `QPAM.convert()`
      - `QPAM.measure()`
      - `QPAM.prepare()`
      - `QPAM.reconstruct()`
    - `QSM`
      - `QSM.convert()`
      - `QSM.measure()`
      - `QSM.omega_t()`
      - `QSM.prepare()`
      - `QSM.reconstruct()`
      - `QSM.treg_index_X()`
    - `QuantumAudio`
      - `QuantumAudio.listen()`
      - `QuantumAudio.load_input()`
      - `QuantumAudio.measure()`
      - `QuantumAudio.plot_audio()`
      - `QuantumAudio.prepare()`
      - `QuantumAudio.reconstruct_audio()`
      - `QuantumAudio.run()`
    - `SQPAM`
      - `SQPAM.convert()`
      - `SQPAM.mc_Ry_2theta_t()`
      - `SQPAM.measure()`
      - `SQPAM.prepare()`
      - `SQPAM.reconstruct()`
      - `SQPAM.treg_index_X()`
    - `requantize_input()`

Figure 3.1: `quantumaudio v0.0.2 content tree`

of an abstract method that is shared between classes. Each Quantum Representation of Audio will contain its own specific implementation of that abstract class, tailored to its own properties and requirements.

As a result, a method such as "`QuantumAudio().prepare()`" would internally call a "`self.encoder.prepare()`" method to create the respective encoding quantum circuit, regardless of the representation used. To some extent, this also *softly* mimics the structure of public and private methods. Put differently, consider the 'QSM.treg_index_X()' method. Even though it can be accessed by the user by writing 'QuantumAudio('qsm').encoder.treg_index_X()' (hence, *soft*)if needed, this method will not show up if the user runs 'dir(QuantumAudio)' (hence, "private").

This reinforces the point that the purpose of this package is to provide a separation of the client (working with quantum audio) from specific APIs.

3.1.1 The quantumaudio workflow

The workflow designed for applications of the `quantumaudio` package is listed below.

1. ***Start*** by having a fixed `numpy` array containing a desired digital audio signal.
2. ***Declare*** an instance of the `QuantumAudio` object, choosing a QRA strategy and store it in a variable, e.g. "`qaudio`".
3. ***Load*** the digital audio into the object - `qaudio.load_input(digital_audio)`; this will create a digital copy inside the `qaudio.input` attribute, as well as estimate the space requirements (in qubits) for this signal.
4. ***Create*** the encoding circuit using the chosen strategy - `qaudio.prepare()`; the generated quantum circuit is stored in the `qaudio.circuit` attribute.
5. ***Process*** the signal by incrementing the circuit obtained from `qaudio.circuit` using your own QSP toolbox.
6. ***Measure*** the circuit by adding projective measurement instructions at the end - `qaudio.measure()`
7. ***Simulate*** the circuit using qiskit's `aer_simulator` - `qaudio.run(shots=1000)` (the "`shots`" parameter being the amout of thimes this circuit will be run and measured; the Qiskit job results are stored in `qadio.result`, and the histogram data from the experiment in `qaudio.counts`.
8. ***Decode*** the histogram data and reconstruct an audio in the form of a `numpy` array (`qaudio.reconstruct_audio()`); the signal is stored in the `qaudio.output` attribute.
9. Optionally, visually compare the input and output with the `qaudio.plot_audio()` method.
10. Optionally, if the audio is sufficiently long, listen to the result inside an IPython environment (i.e. Jupyter Notebook) by running `qaudio.listen(rate=44100)` with the desired sample rate.

11. Sit down, enjoy, and philophise around sound quantumness.

Additionally, every `QuantumAudio` method (apart from `'.plot_audio()'` and `.listen()`) returns the `self` attribute. As a result, these methods can be concatenated and written in a reduced manner. For example, Code 3.2 below will collectivelly instantiate a QuantumAudio class, load a signal, create a preparation circuit, add measurements, and run the simulation - all in a *single line of code*.

```
>> qaudio = QuantumAudio('sqpam').load_input(dig_input).prepare().measure().run(1024)
```

Code 3.2: QuantumAudio() class consrtuctor

In conclusion, the developed structure enables a potential user of this package to test different QAR strategies with a more practical setup, without writing copies for each specific case. In fact, this also indicates that **some users may not even *need* to know the inner-workings or technical details of a given QRA**, opening the doors for artists to use and experiment with the package.

3.2 Simulating Quantum Audio

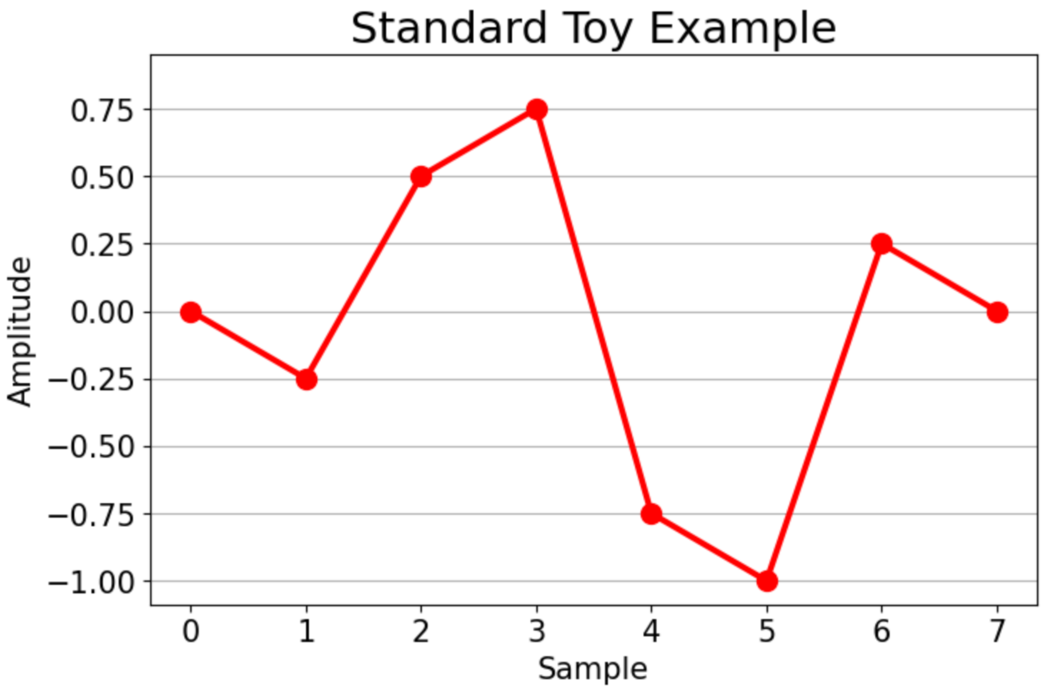
With the `quantumaudio` package, it is possible to run simulations and comparisons of preparation circuits, as well as time and space consumption. Through the `QuantumCircuit().circuit` exposed variable, it can also be used for early-stage signal processing prototyping and testing.

To exemplify the workflow described above, a simple and short audio signal was designed, The purpose of this signal is to provide a toy example for studying the representations, with a small-enough setup that still allows quantum circuit design and calculations to be carried out manually, validating experimental results with the theory. This will be referred as the Standard `quantumaudio` Toy Example (STE).

3.2.1 A standard toy example

The short audio desinged to be the our STE for validating the implementation is given in fig. 3.2a, also, this is numerically displayed in fig 3.2b. The table needs to be processed according to each representation[1]. This example replicates what would be found in a PCM audio file which is 8 samples long and with 3-bit depth amplitude quantization for discrete encoding schemes. To run this examples, code 3.2 was used, only changing the encoder parameter to `'qpam'`, `'sqpam'`, and `'qsm'`, respectively.

[1]Numbers displayed here were rounded to three decimal cases, but the code run with `quantumaudio` used the most precise result from the mapping preprocessing step



(a) Plot Visualisation

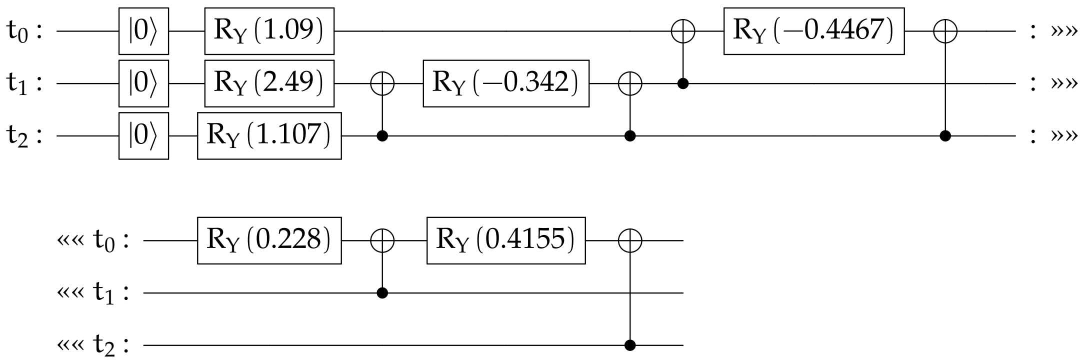
0.00	−0.25	0.50	0.75	−0.75	−1.00	0.25	0.00	Array Visualisation

↓

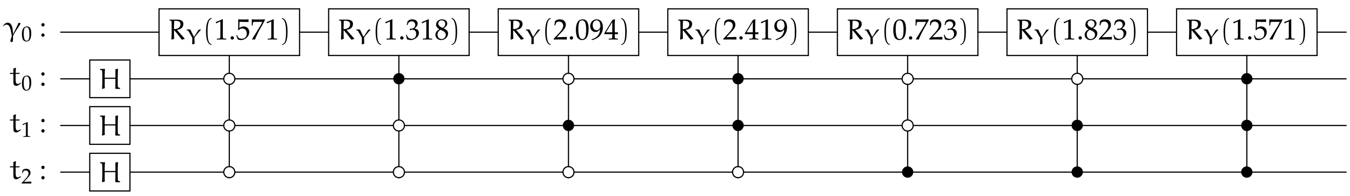
0.324	0.243	0.487	0.568	0.081	0.000	0.406	0.324	to prob. Amplitudes (2.13)
0.785	0.659	1.047	1.209	0.361	0.000	0.912	0.785	to Angles (2.16)
000	111	010	011	101	100	001	000	to 3-bit signed integer (2.22)

(b) Input Mapping

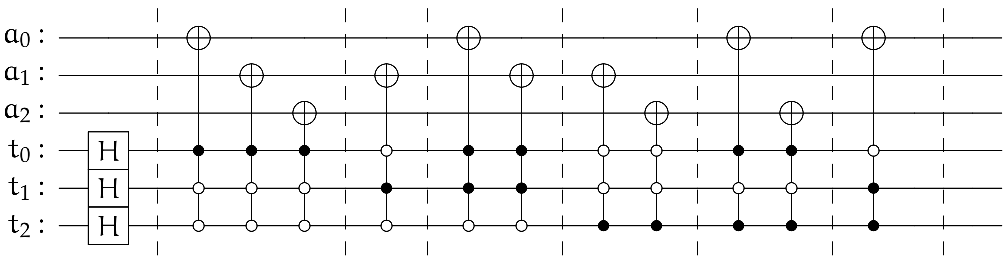
Figure 3.2: A Standard `quantumaudio` Toy Example



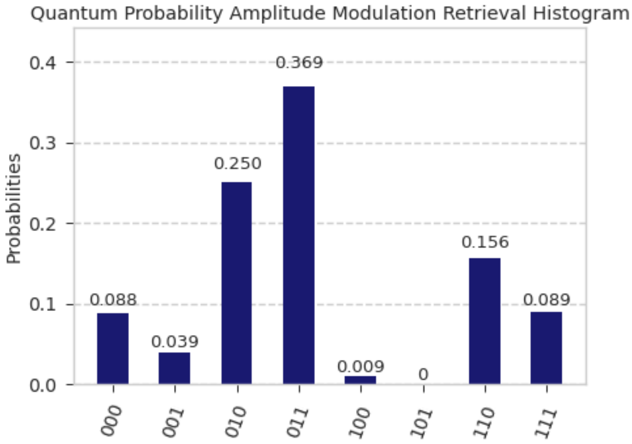
(a) STE QPAM Preparation Circuit



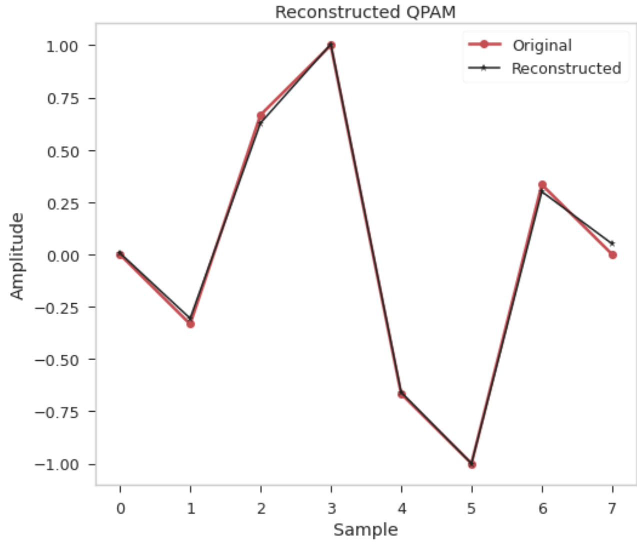
(b) STE SQPAM Preparation Circuit



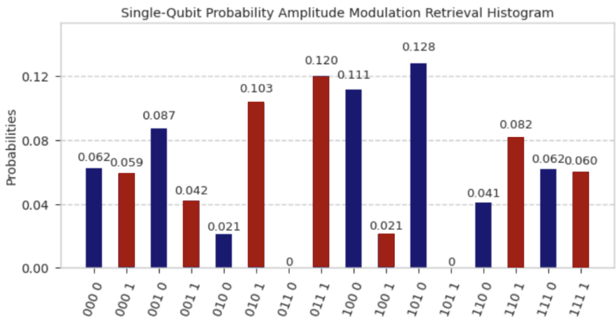
(c) STE QSM Preparation Circuit



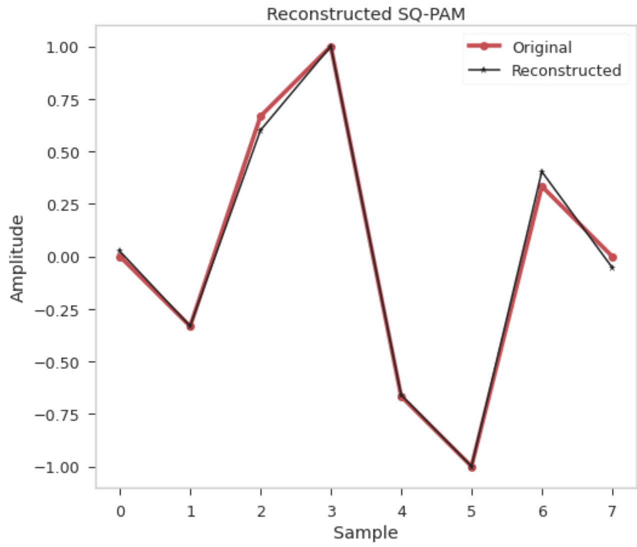
(a) STE QPAM Retrieval Histogram. Shots = 1024



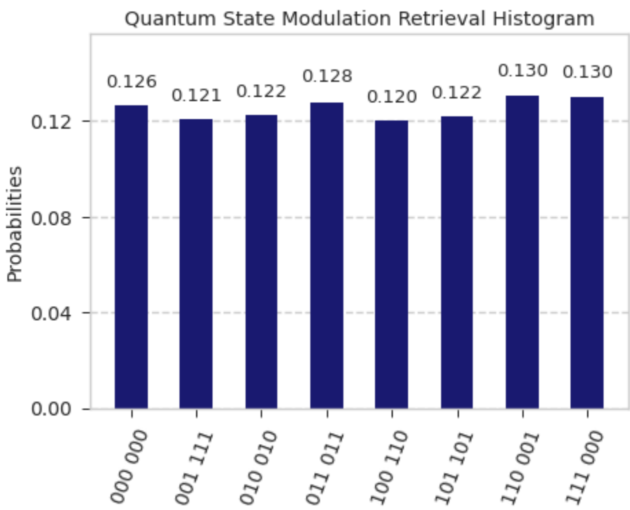
(b) STE QPAM Reconstructed Audio



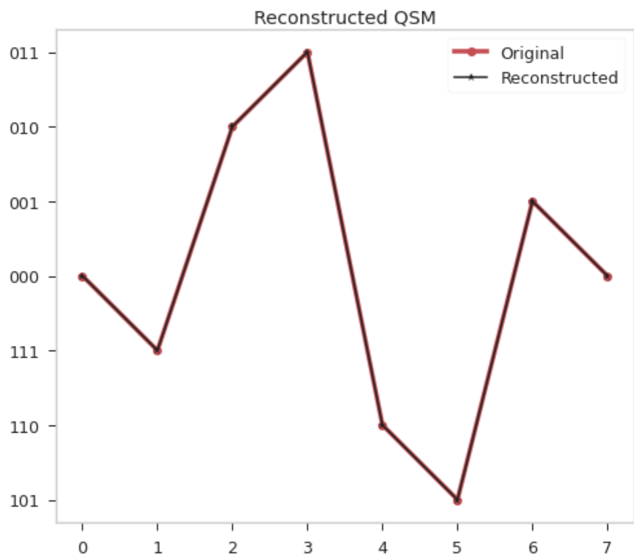
(c) STE SQPAM Retrieval Histogram. Shots = 1024



(d) STE SQPAM Reconstructed Audio



(e) STE QSM Retrieval Histogram. Shots = 1024



(f) STE QSM Reconstructed Audio

Chapter 4

Demonstrations

In this chapter, we move forward with the intention of using Quantum Computing as a musical device (stated in chapter 1 as one of the research questions, **RQ1**). In possession of the theoretical framework (chapter 2) and a practical implementation of an API for running Quantum Representation of Audio (chapter 3), we are able to prototype a digital musical instrument that could use `quantumaudio` within its framework. Then, the objective is to demonstrate possible hybrid classical-quantum exploratory paths of heuristic audio effects with QRA. As a result, a few simple case studies were documented and outlined.

However, before discussing the case studies, there are a few technical considerations and practical design questions to be addressed. Most importantly, we need to build and implement a musical interface that is capable of transforming "quantum audio" into synthesized sounds.

Some examples in this chapter mention audio files available in the accompanying media folder. Refer to the index for information about the sound label number and its respective file name in the folder.

4.1 Interfacing Quantum Computing with Musical Interfaces

There are many approaches to this interfacing problem in the literature. To cite some from the *Quantum Computer Music* Book Miranda (2022b), (including Itaboraí & Miranda (2022)), it is possible to distinguish interfaces that are either closer to a QC-driven (i.e. Python) or musical software-driven ecossystem.

For instance, from a QC perspective, QVTS (Mannone & Rocchesso, 2022b), Quanthoven (Miranda et al., 2022), Qeyboard (Clemente et al., 2022) and others are implementations that are fully encapsulated in Python, making use of Jupyter Notebooks for a quantum computing framework - e.g., Qiskit (Qiskit Contributors, 2023), StrawberryFields (Killoran et al., 2019), etc. - together with audio synthesis implementations inside Python, e.g. Librosa (McFee et al., 2023), Music21 (Cuthbert & Ariza, 2010), etc. In contrast, the Quantum Music Playground (Weaver, 2022) and the QAC Toolkit (Hamido, 2022) have entirely integrated QC languages and/or simulators inside a music software, respectively, Ableton Live and Max MSP. Additionally, Topel et al. (2022) have connected Ableton live directly to quantum hardware.

Apart from completely enclosed strategies, it is possible to connect multiple software - in this case, to exploit the flexibility of Python programming with the powerful capabilities of a music programming language (MPL). In computer music, we can use a music networking protocol to parse musical events between software: Open Sound Control (OSC) (Wright & Freed, 1997). Again, there is a design choice on which framework is the principal. In other words, the performer could be using a MPL to make music, and querying a Python service for running and providing results from a quantum circuit, as with *OSC-Qasm* (Hamido & Itaboraí, 2023). An equivalent scenario could be someone who is mainly using Python - studying a quantum algorithm - and needs a MPL to serve as a synthesis engine for the results (Miranda & Basak, 2022; Itaboraí & Miranda, 2022). The latter strategy is the one used here. To mention an alternative approach, it is also possible to imagine two people operating a musical instrument instead of one. For instance, there could be one person operating Python, while the other is concerned about the musical patches, as seen in *Dependent Origination* (Itaboraí et al., 2023) and in *Rever* (sec. 5.2).

In summary, for the extension of this work, the `quantumaudio` module will be synthesized through SuperCollider (SC) (McCartney, 2002).

4.1.1 Python and SuperCollider

SuperCollider has an advantage for our approach, since it is a client-server based language. While it is possible to configure SC using its dedicated client language - *sclang* - to receive custom OSC messages and relay them to the server - *scsynth* - it is also possible to exploit the fact that the client-server communication itself is *also* conducted over OSC. Thus, it is possible to use Python itself as a custom SC client. There are some known projects that completely implement *sclang* in Python (Kirkbride, 2016; Samaruga & Riera, 2022). However, in search of a lightweight prototyping solution that implements a few of the most frequently used functions of SC, a choice was made in favour of the `python-supercollider` package (Jones, 2020). In so, the implementation of the synthesizers themselves remained mostly at the SC side, hoping for a more platform agnostic implementation in

Python. This has also allowed the integration of `python-supercollider` as a package dependency of `quantumaudio`.

The `python-supercollider` package

This package's basic development concept is delineated as follows:

- Prior to the integration, design a `SynthDef` in *sclang* that uses the data that will be received from Python. This `SynthDef` has a key name attributed to it.
- Compile the `SynthDef` into machine code and store it in a folder that can be accessed by *scsynth*. This can be done automatically by running the '`.store`' method after enunciating the `SynthDef`.
- Run a scsynth server, either from the SuperCollider IDE or from a command-line prompt. It will load all `SynthDef`ss stored.
- In Python, use the `Server` and `Synth` objects to call instances of the custom `SynthDef` by its keyname, using QC data and other Python variables as the arguments and parameters of the synthesizer.
- Play your instrument by changing the synth parameters from Python, using the '`.set()`' method.

4.1.2 Fixing `python-supercollider` Dependencies

All `python-supercollider` objects are essentially generating structured messages according to the SC server syntax, and using an internal OSC client and server to communicate directly with *scsynth*.

However, as of version v0.0.5, this package relies on `LibLO` (Harris & Sinclair, 2019) and `pyliblo` (Sacré, 2015) as the dependency for OSC messaging. The problem arises by the fact that `pyliblo` is currently deprecated, and there are installation complications for `LibLO`, as it often needs to be compiled and installed manually in every operating system. This posed an issue for a more general end-user, as this would mean a significant technical barrier for installing and applying `quantumaudio` musically.

Consequently, during the implementation of `quantumaudio`, this problem was addressed, and the `python-supercollider` package was completely refactored to use a more recent and well-maintained package - Python-OSC (2013) - as the Open Sound Control dependency. This will lead to a new published version of `python-supercollider` (v0.0.6), as a result of this project (Itaboraí & Jones, 2023).

4.2 Interfacing solution: Wavetable Synthesis

It is important to state that when simulating quantum circuits with `quantumaudio`, scalability can become a major issue, as seen in chapter 3. Furthermore, as the number of qubits increases, the complexity of a classical simulation grows exponentially (Jozsa, 2010), eventually making it impractical to run in a live-electronics performance setting. With this limitation in mind, the following question emerges: How to work with audio signals *in the audible range* without overloading a simulator?

The solution encountered was to turn to Wavetable Synthesis (WS).

Wavetable synthesis is a fundamental technique used in digital audio synthesis. It is used to generate basic oscillators, and is arguably the main component of the digital-to-analog conversion. It involves a pointer that cycles through a buffer (i.e. a "lookup table", fig. 4.1) containing pre-stored digital audio amplitudes. The rate through which these samples are looped is controlled and can be varied at any time. When a pointer falls into fractional positions of the buffer, interpolation is used to estimate the amplitude at that position. To make a lookup table audible, it suffices to use an reading rate that falls into the audible region.

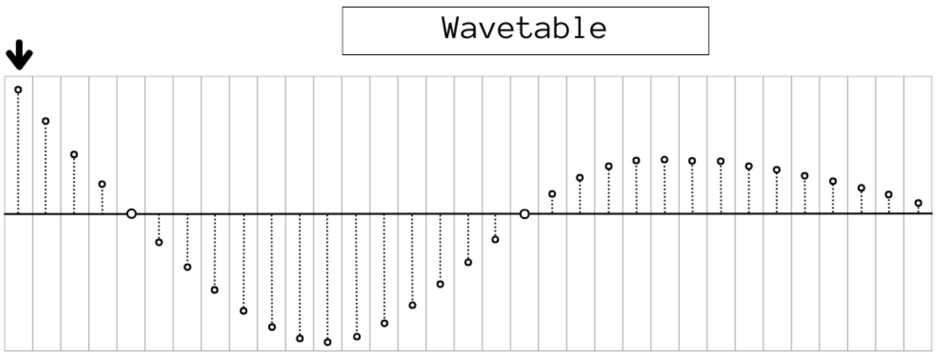


Figure 4.1: A Lookup Table Visualisation

The advantage of using WS is its ability to work with very small signals compared to a typical audible audio sample. For instance, one second of audio at 48000 Hz would require at least 16 qubits for QPAM strategies, whereas a table with 128 or 512 values (typical WS buffer sizes) would occupy less than 10. As a result, WS becomes a viable candidate for low-level quantum-digital audio interfacing.

In summary, this hybrid structure would contain three elements (fig. 4.2): First, a fully implemented WS in SuperCollider, or any audio synthesis engine of choice (right side of the figure). Second, the means of updating the shape of the waveform in real time (i.e. `python-supercollider`). Third, a system capable of simulating (or retrieving results from real quantum hardware) and reconstructing a waveform from a QRA histogram (left side of the figure).

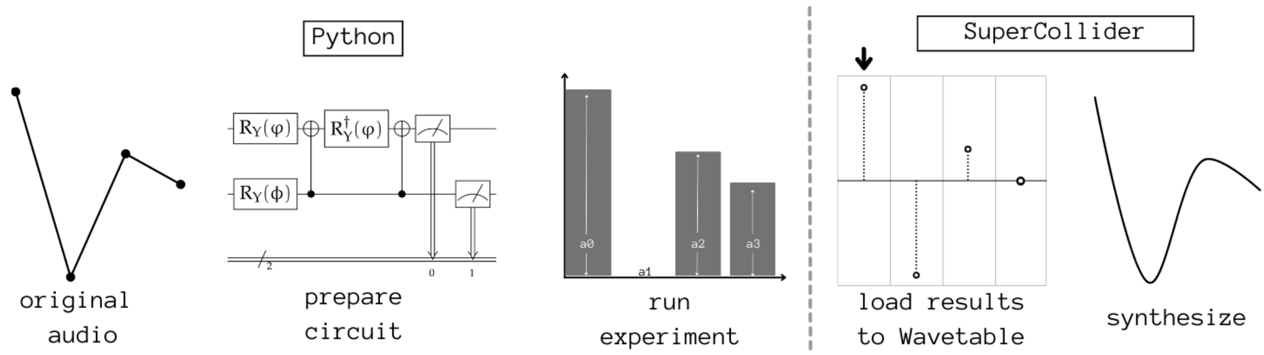


Figure 4.2: Interfacing QRA and SuperCollider

4.2.1 Integrating `quantumaudio` with Supercollider

A simple WS can be deployed in SC in a few lines of code, as shown below (code 4.1). The `Osc` Ugen is SC's implementation of a lookup table oscillator, reading from a buffer. Then, using the `Buffer` object in `python-supercollider`, it is possible to replace the table that the `Osc` is reading from. Finally, SC sounds can be generated and controlled from Python (code 4.2).

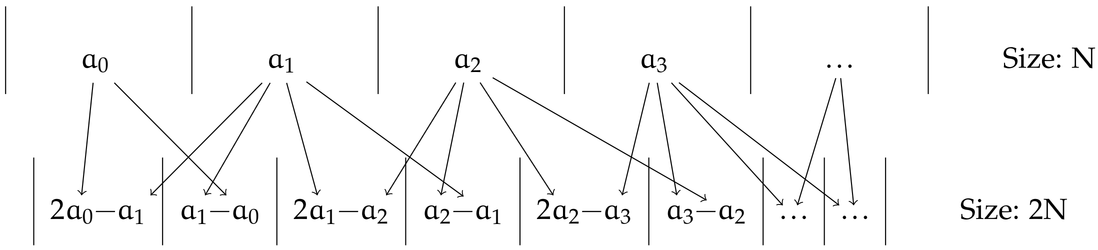
```
SynthDef(\qTable,
    { |buf = 0, freq = 250.0, gain = -6.0, out = 0|
        var sig;
        sig = Osc.ar(buf, freq); //Wavetable read from a buffer
        sig = LeakDC.ar(sig); // Avoid excessive fluctuation and digital distortion
        sig = LPF.ar(sig, 12000); //Low-pass Filter for aliasing purposes
        Out.ar(out, sig*gain.dbamp!2); // Stereo Output
}).store;
```

Code 4.1: SC Wavetable Synthesis example

```
from supercolllider import Server, Buffer, Synth
server = Server() # Connects python-osc with scsynth
buffer = Buffer.alloc(server, len(wavetable)) # Allocates memory for a buffer in scsynth
buffer.set(wavetable) # Populates the buffer
synth = Synth(server, "qTable", {"buf": buffer, "freq": 440}) # Starts playing an instance of the \qTable SynthDef
```

Code 4.2: Using SC Wavetable Synthesis from Python

NOTE: The `Osc` Ugen is an optimised implementation of Wavetable Synthesis, leading to a computationally efficient real-time synthesis. To achieve this, instead of using a plain table with the raw waveform data, there is a preprocessing stage, leading to a buffer with SuperCollider's *Wavetable Format*(SuperCollider, 2013; Scheirer & Ray, 1998). According to its specification , this format enables SC to perform the required WS interpolations using only cheap addition and subtraction operations during runtime, while the required multiplications are made once during preprocessing and stored in the buffer (4.1). As a result, this preprocessing stage needs to be accounted for on the Python side (Code 4.3). Notice that the Wavetable Format requires the buffer to be twice as long as the original table.



(4.1)

```python
def toWavetable(signal):
    wavetable = np.zeros(2*len(signal))
    wavetable[0::2] = 2*signal - np.roll(signal, -1)
    wavetable[1::2] = np.roll(signal, -1) - signal
    return wavetable
```

Code 4.3: Wavetable Format preprocessing

Finally, there is a musical framework for listening to small `quantumaudio` reconstructed signals with SuperCollider.

4.3 Case Studies

4.3.1 Geiger Counter Effect

With a tool capable of generating circuits that encode and decode QARs, a logical initial idea would be to observe and investigate how a signal is reconstructed from quantum measurements. In other words, the first case study investigates a simple *Quantum Audio Bypass Circuit*. As expected, an audio signal represented in a quantum state needs to be measured, which means that the quantum processor will run multiple identical copies of the circuit. The collected measurements are then grouped into a histogram, post-processed, and the reconstructed data is interpreted as an audio signal.

Effectively, as explored in chapter 2, for coefficient-based representations, the more measurements, the better the estimation of the original signal. This is also true for state-based representations, provided that enough measurements are made to ensure a perfect reconstruction of the signal.

With that in mind, it is possible to compute and listen to the reconstructed signals at various stages of the measurement process. This is a potential heuristic: to analyse audio reconstruction by purposely *underestimating* the number of shots required to retrieve a sufficient approximation of the audio. By synthesizing the results, this becomes an initial idea for an audio effect. Furthermore, the number of measurements (*shots* in Qiskit terminology) would serve as a parameter for the effect.

An illustration of this is shown in Figure 4.3, which contains the results of a 4-qubit QPAM circuit. The signal encoded in this bypass circuit consists of two periods of a sinewave.

By varying the number of shots, different auditory results can be achieved. [1]

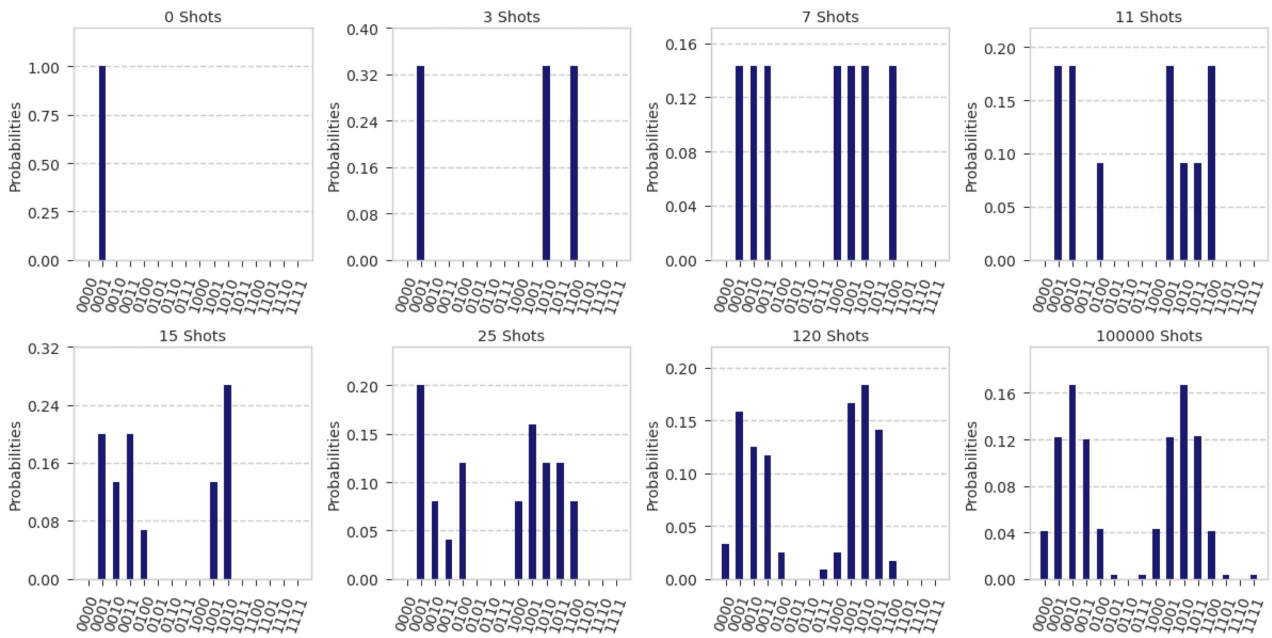


Figure 4.3: Different retrievals of a 4-qubit QPAM loopback circuit with 1, 3, 7, 11, 15, 25, 120 and 100000 shots, respectively.

Expanding from the idea displayed in fig. 4.3, the following heuristic effect is achieved, as described by the following pseudo-code[2]:

1. A desired audio sample is chosen, loaded and synthesised in SuperCollider using the`qTable` `SynthDef` defined in 4.1 (remember the SC wavetable format, which doubles the table size)

2. A QPAM state is created by encoding the same audio sample and bypassed back to digital

[1]How could this be used artistically? (see chapter 5)

[2]Example codes for case-studies 1 and 2, written in Jupyter Notebook, are available in the `quantumaudio` documentation (Itaboraí, 2023).

through projective measurements.

3. The generated circuit is simulated through a sequence of individual experiments. The number of measurements in each experiment is defined by a set of varying (and crescent) numbers stored in a list.

4. Every time an experiment is completed, a new signal is reconstructed.

5. The new signal replaces the wavetable being synthesized, using the `Buffer.set()` method, changing the auditory result in real time

6. Optionally, a time delay is added before repeating the process from step 2.

7. Steps 2-6 are repeated until the list of shots is finished.

Below are three examples of this effect in practice using the algorithm above, but varying the parameters, as detailed in tables 4.1 and 4.2.

Example 0: Quantum "Dithering" - By simply running one experiment with 1024 shots, it is possible to verify that the number of measurements was indeed underestimated, leading to a signal that follows the global profile of the sinusoindal shape, but contains certain amount of noise (fig. 4.4). This is similar to what happens when Dithering is applied to a digital signal - although with the final intention of masking potential quantization noise, which is not the case here.

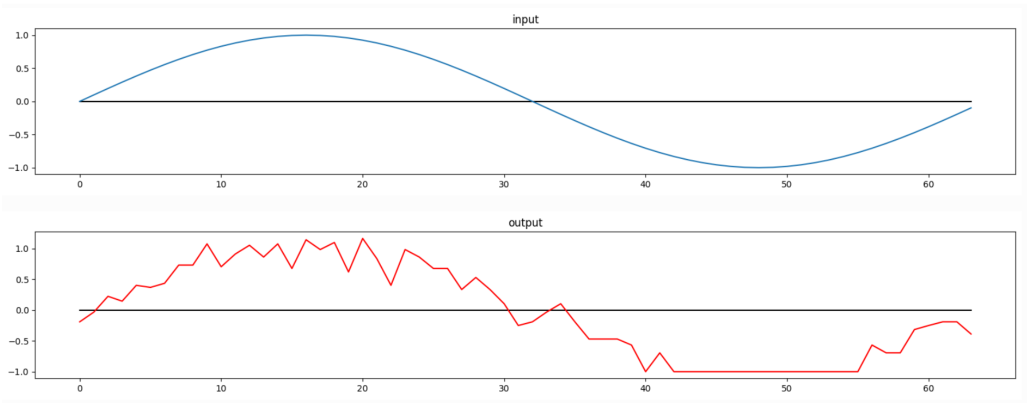


Figure 4.4: Example 0: "Dithering"

Example 1: Noise transient attack - The auditory result is close to a rapid succession of different tables, perceptually heard as a glitch (see **Sound 4.1**). This example is represented in fig. 4.5, and depicts different *regimes* of this effect. First, there is a randomized sequence of impulses (1 and 10 shots). Then, there is a section with saturated noise (100 shots). From there, the sound is seen as noisy, but a periodic profile starts to appear more prominently (500-1000). Finally, the last stage is represented by a form progressively closer to the original input (10000-100000).

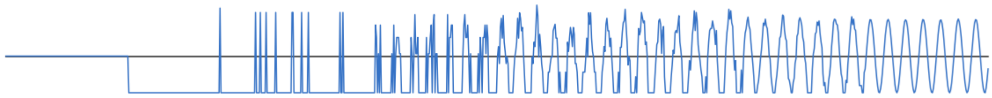


Figure 4.5: Illustration of a Example 1

Example 2: Changing timbral qualities - By adding a delay between updates, it is possible to start to notice how each result leads to a different waveform, leading to varied timbres. Additionally, the increase in *shots* and the process of convergence onto the original signal becomes more perceivable to the ear (see **Sound 4.2**).

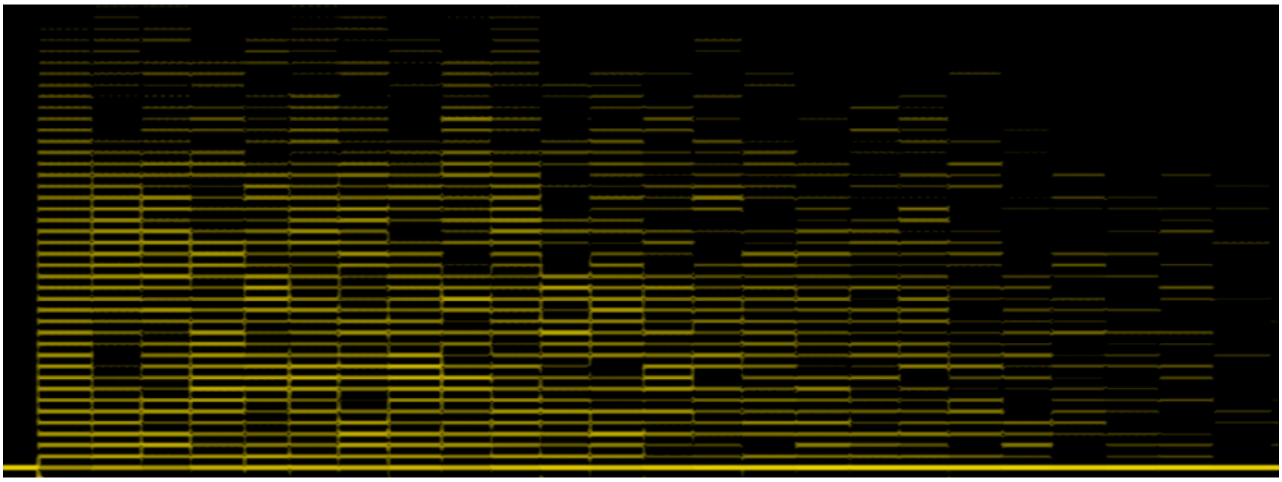


Figure 4.6: Example 2: Spectrogram depicting the changes in harmonic components at each iteration.

Example 3: The Geiger Counter Effect - For this example, a larger signal was used, with a slightly different approach. Instead of using a wavetable containing one period of a sine function (with a reading speed at an audible range), the parameter roles were inverted. In other words, the encoded signal contains 100 periods (audible) and the wavetable is read at a rate of 1Hz. As a result, this effect clearly and didactically moves between sound regimes. What is visually depicted in example 1, here is audibly perceived (see **Sound 4.3**). The sound starts almost silent. A first measurement raises a peak, generating a click train pulse. With a few measurements, these clicks will generate more complex rhythmic patterns. Eventually, the clicks accumulate to a point where a saturated noise emerges. The auditory result up to this stage resembles the sound of a Geiger Counter, measuring increases in ionising radiation. As more measurements are computed, the noise starts to be perceived almost as white noise. As the signal approximates the original audio form, the 100Hz frequency appears smoothly and gains presence over the vanishing noise, until it is the only sound that is noticed.

The auditory result from this example is so emblematic and sonically representative of a quantum measurement procedure for retrieving quantum information - and, in addition, *strongly* similar to the sonification process that occurs on a real Geiger Counter - that this entire case study was then named *The Geiger-Counter Effect (GCE)*.

The spectrogram of this experiment is shown in Fig. 4.7.

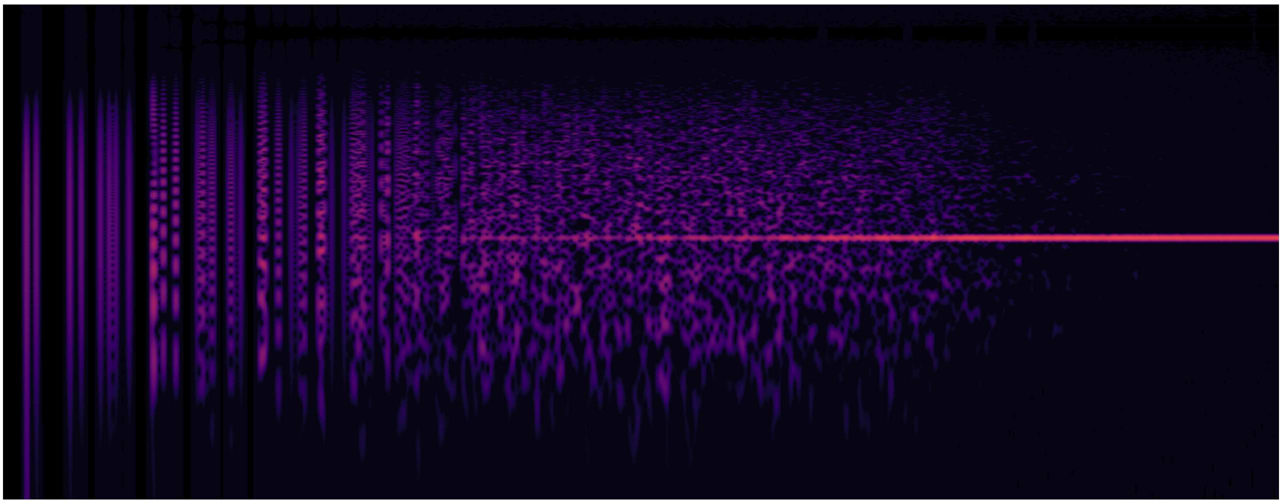


Figure 4.7: Spectrogram of the effects of an underestimated QPAM bypass circuit applied to a sinewave.

4.3.2 R_Y Rotations

In this demonstration, a simple signal processing procedure using `quantumaudio` will be shown.

Taking advantage of the fact that SQPAM representations use a single qubit to encode all audio information with controlled rotations, let us explore the effect of further applying R_Y rotations to the encoded signal. The routine is defined below.

1. A desired audio sample is chosen, loaded and synthesised in SuperCollider using the `qTable SynthDef` defined in 4.1.
2. An SQPAM state is created by encoding the same audio sample.
3. Append a simple parametrized $R_Y(\rho(k))$ gate to the circuit at the amplitude register. The parameter $\rho(k)$ is defined in the table 4.1.
4. Measure back the circuit and reconstruct the signal.
5. Replace the SC wavetable, similar to the GCE, but now updating the parameter to $\rho(k+1)$
6. Repeat steps 2-5 until a full rotation occurs ($\rho(k+n) >= 2\pi$).

A method to visualise this effect would be to rely on a Bloch Sphere representation of a_{reg}. Consider that for every index, the a_{reg} qubit will be in a state that lies on a region of the sphere that circumscribes all possible real-valued probability amplitudes. Since t_{reg} is in superposition, we could visualize all samples simultaneously in a circle. Additionally, if an R_Y gate is applied, all amplitudes would be *shifted* by a certain amount $\rho(k)$ (fig. 4.8).

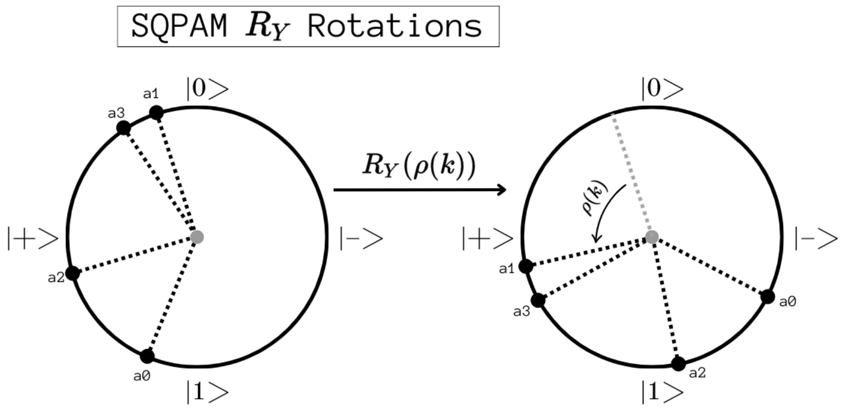


Figure 4.8: Visualising SQPAM R_Y rotations

Example 4: A Noisy Inversion process - It is possible to verify an interesting mutation effect in this example (fig 4.9). The cosine shape turns into an 'm' shape and eventually leads to an inverted cosine. On its way back it then becomes a '*w*' and finally back to cosine again. The resulting effect of the m and *w* shapes is that the sound is perceived as being one octave higher, as the periodicity of the wave has effectively doubled (fig 4.10, **Sound 4.4**).

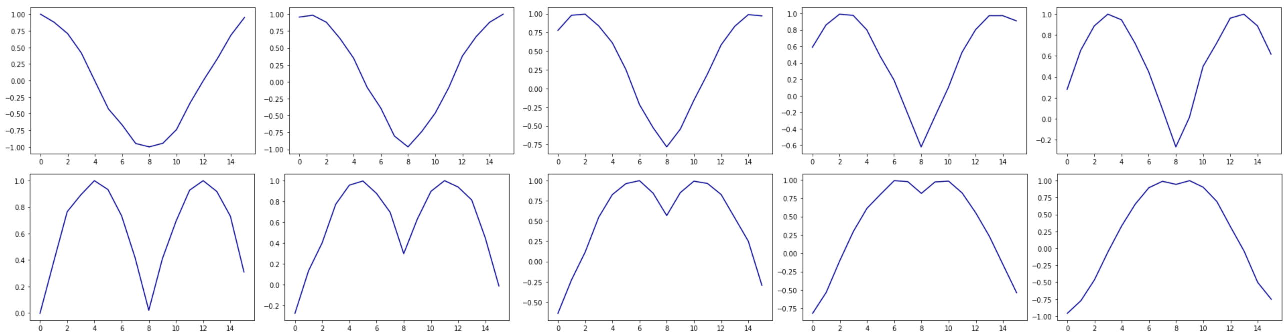


Figure 4.9: Example 4: Rotatinga Cosine waveforms

4.3.3 Quantum Audio Interference Expansion

This case study, named Quantum Audio Interference Expansion (QAIE), explores the influence of subdividing the QRA superposition data structure. This indexing strategy allows for compact information storage, since the nuber of possible indexes scales esponentially with the number of

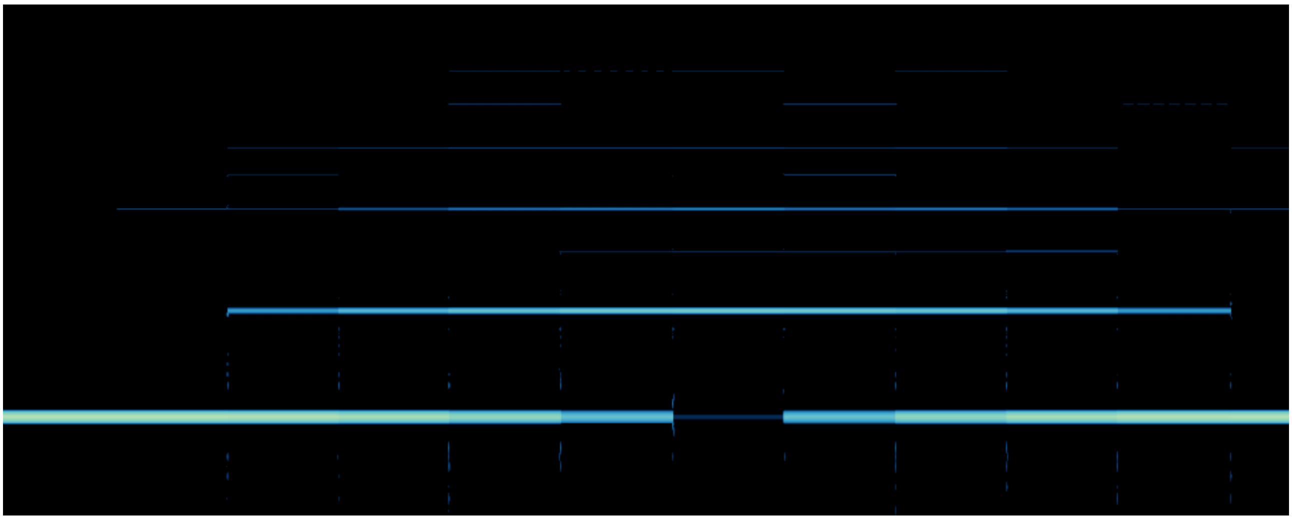


Figure 4.10: Example 4: Spectrogram Visualisation (Logarithmic bins). Lowest Line is the wavetable fundmental

qubits.

In other words, to double the size of a signal, it suffices to add just one qubit to t_{reg}. This heuristic effect (QAIE), however, exploits this fact, but goes in the opposite direction: If two halves of a QRA are encoded separately, more qubits will be required (4.2). As a result, the signal as a whole will take more quantum computational space.

$$\begin{aligned} \text{One signal of size} N &\longrightarrow \lceil \log N \rceil \text{qubits} \\ \text{One signal of size} 2N &\longrightarrow \lceil \log 2N \rceil = 1 + \lceil \log N \rceil \text{qubits} \\ \text{Two signals of size} N &\longrightarrow 2\lceil \log N \rceil \text{qubits} \end{aligned} \tag{4.2}$$

The heuristic concept is simple: To encode two sections of a signal (or generally, two independent signals) using two indexes, t_{reg1} and t_{reg2}, but sharing the same amplitude register [3] a_{reg}.

The pseudo-code for this demonstration is described below, and also illustrated in the pseudo-circuit:

step.1 Encode two QRA independently. In other words, there will be two quantum audio states $|A_1\rangle$ and $|A_2\rangle$, indexed separatedly (t_{reg1} and t_{reg2}).

step.2 Create a third QRA, $|A_{QAIE}\rangle$, where $a_{regQAIE} = a_{reg1} = a_{reg2}$ and $t_{regQAIE} = t_{reg1} \otimes t_{reg2}$.

step.3 *Merge* both circuits by simply concatenating the preparation instructions from both audios, using their respective register (fig. 4.11).

[3]Notice that this is **different** from the multichannel audio discussed in chapter 2, where both t_{reg} and c_{reg} are used *simultaneously* as control conditions when applying Value-Setting Operations. In this example, there are effectively *Two independent sets of VSOs* taking place from separate registers

step.4 Measure the collective state and interpret the time registers as one collective index ($|t_{reg1}t_{reg2}\rangle$) - not two - while decoding the signal.

As a result, the decoded sound will not be double the size, but ***exponentially*** larger instead(4.3), considering that it was the amount of qubits on the indexer that have doubled. Furthermore, the resulting amplitudes of this expanded signal will be a combined and mixed version of the information from both original signals; since the indexes were mixed together ($|t_{reg1}\rangle \otimes |t_{reg2}\rangle$. Depending on the QRA strategy, this will lead to different *interference patterns*.

$$\text{(2 signals) } N + M \textit{ samples} \xrightarrow{\text{Encode}} \text{(QRA) } \underbrace{\lceil \log N \rceil}_{n} . \underbrace{\lceil \log M \rceil}_{m} \textit{ qubits} \xrightarrow{\text{Decode}} \text{(1 signal) } N^m \textit{ samples} \tag{4.3}$$

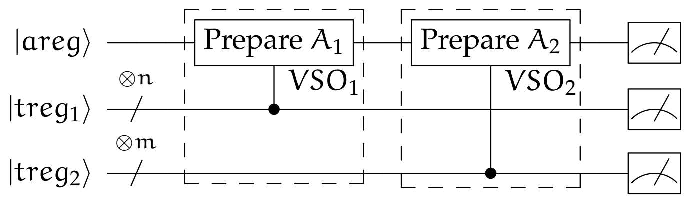


Figure 4.11: QAIE Circuit Sketch

Example 5: QPAM *pseudo*-convolution - Starting with QPAM representation (fig, 4.12), QAIE reveals a (*misleadingly*) surprising outcome. It appears that the shape of the first signal emerges whenever there is a peak in the second signal. This strongly resembles the effect of *impulse response* that appears when performing a *linear convolution*. However, there are *fundamental* differences between the QAIE process and convolution.

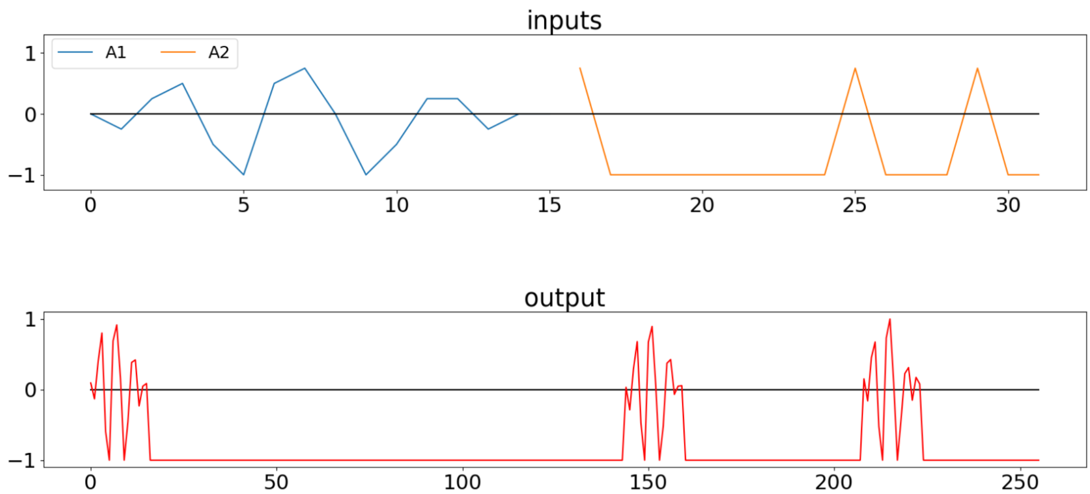


Figure 4.12: Example 5: QAIE initial experiment with QPAM and impulse signals

For instance, the size of the signal is significantly larger and follows the relation presented in (4.3), compared to the output size from a typical linear convolution $(N + M - 1)$.

Also, the operation that is carried out here does not commute in respect to the signal inputs (as depicted in fig 4.11). In other words, $QAIE(|A_1\rangle, |A_2\rangle) \neq QAIE(|A_2\rangle, |A_1\rangle)$. A possible explanation for this is the fact that the t_{reg} merging process leads to the first input being indexed by the Least Significant Bits - whereas the second input gets the Most Significant Bits of t_{reg}.

Furthermore, this familiarity may only occur when using QPAM, as will be observed in the following example.

Example 6: Expanded Interference Patterns - In example 6, we will observe how different QRA, as well as different input order will lead to distict outcomes. The inputs A_1 and A_2 are described in table 4.1 and depicted in fig.4.13. For each representation, QAIE was applied two times - $QAIE(|A_1\rangle, |A_2\rangle)$ and $QAIE(|A_2\rangle, |A_1\rangle)$.

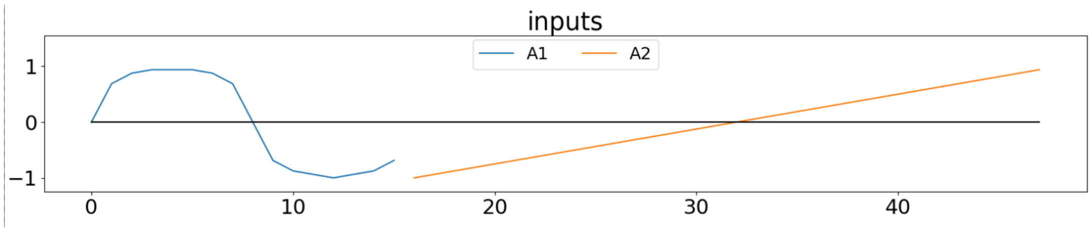


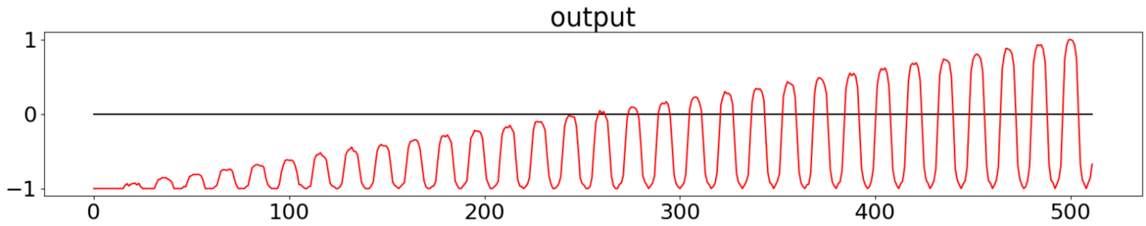
Figure 4.13: QAIE inputs.

6a) - QPAM As in Example 5, Applying QAIE to QPAM leads to the first input being repeated periodically (M times), while the amplitude of each repetition is *modulated* by the second input (fig. 4.14). Furthermore, it is possible to verify (from **step.2**) that, *in this case*, the resulting audio is exactly $|A_1\rangle \otimes |A_2\rangle$ ($|A_2\rangle \otimes |A_1\rangle$).

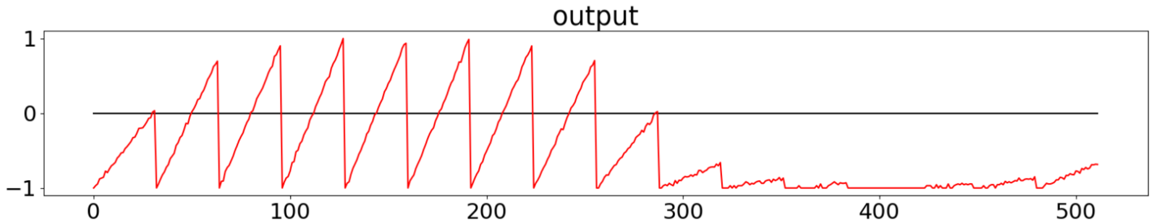
6b) - SQPAM With this QRA strategy, the interference pattern changes (figs. 4.15), resembling the shape of the results obtained in Example 4 (R_Y rotations).

6c) - QSM In contrast, a *discrete* representation encodes information by switching the qubit eigenstates ($|0\rangle$, $|1\rangle$) , resulting in sudden jumps in the profile of A_1 (fig. 4.16); whenever the A_2 encoding state flips a qubit in amplitude register. This is specially clear in fig. 4.16b, where the roles of A_1 and A_2 are switched.

Toward artistic applications, this case-study could also be a symbolic analogy of the known aphorism: *"The whole is greater than the sum of its individual parts"*.

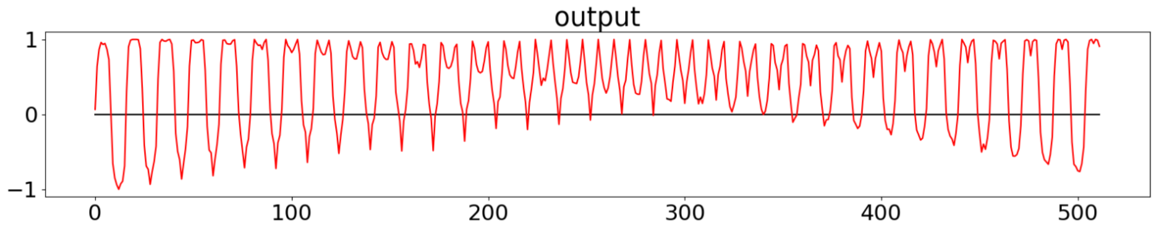


(a) QAIE($|A_1\rangle$,$|A_2\rangle$)

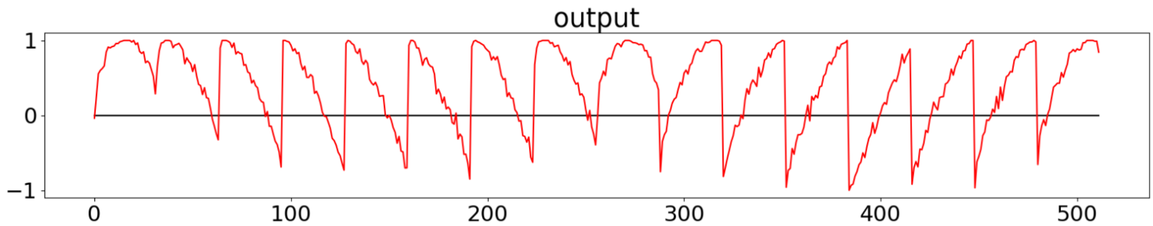


(b) QAIE($|A_2\rangle$,$|A_1\rangle$)

Figure 4.14: Example 6a (QPAM)

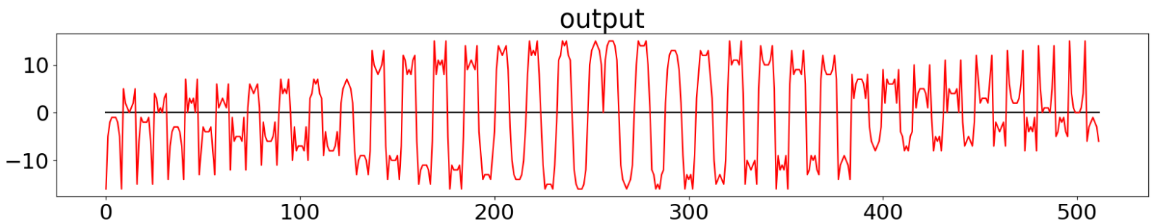


(a) QAIE($|A_1\rangle$,$|A_2\rangle$)

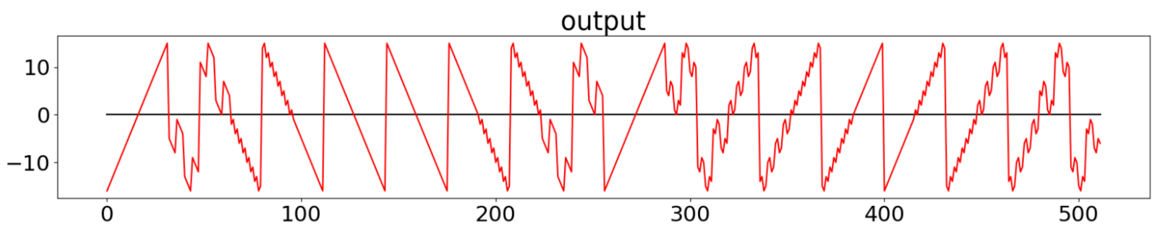


(b) QAIE($|A_2\rangle$,$|A_1\rangle$)

Figure 4.15: Example 6b SQPAM



(a) QAIE($|A_1\rangle$,$|A_2\rangle$)



(b) QAIE($|A_2\rangle$,$|A_1\rangle$)

Figure 4.16: Example 6c (QSM)

Case-Study 1	**Input Audio**	**QRA**	**Input size** [qubits]	**Wavetable size** [samples]	**Reading Speed** [Hz]	**Delay between experiments** [ms]
Example 0	$\sin(2\pi t)$; $t \in [0,1)$	QPAM	6	128	350	-
Example 1	$\sin(2\pi.5.t)$; $t \in [0,1)$	QPAM	7	256	350	0
Example 2	$\sin(2\pi t)$; $t \in [0,1)$	QPAM	8	512	250	200
Example 3	$\sin(2\pi.100.t)$; $t \in [0,1)$	QPAM	10	1024	1	1000

Case-Study 2	**Input Audio**	**QRA**	**Input size** [qubits]	**Wavetable size** [samples]	**Reading Speed** [Hz]	**Rotation param.** [radians]
Example 4	$\cos(2\pi t)$; $t \in [0,1)$	SQPAM	$4_{t_{reg}}+1_{a_{reg}}$	32	250	$\rho(k)=\frac{\pi k}{10}$; $k \in \{0,1,...,20\}$

Case-Study 3	**Input 1**	**Input 2**	**QRA**	**Input size** [samples]	**Wavetable size** [samples]	**Shots**	**Read. Speed** [Hz]
Example 5	Array 1 (Table 4.2)	Array 2 (Table 4.2)	QPAM	$16_{(A_1)}+16_{(A_2)}$	32	4000	250
Example 6	$\sin(2\pi t)$; $t \in [0,1)$	$t-\lfloor t \rfloor$; $t \in [0,1)$	QPAM	$16_{(A_1)}+32_{(A_2)}$	1024	500000	250
			SQPAM			200000	
			QSM			10000	

Table 4.1: Demonstrations parameter reference table

Case-Study 1	**List of *shots* for each experiment (total of 32)**
Ex. 0	[1024]
Ex. 1	[1, 10, 100, 500, 1000, 10000, 100000]
Ex. 2 and 3	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 1000, 2000, 3000, 4000, 5000, 10000, 20000, 50000, 100000]
Case-Study 3., Ex.5	Input Arrays
Array 1	[0, -1, 1, 2, -2, -4, 2, 3, 0, -4, -2, 1, 1, -1, 0, 0]
Array 2	[3, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 0, 3, 0, 0, 0]

Table 4.2: Caption

Chapter 5

Artistic Outputs: The Quantum Itinerary of Sound

This chapter will take an look at the practical music implementations of some of the concepts explored so far. The focus now shifts to a description of the artistic results, accomanied by some relevant aesthetic decisions that seeked to give a sense of meaning to the process of composing with quantum representations of audio, particularly, focusing on quantum measurement. Two practical applications will be described, in the form of music compositions, or artistic studies: *Rasgar, Saber* and *Rever*; respectivelly, an acousmatic piece and a directed improvisation process.

A stereo reduction of Rasgar Saber can be found published on YouTube (Itaboraí, 2023b), as well as a recording from one of the initial improvisation sessions of *Rever* (Itaboraí & Vicente, 2023). Some examples in this chapter mention audio files available in the accompanying media folder. Refer to the index for information about the sound label number and its respective file name in the folder.

Precious Chance

To provide an aesthetic endorsement for the process of projective measurements highlighted by the Geiger-Counter Effect, Duchamp's *Three Standard Stoppages* (fig. 5.1) can help justifying the inherent randomness of the result. Tavares will write about this work by affirming: "*Chance is conserved as if it were **precious**, and more: it becomes the new reference point*"†(Tavares, 2021, p. 31).

In analogy, every time one takes measurements from a QRA to reconstruct a signal, they would be essentially throwing a string into the ground, rewriting the definition of an audio file and preciously storing it.

Figure 5.1: Duchamp's Three Standard Stoppages, exhibited at Tate Modern London, Jun 2023

What is a Quantum Musical Instrument? - A Quantum Itinerary of Sound

By reading through a collection of Walter Smetak's texts and anecdotes, the following passages are brought to attention:

> *"THE HUMAN, INSTRUMENT AND SOUND [...] the instrument is also a vehicle of the sound. The knight runs on the horse, but the horse is the one actually moving, the knight drives the horse."*[(†)] (Smetak, 1974/2019, p. 234)

> *"Sonic Plastics: [...] These instruments come from a process of acoustic research in which the ITINERARY OF SOUND was examined, in its spititual, psychic and finally physical origin. The forms in general express a symbolism, a language, which approaches a world of static forms."*[(†)] (Smetak, 1967/2019, p. 19)

This paints the notion that a musical instrument's function is to Carry sound, bearing it, transforming it, and guiding it between distinct media, until it reaches a musical ear. With the emergence of quantum computing, a new concept could be coined, one that describes a QUANTUM ITINERARY OF SOUND.

5.1 Rasgar, Saber

The first outcome of Smetak's passages was to ventilate the idea of having *static forms that express a simbolism*. In other words, together with Smetak's persistence on introducing sculpture and plastics into instrument design motivated the idealisation of an acousmatic study piece as a Sound Sculpture. In other words, the idea was to integrate and craft the notion of Quantum Representation of Audio into a visually auditory form. To that end, the main material chosen for the composition and to be a carrier of quantum audio was 'paper'.

5.1.1 Paper as Sonic Material

The paper as the central object of the piece was brought fourth by previous experience in *origami* practice. On the occasion of COVID-19 Lockdown, I attempted to fold a 4m x 4m Ryu Jin 3.5 [1]. Its folding process (still unfinished), summing up 150h of work, was documented - partially in video, partially as timelapse - for posterior use (Fig. 5.2).

Figure 5.2: Snapshot of a recording of the 4x4m RyuJin 3.5 used to extract sounds from paper

As a result, project's raw video recordings were recalled and the audio extracted from them. They

[1] Satoshi Kamiya: *World of Super-Complex Origami*. Sohimu, 2010. Ryu Jin 3.5 is a historical super-complex model considered to be a milestone in modern origami design. The hybridisation of traditional and tessellation techniques, the innovative transitions for multilevel box-pleating grids, and its singular assimetric design that folds into a perfectly simmetric shape crown this model as a masterpiece of its time.

became the main source material in the piece. The rhythmic crinkling, shuffling, and swiping sounds of the paper provided a great source for musical gestures. To multiply the materials, a Musique Concrete transformation approach was used, with a collection of noise correction tools avaiable in Izotope RX10; providing a set of radically processed and distorted results. The attempt was to achieve constitutive materials that were spectromorphologically similar to the results from chapter 4, particularly, the Geiger-Counter Effect[2], allowing for the creation of richer sound objects and relational material that incorporated the paper sounds with the QRA effect.

Additionally, some applications of the Variational Quantum Harmonizer (Itaboraí et al., 2023) were used to drive music spatialization, pitchshift and grain duration of a granular synthesiser. Although the research with VQH - which is also a quantum musical interface - lies outside the scope of this text, it is worth mentioning that the GCE was used primarily as a tool for crafting constituting material, whereas VQH (and consequently, granular synthesis) was more applied in structural directionalities and connections between gestures.

Paper as Poetics, Voice as Material The name of the piece originates from the following narrative, that depicts the process of quantum measurement poetically using paper (resonating with the GCE): *Imagine an empty Envelope from which its invisible contents are full of Meaning, Knowledge and undeniable Truth. On the top of the envelope, a note: - "Rasga-me, e Saberás" (Tear me, and you will Know).*

This subsequently aligned with a poem that I authored in 2017, translated below, named ESTILHAÇO (FRAGMENT). There are recordings of this poem through the voice of a lyric singer, Tomás Mistrorigo, which were made in 2019. The poem was read in several ways, with deliberate voice distortions and/or omission of vowels at times. These materials were also used as source for *Rasgar, Saber*.

[2]At the time of music writing, the QAIE demonstration had not yet been developed.

Original	*Translation*
ESTILHAÇO	*FRAGMENT*
Antes,	Before,
o papel branco	the white paper
só.	only.
Unidade.	Unity.
Só	Alone (/such)
que pra um ser muitos,	that for one to be many,
que é o natural,	which is what is natural,
dividiu-se	has been divided
dobrou-se	duplicated
fragmentou-se	fragmented
e estilhaçou por inteiro	and shattered in its entirety
Só	Just (/alone)
pedacinhos de papel branco	pieces of white paper
retalho rasgado roído raspado	torn shred scraped snips
unicamente estilhaço.	uniquely shattered.
Só	Only
depois,	then,
como é natural	naturally
pegou cada pedaço,	picked up each piece,
juntou um a um,	patched them together one by one,
reformou disformou informou	reformed disformed informed
não em papel branco,	not into white paper,
Madeira.	Wood.
estilhaços de papeis brancos.	pieces of white paper.
Multidade.	Multity.

Paper (and voice) as quantum audio

In this section, a selection of prominent examples of materials processed with `quantumaudio` for the piece are shown. Most generated materials used the Geiger-Counter Effect. For these cases, the method was to save each individual step of the GCE (i.e, signals decoded with varying amount of shots) into a separate buffer and load them into a 2D Wavetable Synthesis (Code 5.1. The buffer selection for the synthesis can then be controlled via mouse, using the "`MouseX`" UGen.

```
SynthDef(\qTable2D,
    { |minbufn = 0, maxbufn = 10, freq = 250.0, gain = -6.0, out = 0|
        var sig, buf;
        buf = MouseX.kr(minbufn, maxbufn);
        sig = VOsc.ar(buf, freq); //2D Wavetable, reading from a buffer list
        sig = LeakDC.ar(sig); // Avoid excessive fluctuation and digital distortion
        sig = LPF.ar(sig, 12000); //Low-pass Filter for aliasing purposes
        Out.ar(out, sig*gain.dbamp!2); // Stereo Output
}).store;
```

Code 5.1: 2D Wavetable Synthesis example in SuperCollider

1. First, a short snippet of sound that contained two very clear rythmic transients from flicking paper (listen to **Sound 5.1**) was encoded using a 12-qubit QPAM for a Geiger-Counter Effect.. Interestingly, for a low number of shots, the measured results have aligned with the rythmic attacks of the original audio. The output for a reading speed of 2Hz can be heard in **Sound 5.1b**. This sound in particular was ultimately not included in the composition.

2. Another result not included in *Rasgar, Saber* was an application of the R_Y Rotations in a 8-qubit SQPAM (including a_{reg}). The input was a short section of a pronounced vowel (**Sound 5.2**), and the rotation steps are the same as in Case Study 2 (Table 4.1), except $k \in \{1,2,...,10\}$. The result is shown in (**Sound 5.2b**), where each rotation step represents three seconds of sound.

3. **Sound 5.3** - Is a 12-qubit version of **Sound 4.3** from section 4.3.1; but reversed. It is found at 4:22 in the piece.

4. A short extract from Mistrorigo's voice reading the word "*Rasgado*"[Torn apart] (**Sound 5.4**) was

processed in a 15-qubit GCE, and resynthesized with a 2D wavetable and reading speed of 350Hz (see **Sound 5.4b**) appears at 2:59.

5. A low-frequency oscillator (LFO) went through a 12-qubit GCE, and the synthesized wavetable was used as a control signal for amplitude modulation. The carrier signal was a sinusoidal note at 390Hz. The reading speed of the modulation was 7Hz (**Sound 5.5**).

6. *idem*, but the carrier signal was a harmonic entity (chord) made with additive synthesis[3]. For a low reading rate ($\approx$ 3Hz), interesting rhythmic patterns emerge (**Sound 5.5b**), appearing in two separate moments (1:25, 8:06). With an audible reading rate, it becomes an example of *ring modulation*; **Sound 5.5c** is a recording of an improvisation in which the 2D wavetable was played at different reading frequencies. This improvisation is prominently heard at the beginning of the second half of the piece (5:36).

7. *idem*, but the carrier signal is a full read of the *Estilhaço* poem, where only the consonants were pronouced (**Sound 5.6**). Several outputs were generated, with varying wavetable speeds. One example (33Hz) can be heard in **Sound 5.6b**. They were used in two moments (1:13, 7:55).

Paper as Sonic Plastics

To convey the visual bonding of the project, a mechanism was incorporated to be coupled with paper - Electro-Mechanical Transducers (EMT) (fig. 5.3) - forming a Sonic Plastics with origami pieces that emit sound. In summary, this operates as a custom speaker, that purposely dirsorts the outcoming sounds according to the resonances and mechanical properties of the paper that is coupled. As a result, the piece was divided into two different planes. One plane would have four traditional speakers (the digital realm, whereas the other consists of four origami-speakers (the quantum realm), forming a Sonic Plastics that dialogues with the alegory of Duchamp's Stoppages. There is a composed directionality in *Rasgar, Saber* in which the predominance of sounds that were processed by GCE, as well as the density of objects spatialised onto the paper in relation to the classical speakers increases, until the only sounding drivers are the origami-speakers, by the end.

The paper as sonic plastics promotes a strong interrelation between the original material, the poetical meaning, and the technical aspects. This is reinforced by the central origami model, which was designed to be a mask. In result, it is not only the paper sounds that are materialising through paper, but a voice that regains a face. Moreover, it further distorts and interferes with the final auditory

[3]Chord Notes: C3; G#3; G4; A4; Bb4; B5; D6; F6; F#6; C#7; E7; F#7; A7

Figure 5.3: *The Ear* origami-speaker prototype. 2022

Figure 5.4: Two of the Origami-Speakers. The Voice & The Ears, Respectively. Sonic Plastics by Paulo Itaborai, 2023.

result, proposing an ITINERARY. For instance, examining the path of Sound 5.6 described above, the first distortion layer happens when Mistrorigo reads the poem, filtering out the vowels from his pronunciation. Then, it gets modulated by a LFO that has been processed by GCE; this filters the continuity of the sound, as it gets scattered by the discontinuous modulating signal. Then, it is filtered by the mechanical impulse response of the origami-speaker and is materialised by the paper. It can be argued that the quantum element was successfully incorporated in this ITINERARY, in an artistic sense, which approaches**Research Question 3:** *How to put in practice the use of Quantum Computing in a Musical Artistic Process and/or Live Performance?*.

5.2 Rever

The proximity of the results of applications of QRA with analog sounds led me to reach Dino Vicente [4]. Our artistic collaboration - which spans over many projects, beyond the scope of this work - would nourish a long dialogue and a rich artistic process on the early investigations of quantum computer music, starting from the artistic lens. The result of this interaction is an ever-changing, ever-in-contruction technical-artistic improvisation, which we named "Rever".

The piece revolves on analog synthesis, by means of using a 1974 Moog 15 synthesizer. Dino has a model of this unique instrument at his studio in Brazil, which is being kept in impecable shape for almost 50 years. Even while containing a limited set of oscillators, filters and tones, the generated sounds are spectrally rich, and the timbre, deep. If one attempts to reproduce this timbre in a music programming laguage, it may be argued that it could probably lead to months of careful sound design. In contrast, this is a seamless job for Moog 15 [5]. The central question of this collaboration then became: *How would the wavetable synthesis, used as the carrier of Quantum Representation of Audio, sound like in a Moog 15?*

Being Dino himself very knowlegeable of Smetak's work, it didn't take long until this project was approached as an improvisational process that aims to achieve a Sound Plastics. And in regard to the coined concept of QRA being a QUANTUM ITINERARY OF SOUND, this process could be a textbook example: *From Quantum to Analog*.

5.2.1 The Itinerary

The *blueprint* of the itinerary is described in fig. 5.5. It is designed to have a full, inter-connected system where Moog 15 (the analog), music programming languages (the digital), and the `quantumaudio` package (the quantum), are exchanging information and creating musical gestures, dialogues, and narratives. Although this blueprint describes a general musical instrument, without restricting the project to QRA in the long term, this is where the short-term focus was centred.

During a period of 7 months where this project intersected with the mastars program (Jan 2023- Jul 2023), it was possible to produce a first prototype. Results generated on the GCE were passed over

[4] Dino is a composer-producer, music experimentalist, pioneer, and early adopter of the use of modular synthesizers in the brazilian electronic music scene.

[5] It is important to highlight this fact, as one of the focuses of this work is to investigate a new media for audio signals. In other words, after the field of QRA starts to solidify more, with a larger portfolio of encoding schemes and QSP algorithms, it will become clearer which techniques (and which aesthetical sounds) are more convenient to be deployed in a quantum processor, compared to other media.

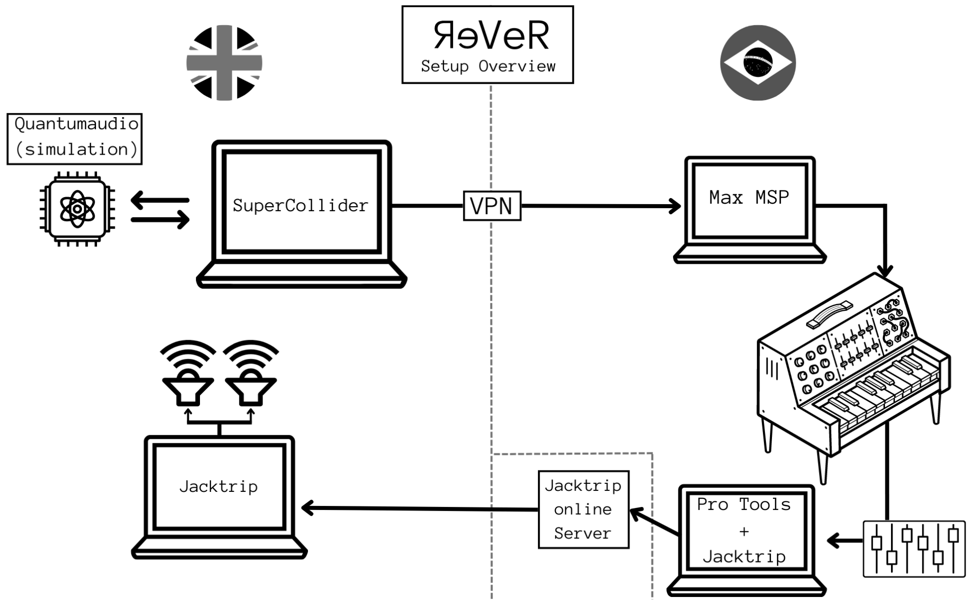


Figure 5.5: Simplified Diagram for *Rever*

a VPN network to Brazil, using OSC protocol. A viable example of OSC communication for QCM between Sao Paulo and Plymouth had already been tested in a previous work (Hamido & Itaboraí, 2023). There, the results were loaded into a wavetable synthesis in Max/MSP (in a similar way to that introduced with SC in chapter 4), and ultimately used as analog control signals to be added to the Moog 15 patch. Finally, the synthesized sounds would find their way back to Plymouth by means of using a Telematic Music tool, such as JackTrip WebRTC (Sacchetto et al., 2021).

After a period of technical implementation and testing - which involved validating the connection and the viability of remote rehearsals and performance - a set of initial improvisations was carried in the month of July, one of which was documented in video (Itaboraí & Vicente, 2023). A snapshot of this recording is depicted in Fig. 5.6. On the bottom right corner, it is possible to see a visualisation of a wavetable in Max, which is being updated by hand gestures (top left corner) that could trigger and send the results of a new QRA simulation. There is also a visualisation of the same wavetable on the top left, in which the shape was reinforced in fig. 5.6

5.2.2 The Patch

The general layout of the mapping between quantum and analog synth is depicted in Fig. 5.8

On the British side, the `quantumaudio` package was operated in a Jupyter Notebok to generate two wavetables from a set of three basic waveforms (Sine, Square and Sawtooth) using GCE. The main GCE has 9 qubits (512 samples), and a secondary has 3 (8 samples). As seen in section 4.2, the results were loaded into a buffer in SuperCollider. Optionally, the second buffer could also be manipulated

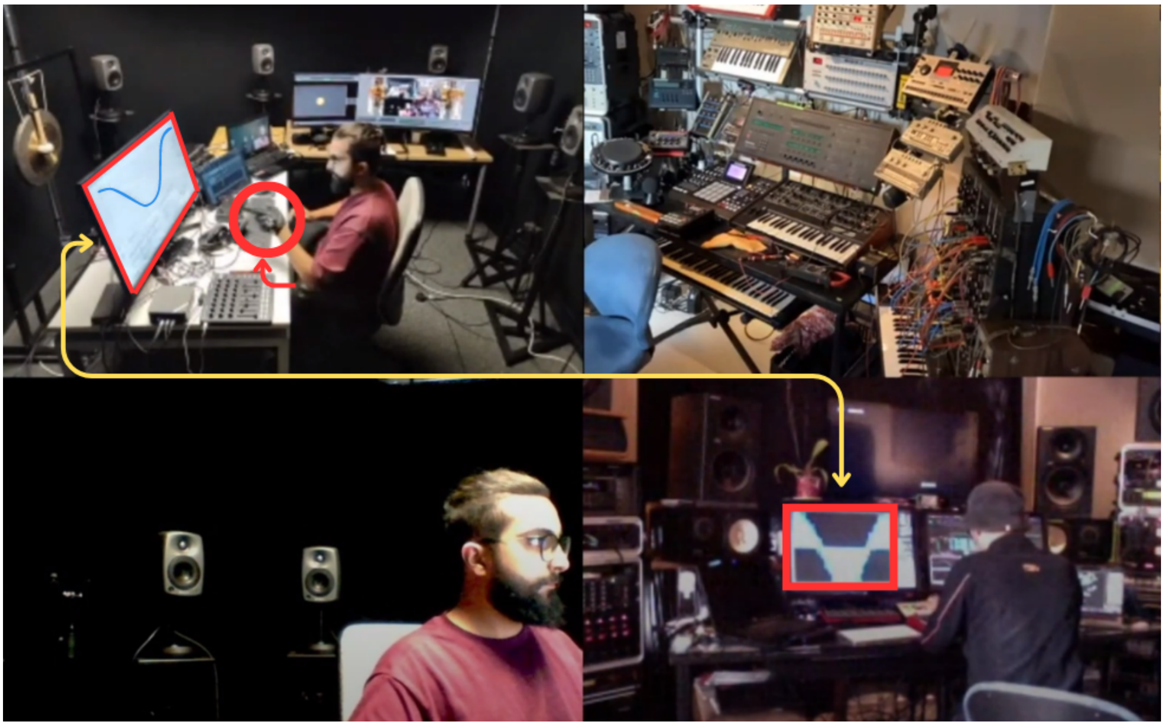


Figure 5.6: Snapshot of an improvisation session. On the left, ICCMR Studio, Plymouth. On the Right DVM Studios, São Paulo

via MIDI. Instead of instantiating a synthesizer, this data, now in SC, was relayed to Dino via OSC.

On the Brazilian side, there is a Max/MSP patch (designed by the author) connected to the network awaiting for new data. Incoming information causes Max to update two buffers, linked to oscillators (WS1 and WS2). The first is the one seen in Fig. 5.6 and is directly synthesized and sent to Moog 15.

The second wavetable was slightly diffent. Instead of controlling the shape of the oscillator **WS2**, it was shifting the frequencies of 8 sound partials, using a harmonic proportional mapping together with additive synthesis. In other words, the information transmitted from Buffer 2 to WS2 is used to shift, proportionally, the position of a note from n-th partial of a harmonic series (5.1). The fundamental frequencies of both oscillators was controlled by Dino.

$$\mathrm{freq}_n(t) = (n - c_n(t))\mathrm{freq}_1 \tag{5.1}$$

Finally, the signal reached Moog 15. in general terms, for **WS1**, the synthesized signal was fed into a splitter. Subsequently, the multiplied output was used to modulate two parameters of the resulting synthesis: i) The cutoff frequency of a Low-Pass Filter; ii) The frequency of Moog's Oscillator 3.

For **WS2**, the signal was also directed to a splitter, where one of the outputs was relayed to the sound mixer. and heard. The other splitter output acted as a control to *synchronize* the phases of Moog's Oscillators 1 and 2. The remaining of the analog patch, that is, what happened to Oscillators 1, 2 and 3

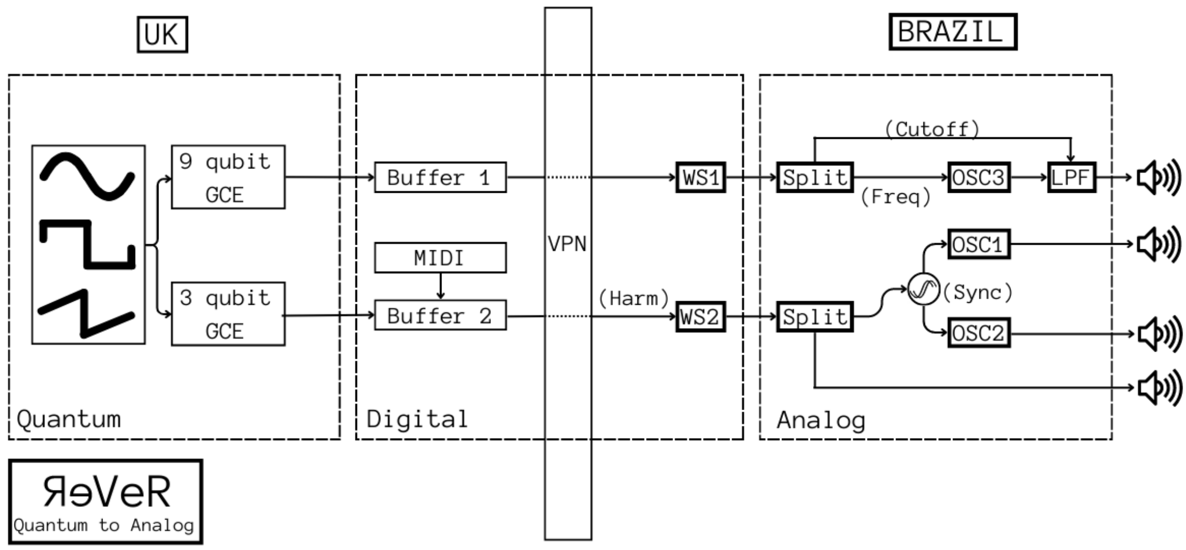


Figure 5.7: Max/MSP Patch

after their generation, and any unused synthesis parameters constituted where Dino had the artistic liberty to intervene and create more complex patches during the improvisation.

As a result, the GCE was used to drive the synthesis of Moog 15.

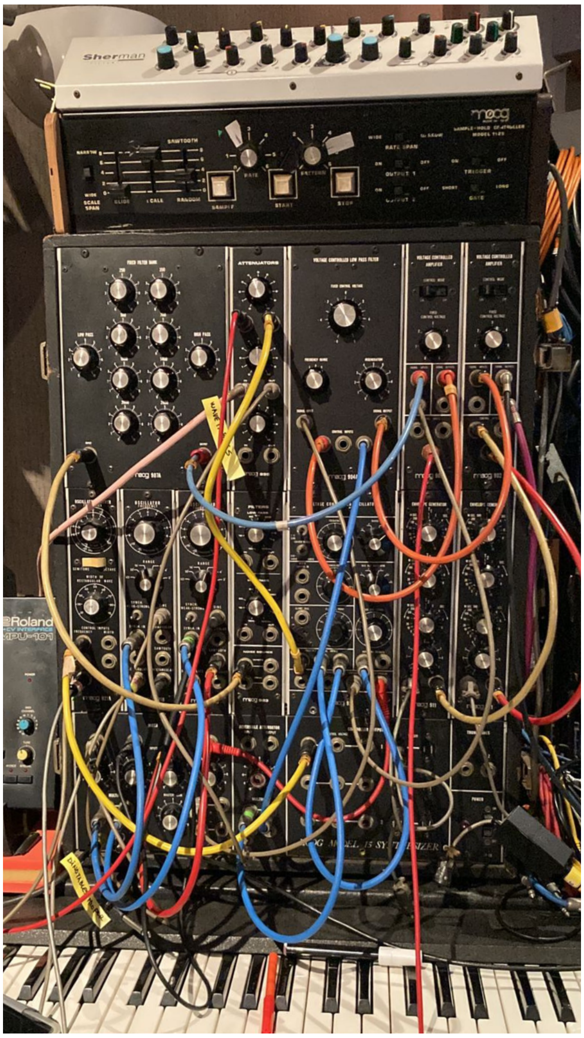


Figure 5.8: Photo of Moog 15's patch

Chapter 6

Concluding Discussion

6.1 Reflections on Quantum and Music

In this section, there will be commentaries on the creative process, and the insights gained from the challenges of attempting to conciliate theory and practice. The experiences documented here serve as both a summation of the research journey and a potential guide for those who may follow in these exploratory footsteps of binding technical and artistic scopes together.

6.2 Reconciling the irreconcilable

Initial efforts to integrate simple quantum audio manipulations into composition led to a 5-month period of creative halt on *Rasgar, Saber*. On reflection, a number of factors may have contributed to this stagnation.

By engaging in other technical implementations and investigations within the quantum computer music group in ICCMR, there was an opportunity for exploring other subfields of Quantum Computer Music, such as sonification - the Q1Synth Instrument (Miranda et al., 2023) and later the Variational Quantum Harmonizer (Itaboraí et al., 2023) - as well as quantum-computer assisted composition - continuing[1] the QuSing project (Miranda & Siegelwax, 2022) - both led to compelling collaborations that were presented at Goethe Institut (Sounding_Qubits, 2023).

A plausible conclusion (which would ideally require a detailed meta-analysis to confirm) is that projects focussing on control and mapping problems would allow for a faster deployment into live performances. This reflection - which leads to a future research hypothesis - is supported by

[1] At the time of writing, this continuation is still unpublished work.

the fact that the choice of synthesis in a sonification/mapping project, for example, is ultimately a creative choice. Although QRA could arguably be a specific type of histogram sonification - or on a different note, used in a complex mapping setup, where the signal is interpreted as a serial value for a structural composition parameter or musical form - this contrasts with the approach taken in this text, where the process of encoding and decoding sounds had a central stage, together with lower-level audio generation and manipulation. This may be precisely the difference between what is formally distinguished as Relational Material in one end, and Constituent Material in the other (). Put differently, there was a focus in this project to investigating quantum audio as a constituent material.

A logical conclusion would be that only by studying how to constitute sound in a quantum machine, there could be room for the origination of new musical instrument frameworks that are exclusive to quantum technologies: QUANTUM ITINERARY OF SOUND. Even so, as observed with *Rever*, musicians and instrument designers need to achieve a confortable level of experience working with an ITINERARY before being able to achieve performative musical narratives during improvisation.

However, before getting too inspired by an unheard quantum world, it is worth reminding that there was already a historical period in which organised sound was dissected to its atom, as mentioned in the introduction. Hence, it is also a possibility that, in the end, the artistic results from QRA will push sounds towards a reinvention/rediscovery of analog electronic music. Even then, it may not cause a crisis in music composition. As Menezes would have complemented: *"Composing therefore means re-composing. Thus, there are no composers, but merely re-composers"*[(†)](Menezes, 2013, p. 25). Re-invention and re-composition are, ultimately, the core activities of the composer. This is also what partly originates the name of the piece, *Rever*, in addition to a concrete poem from Augusto de Campos of same name.

6.2.1 Training our ears

In one of his texts entitled "Statistic and Psychologic Problems of Sound". Meyer-Eppler highlights the importance of eletronic sound experiments - carried out in the early years of electronic music - in the way we listen to sounds today. "*Research into electrical methods of generating sound or noise has revealed a large number of phenomena which can only be discovered in instrumental sounds after the ear has been prepared by electroacoustic experiments.*"[(†)](Meyer-Eppler, 1958). I may extrapolate his point to a broader sense, arriving at the question: How new technologies might tamper with our forms of listening to music?

We may find a good illustration by looking into a piece for piano: *Klavierstücke XI* Stockhausen (1957).

In simple terms, the pianist randomly looks and picks one of 19 score measure sketches and plays it.

At the end there are strict instructions on how to play the next measure (e.g. with a certain intensity - pianissimo or mezzoforte, etc. - and/or articulation - staccato - and/or tempo). Then, after navigating around the paper and reaching the same measure for the third time, the piece ends. This means that, effectivelly, the piece could be completely different and unique every single time it is performed. Now, consider the following exercise: That the same 19 measures are now in a *quantum superposition*. The quantum circuit that represents that score has several *couplings* and relational complexity. Then, the task of the pianist would be to make successive *projective measurements*, impacting the configuration of the score.

With that in mind, it is possible to change one's relationship to that piano piece, placing oneself in the position of a listener of *collapsed piano configurations* - which reinforces the point raised by Meyer-Eppler and also suggests a kind of musical form that may be conveniently formulated in terms of Quantum Computer Music. Nevertheless, whether the practical and aesthetic deployments of quantum algorithms in music composition will have a significant impact - compared to classical methods - remains to be seen.

This could be left as an open research questions for future work: If quantum-inspired music already exists since the very discovery of quantum mechanics, what makes quantum computer music different? Would we not be just reproducing the randomicity present in the serialist movement and the relationship with noise present in stochastic music? How could QC change the way we hear sounds?

6.2.2 *Rasgar, Saber* - Final notes: The rupture transition

To solidify the aesthetic narrative built in this study, it is possible to draw inspiration from *Kontakte* (Stockhausen, 1966). This piece represented a historic transition in the compositional process of Stockhausen and an overall disruption in the electroacoustic music scene. In his narrative, this rupture could be summarised in the iconic transition between the two movements of the piece (17:50). There, Stockhausen expressed a concept of Universal Harmony, binding rhythmic and pitch perception together. A continuous train-pulse sound that keeps being slowed down, until it breaks the perceptual continuum of pitch and becomes a rythmic pattern. Then, it is further slowed down, until the timbral characteristics of the percussive attacks are perceived, arriving again at a continuous sound.

As a result, *Rasgar, Saber* was also divided in two sections, with a clear disruptive transition in the mid-point (fig. 6.1). In this case, the "Universal Harmony" was performed by the Geiger-Counter effect (Sound 5.3), exploring the transition between the continuous sound (the original sinewave) reaching noise, and finally, the decreasing geiger-counter clicks. At this *rupture* moment, the poetic

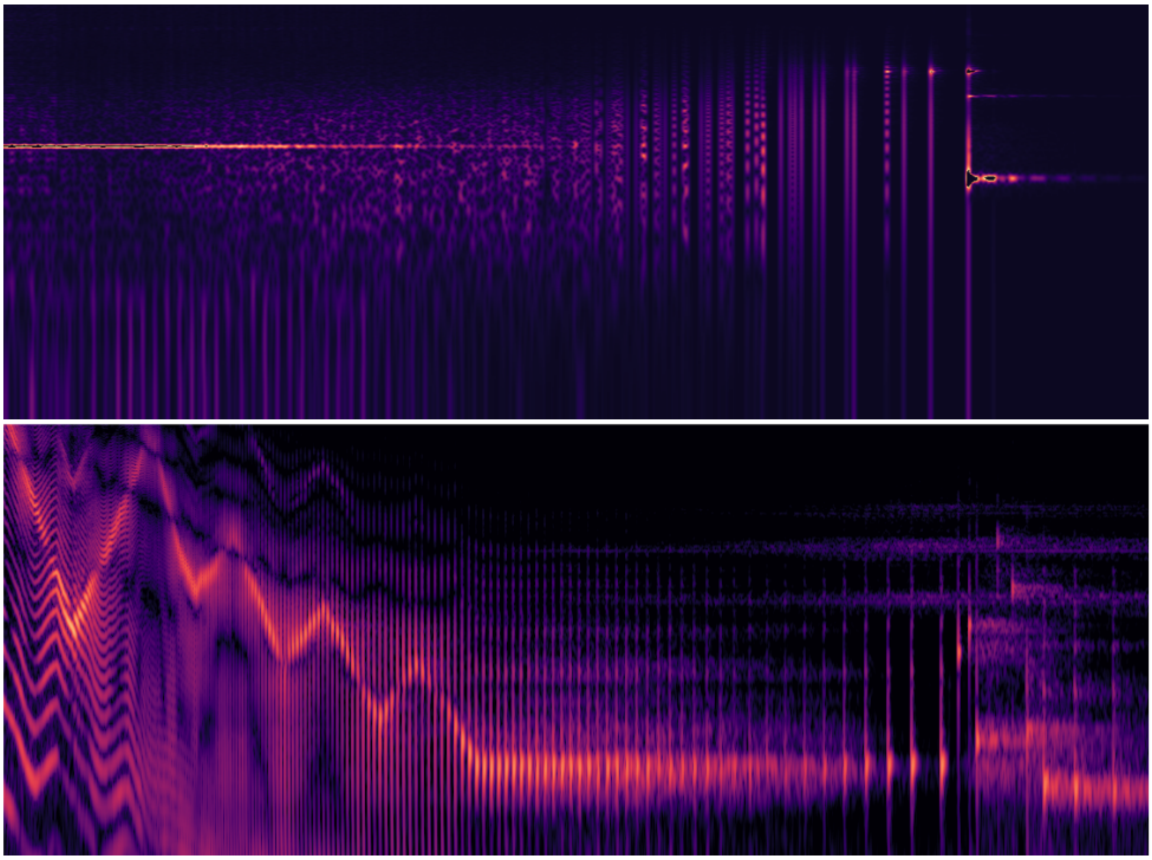


Figure 6.1: Spectrogram Comparison between the Rupture (above) with the Universal Harmonics of Kontakte (below)

"*Envelope*" would be torn apart, in a moment of shocking revelation - *electronic music, recomposed*.

Additionally, this transition is also directly reflected in the form of the origami sculptures, since the models are, too, *literally torn apart* on live stage at that moment, visually completes the proposed rupture. Although not present in the electronics-only recording, this moment was documented in video (see **Video 6.1**).

After being desfigurated, the sounds got further distorted when they passed through the torn origami-speakers. To enchance the effect for the second half of the piece, an additional LFO was introduced to cause the paper to vibrate at one of its resonant frequencies (see **Video 6.2**).

6.3 Concluding remarks

In this work, the notion of Quantum Representation of Audio was used as the guiding thread for exploring the ealy adoption od Quantum Computing technologies in artistic expression. In the literature review, the **Research Question 2:** *How to encode Digital Audio in Quantum Circuits?* was addressed. The published work Itaboraí & Miranda (2022) was revisited, leading to a systematic survey on QRA that was able to better classify and categorise different techniques. This was also presented with a unified notation, and a didactical framework aiming to reach researchers in the Computer Music and Digital Signal Processing fields. In the new system devised, one additional QRA strategy was proposed by the author (SQPPM, Sec. 2.8.2), as a natural follow-up from the work of Grigoryan, using the same approach proposed by Le et al. (2011) (SQPAM) for quantum images. In addition, it was shown that SQPPM signals can be converted into a variant of SQPAM by applying only two instructions: $Z\sqrt{X}^{\dagger}$. This implies that chapter 2 has brought potential unprecedented contributions to the field of QCM.

Then, to address **Research Question 1:** *How to use Quantum Computing as a Musical Instrument?*, three representations (QPAM, SQPAM, QSM) were implemented in a preliminary Python package, `quantumaudio` (chapter 3), that allows a quick operationalisation for encoding and decoding audio signals in quantum simulation. Its API structure permits that an end-user of the package to work with QSP without specialised knowledge of QRA. To demonstrate this point - moving towards an artistic scope - a series of demonstrations were carried out in chapter 4 where a number of heuristic "quantum audio effects" suggested potential applications in electronic music. A structure design for a musical interface was developed to map the recontructed signals into the waveform of wavetable oscillators.

Finally, addressing **Research Question 3:** *How to put in practice the use of Quantum Computing in a Musical Artistic Process and/or Live Performance?*, one of the effects discussed, GCE, was incorporated in the process of two artistic results, leading to a connection between quantum measurement and early-day analog electronic music aesthetics.

Drawing from the list above, the aspects of the artistic methodology being applied in the music composition are listed:

- Investigating and improvising with new techniques
- Creating sound results by observing (measuring) the quantum states that represent audio in the quantum machine.
- Explore basic mechanisms of digital sound creation to interface quantum representations of audio with sound synthesis.

Acronyms

$\mathtt{a}_{\mathtt{reg}}$ amplitude register. 21–27, 29, 32, 36, 37, 61–63, 65, 73, 90

$\mathtt{c}_{\mathtt{reg}}$ channel register. 32–34, 63

$\mathtt{t}_{\mathtt{reg}}$ time register. 16, 21, 24, 26, 27, 29, 32, 35, 40, 62, 63, 65

RQ1 **Research Question 1:** *How to use Quantum Computing as a Musical Instrument?*. 4, 7, 52, 85

RQ2 **Research Question 2:** *How to encode Digital Audio in Quantum Circuits?*. 6, 84

RQ3 **Research Question 3:** *How to put in practice the use of Quantum Computing in a Musical Artistic Process and/or Live Performance?*. 9, 75, 85

scsynth SC Server. 54

AM Amplitude Modulation. 13

CB Coefficient-Based. 17

DFT Disctete Fourier Transform. 36

DSP Digital Signal Processing. 7, 26, 44

EMT Electro-Mechanical Transducers. 74

FM Frequency Modulation. 13

FRQA Flexible Representation of Quantum Audio. 26

FRQI Flexible Representation of Quantum Images. 21, 30, 32

FTQR Fourier Transform Qubit Representation. 30

GCE Geiger-Counter Effect. ix, 61, 68, 71, 73–77, 79, 85

MPL music programming language. 53

MQSM Multichannel Quantum state Modulation. 32, 40

MSQPAM Multichannel Single-Qubit Probability Amplitude Modulation. 33, 34, 41, 42

NEQR Novel Enhanced Quantum Representation for digital images. 26, 27

NISQ Noisy Intermediate-Scale Quantum Computig. 2, 39

OSC Open Sound Control. 53, 54, 77

PAM Pulse Amplitude Modulation. 14

PCM Pulse Code Modulation. 14, 27, 34

PM Phase Modulation. 13

PMQA Probability amplitude-encoded Multichannel representation for Quantum Audio signals. 33

PPM Pulse Position Modulation. 14

PSQPAM Partitioned Single-Qubit Probability Amplitude Modulation. 34

PWM Pulse Width Modulation. 14

QAIE Quantum Audio Interference Expansion. vii, 62–65, 71

QC Quantum Computing. 52–54, 83

QCM Quantum Computer Music. 1, 2, 42, 77, 81, 83, 85

qDCT Quantum Discrete Cosine Transform. 42

QFT Quantum Fourier Transform. 36, 39, 42

QLM Qubit Lattice Model. 16, 31, 35, 42

QPAM Quantum Probability Amplitude Modulation. vii, 23, 39, 42, 45, 50, 51, 55, 58, 64–67, 73, 85

QPE Quantum Phase Estimation. 36, 37

QPPM Quantum Probability Phase Modulation. 35, 36

QRA Quantum Representation of Audio. vii, 6, 7, 9–12, 14, 16–18, 21, 24–26, 32, 37–39, 42, 44–48, 52, 55, 62–65, 67, 68, 70, 71, 76, 77, 82, 84, 85, 90

QRDA Quantum REpresentation of Digital Audio. 26

QRDS Quantum Representation of Digital Signals. 27

QRI Quantum Representation of Images. 6, 11, 12, 16, 17, 26, 29, 30, 32, 34, 39, 41, 44

QSM Quantum State Modulation. vii, 26, 27, 29, 32, 39, 40, 42, 45, 50, 51, 65–67, 85, 89

QSM-fl QSM - Floating-Point Variant. 27

QSM-fp QSM - Fixed-Point Variant. 27, 42

QSM-s QSM - Signed Integer Variant. Also written as just QSM. 27, 40, 45

QSM-u QSM - Unsigned Integer Variant. 27, 42

QSP Quantum Signal Processing. 6, 7, 10, 11, 26, 38, 39, 42, 44, 47, 76, 85

SB State-Based. 17

SC SuperCollider. 7, 45, 53–58, 61, 77, 78, 87

SQPAM Single-Qubit Probability Amplitude Modulation. vii, 21–25, 33, 34, 37, 45, 50, 51, 61, 62, 65–67, 73, 85, 89

SQPAM-c SQPAM - Cartesian Variant. 25

SQPAM-t SQPAM - Trigonometric Variant. Also written as just SQPAM. 45

SQPPM Single-Qubit Probability Phase Modulation. 36, 37, 90

STE Standard `quantumaudio` Toy Example. vii, 48–51

TQPAM Two-Qubit Probability Amplitude Modulation. 33

VQH Variational Quantum Harmonizer. 71

VSO Value-Setting Operation. vii, 22, 24, 25, 28, 35, 36, 63

WDR Westdeutscher Rundfunk. 2

WS Wavetable Synthesis. 7, 55–57

Appendix A

Expanded Calculation of the Quantum Audio Conversion in section 2.8.2

Let us verify how to arrive at the result shown in equation (2.42), since it does not appear elsewhere in the QRA literature.

Starting from an SQPPM, we apply a $\sqrt{X}^{\dagger}$ Gate at the amplitude register:

$$\left|A_{SQPPM}\right\rangle = \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1} \left(|0\rangle + e^{i\lambda_k}|1\rangle\right) \otimes |k\rangle \tag{A.1}$$

$$\sqrt{X}^{\dagger} \otimes I^{\otimes n} \left|A_{SQPPM}\right\rangle = \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1} \sqrt{X}^{\dagger} \left(|0\rangle + e^{i\lambda_k}|1\rangle\right) \otimes I|k\rangle \tag{A.2}$$

Changing the middle part to matricial form:

$$\sqrt{X}^{\dagger} \otimes I^{\otimes n} \left|A_{SQPPM}\right\rangle = \frac{1}{\sqrt{2N}} \sum_{k=0}^{N-1} \underbrace{\frac{1}{2} \begin{pmatrix} 1-i & 1+i \\ 1+i & 1-i \end{pmatrix} \begin{pmatrix} 1 \\ e^{i\lambda_k} \end{pmatrix}}_{\zeta_k} \otimes |k\rangle \tag{A.3}$$

Developing the calculations of ζ_k:

$$\zeta_k = \frac{1}{2} \begin{pmatrix} (1-i) + (1+i)e^{i\lambda_k} \\ (1+i) + (1-i)e^{i\lambda_k} \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \left(1 + e^{i\lambda_k}\right) - i\left(1 - e^{i\lambda_k}\right) \\ \left(1 + e^{i\lambda_k}\right) + i\left(1 - e^{i\lambda_k}\right) \end{pmatrix} \tag{A.4}$$

Using a common factoring strategy:

$$\zeta_k = \frac{1}{2} e^{i\lambda_k/2} \begin{pmatrix} \left(e^{i\frac{\lambda_k}{2}} + e^{-i\frac{\lambda_k}{2}}\right) + i\left(e^{i\frac{\lambda_k}{2}} - e^{-i\frac{\lambda_k}{2}}\right) \\ \left(e^{i\frac{\lambda_k}{2}} + e^{-i\frac{\lambda_k}{2}}\right) - i\left(e^{i\frac{\lambda_k}{2}} - e^{-i\frac{\lambda_k}{2}}\right) \end{pmatrix} \tag{A.5}$$

$$= \frac{1}{2} e^{i\lambda_k/2} \begin{pmatrix} 2\cos\left(\frac{\lambda_k}{2}\right) + 2\left(i^2\right)\sin\left(\frac{\lambda_k}{2}\right) \\ 2\cos\left(\frac{\lambda_k}{2}\right) - 2\left(i^2\right)\sin\left(\frac{\lambda_k}{2}\right) \end{pmatrix} = e^{\lambda_k/2} \begin{pmatrix} \cos\left(\frac{\lambda_k}{2}\right) - \sin\left(\frac{\lambda_k}{2}\right) \\ \cos\left(\frac{\lambda_k}{2}\right) + \sin\left(\frac{\lambda_k}{2}\right) \end{pmatrix} \tag{A.6}$$

$$\tag{A.7}$$

More Factoring:

$$\zeta_k = e^{i\lambda_k/2} \begin{pmatrix} \sqrt{2}\left[\cos\left(\frac{\lambda_k}{2}\right)\frac{1}{\sqrt{2}} - \sin\left(\frac{\lambda_k}{2}\right)\frac{1}{\sqrt{2}}\right] \\ \sqrt{2}\left[\cos\left(\frac{\lambda_k}{2}\right)\frac{1}{\sqrt{2}} + \sin\left(\frac{\lambda_k}{2}\right)\frac{1}{\sqrt{2}}\right] \end{pmatrix} \tag{A.8}$$

$$\frac{1}{\sqrt{2}} = \cos\left(\frac{\pi}{4}\right) = \sin\left(\frac{\pi}{4}\right) \quad \therefore \quad \zeta_k = \sqrt{2} e^{i\lambda k/2} \begin{pmatrix} \cos\left(\frac{\lambda_k}{2}\right)\cos\left(\frac{\pi}{4}\right) - \sin\left(\frac{\lambda_k}{2}\right)\sin\left(\frac{\pi}{4}\right) \\ \cos\left(\frac{\lambda_k}{2}\right)\sin\left(\frac{\pi}{4}\right) + \sin\left(\frac{\lambda_k}{2}\right)\cos\left(\frac{\pi}{4}\right) \end{pmatrix} \tag{A.9}$$

$$\therefore \zeta_k = \sqrt{2} e^{i\lambda_k/2} \begin{pmatrix} \cos\left(\frac{\lambda_k}{2} + \frac{\pi}{4}\right) \\ \sin\left(\frac{\lambda_k}{2} + \frac{\pi}{4}\right) \end{pmatrix} = \sqrt{2} e^{i\lambda_k/2} \left[\cos\left(\frac{\lambda_k}{2} + \frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\lambda_k}{2} + \frac{\pi}{4}\right)|1\rangle\right] \tag{A.10}$$

Therefore,

$$\sqrt{X}^{\dagger} \otimes I^{\otimes n} \left|A_{SQPPM}\right\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{i\lambda_k/2} \left[\cos\left(\frac{\lambda_k}{2} + \frac{\pi}{4}\right)|0\rangle + \sin\left(\frac{\lambda_k}{2} + \frac{\pi}{4}\right)|1\rangle\right] \otimes |k\rangle = \text{eq. (2.42)} \tag{A.11}$$

Q.E.D.

Bibliography

Ades, D., Neil, C., & Hopkins, D. (2021). *Marcel Duchamp (World of art)*. Thames and Hudson Ltd, 2 ed.

Allen, S., & Bryan, K. (2008). *Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing*. John Wiley & Sons.

Asaka, R., Sakai, K., & Yahagi, R. (2020). Quantum circuit for the fast fourier transform. *Quantum Information Processing*, *19*(8), 277. https://doi.org/10.1007/s11128-020-02776-5

Barthel-Calvet, A.-S. (2022). *Revisiting the Historiography of Postwar Avant-Garde Musicof Information Theory*. Taylor & Francis. https://doi.org/10.4324/9781315108162

Buck, A. K. (2018). *Fanstasia essata: computer as a musical device*. http://hdl.handle.net/11449/157503

Chaharlang, J., Mosleh, M., & Rasouli-Heikalabad, S. (2020a). A novel quantum steganography-steganalysis system for audio signals. *Multimedia Tools and Applications*, *79*(25-26), 17551–17577.

Chaharlang, J., Mosleh, M., & Rasouli Heikalabad, S. (2020b). Proposing a new and comprehensive method for quantum representation of digital audio signals. *Electronic and Cyber Defense*, *8*(4), 139–152. [Language: Persian].

Chen, K., Yan, F., Iliyasu, A. M., & Zhao, J. (2019). Dual quantum audio watermarking schemes based on quantum discrete cosine transform. *International Journal of Theoretical Physics*, *58*(2), 502–521. https://doi.org/10.1007/s10773-018-3950-9

Clemente, G., Crippa, A., Jansen, K., & Tüysüz, C. (2022). New directions in quantum music: Concepts for a quantum keyboard and the sound of the ising model. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 433–445). Springer. https://doi.org/10.1007/978-3-031-13909-3_17

Cline, D. (2019). Two Concepts of Indeterminacy in Music. *The Musical Quarterly*, *102*(1), 82–110. https://doi.org/10.1093/musqtl/gdz007

Cuthbert, M. S., & Ariza, C. (2010). Music21: A toolkit for computer-aided musicology and symbolic music data. J. S. Downie, & R. C. Veltkamp (Eds.) *ISMIR*, (pp. 637–642). International Society for Music Information Retrieval.

Dang, Y., Jiang, N., Hu, H., Ji, Z., & Zhang, W. (2018). Image classification based on quantum k-nearest-neighbor algorithm. *Quantum Information Processing*, *17*(9), 239. https://doi.org/10.1007/s11128-018-2004-9

Derrida, J. (1996). Remarks on deconstruction and pragmatism. In S. Critchley, & C. Mouffe (Eds.) *Deconstruction and Pragmatism*, (p. 84). Routledge.

Esposito, M., Uehara, G., & Spanias, A. (2022). Quantum machine learning for audio classification with applications to healthcare. *2022 13th International Conference on Information, Intelligence, Systems & Applications (IISA)*, (pp. 1–4).

Fletcher, N. H., & Rossing, T. D. (2012). *The physics of musical instruments*. Springer Science & Business Media. https://doi.org/10.1007/978-0-387-21603-4

Flusser, V. (1986). The photograph as post-industrial object: An essay on the ontological standing of photographs. *Leonardo*, *19*(4), 329–332. http://www.jstor.org/stable/1578381

Gabor, D. (1947). Acoustical quanta and the theory of hearing. *Nature*, *159*, 591–594. https://doi.org/10.1038/159591a0

Gill, S. S., Kumar, A., Singh, H., Singh, M., Kaur, K., Usman, M., & Buyya, R. (2021). Quantum Computing: A Taxonomy, Systematic Review and Future Directions. Tech. rep., arXiv (Emerging Technologies). http://arxiv.org/abs/2010.15559

Grigoryan, A. M., & Agaian, S. S. (2020). New look on quantum representation of images: Fourier transform representation. *Quantum Information Processing*, *19*(5), 148. https://doi.org/10.1007/s11128-020-02643-3

Hamido, O. C. (2022). QAC: Quantum-computing aided composition. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 159–195). Springer. https://doi.org/10.1007/978-3-031-13909-3_8

Hamido, O. C., & Itaboraí, P. V. (2023). OSC-Qasm: Interfacing music software with quantum computing. C. Johnson, N. Rodríguez-Fernández, & S. M. Rebelo (Eds.) *Artificial Intelligence in Music, Sound, Art and Design*, (pp. 372–382). Springer. https://doi.org/10.1007/978-3-031-29956-8_24

Harris, S., & Sinclair, S. (2019). *liblo: Lighweight OSC implementation*. [Source Code. Version 0.10.0]. https://pypi.org/project/python-osc/

ISQCMC (2021). *1st International Symposium on Quantum Computing and Musical Creativity*. [Symposium]. https://iccmr-quantum.github.io/1st_isqcmc/

Itaboraí, P. V. (2023). *Quantumaudio Package*. [Source Code]. https://github.com/iccmr-quantum/quantumaudio

Itaboraí, P. V. (2023b). *"Rasgar, Saber" by Paulo Itaborai - work-in-progress version , Eletronics Only (without distortion)*. [Audio Recording]. YouTube. https://www.youtube.com/watch?v=xidngkJGkmA

Itaboraí, P. V., & Jones, D. J. (2023). *Python client for SuperCollider: Refactoring 'server.py' and migrating to 'python-osc'*. [Source Code Pull Request. Version v0.0.6]. https://github.com/ideoforms/python-supercollider/pull/12

Itaboraí, P. V., & Miranda, E. R. (2022). Quantum representations of sound: From mechanical waves to quantum circuits. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 223–274). Springer. https://doi.org/10.1007/978-3-031-13909-3_10

Itaboraí, P. V., & Vicente, D. (2023). *ReVeR - by Paulo Itaborai and Dino Vicente - Technical Rehearsal*. [Video]. YouTube. https://youtu.be/sJM55HmntPU

Itaboraí, P. V., Schwägerl, T., Yáñez, M. A., Crippa, A., Jansen, K., Miranda, E. R., & Thomas, P. (2023). Variational quantum harmonizer: Generating chord progressions and other sonification methods with the vqe algorithm. *2nd International Symposium on Quantum Computing and Musical Creativity (ISQCMC Berlin)*. Berlin. https://doi.org/10.5281/zenodo.10206731

Jones, D. J. (2020). *Python client for SuperCollider*. [Source Code. Version v0.0.5]. https://github.com/ideoforms/python-supercollider

Jozsa, R. (2010). Classical simulation and complexity of quantum computations. F. Ablayev, & E. W. Mayr (Eds.) *Computer Science – Theory and Applications*, (pp. 252–258). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-13182-0_23

Khan, R. A. (2019). An improved flexible representation of quantum images. *Quantum Information Processing*, *18*(7), 201. https://doi.org/10.1007/s11128-019-2306-6

Killoran, N., Izaac, J., Quesada, N., Bergholm, V., Amy, M., & Weedbrook, C. (2019). Strawberry Fields: A Software Platform for Photonic Quantum Computing. *Quantum*, *3*, 129. https://doi.org/10.22331/q-2019-03-11-129

Kirkbride, R. (2016). Foxdot: Live coding with python and supercollider. *Proceedings of the International Conference on Live Interfaces*, (pp. 194–198).

Klappenecker, A., & Rotteler, M. (2001). Discrete cosine transforms on quantum computers. *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.*, (pp. 464–468). IEEE.

Le, P. Q., Dong, F., & Hirota, K. (2011). A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Information Processing*, *10*, 63–84.

Li, H.-S., Qingxin, Z., Lan, S., Shen, C.-Y., Zhou, R., & Mo, J. (2013). Image storage, retrieval, compression and segmentation in a quantum system. *Quantum Information Processing*, *12*(6), 2269–2290. https://doi.org/10.1007/s11128-012-0521-5

Li, P. (2022). Quantum implementation of the classical canny edge detector. *Multimedia Tools and Applications*, *81*(8), 11665–11694. https://doi.org/10.1007/s11042-022-12337-w

Li, P., Guo, J., Wang, B., & Hao, M. (2019). Quantum circuits for calculating the squared sum of the inner product of quantum states and its application. *International Journal of Quantum Information*, *17*(05), 1950043. https://doi.org/10.1142/S0219749919500436

Li, P., Wang, B., Xiao, H., & Liu, X. (2018). Quantum representation and basic operations of digital signals. *International Journal of Theoretical Physics*, *57*(10), 3242–3270. https://doi.org/10.1007/s10773-018-3841-0

Lisnichenko, M., & Protasov, S. (2023a). *QPI*. [Source Code]. https://github.com/UralmashFox/QPI/

Lisnichenko, M., & Protasov, S. (2023b). Quantum image representaion: a review. *Quantum Machine Intelligence*, *5*, 2. https://doi.org/10.1007/s42484-022-00089-7

Liu, W.-J., Gao, P.-P., Yu, W.-B., Qu, Z.-G., & Yang, C.-N. (2018). Quantum relief algorithm. *Quantum Information Processing*, *17*(10), 280. https://doi.org/10.1007/s11128-018-2048-x

Mallat, S. (1999). *A wavelet tour of signal processing*. Elsevier. https://doi.org/10.1016/B978-0-12-374370-1.X0001-8

Mannone, M., & Rocchesso, D. (2022a). Quanta in sound, the sound of quanta: A voice-informed quantum theoretical perspective on sound. In Miranda, E. R. (Ed.), *Quantum Computing in the Arts and Humanities: An Introduction to Core Concepts, Theory and Applications*, (pp. 193–226). Springer. https://doi.org/10.1007/978-3-030-95538-0_6

Mannone, M., & Rocchesso, D. (2022b). Sing and measure: Sound as voice as quanta. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 297–312). Springer. https://doi.org/10.1007/978-3-031-13909-3_12

Maor, E. (2020). *Music by the Numbers: From Pythagoras to Schoenberg*. Princeton University Press.

McCartney, J. (2002). Rethinking the computer music language: Super collider. *Computer Music Journal*, *26*(4), 61–68. http://www.jstor.org/stable/3681770

McFee, B., McVicar, M., Faronbi, D., Roman, I., Gover, M., Balke, S., Seyfarth, S., Malek, A., Raffel, C., Lostanlen, V., van Niekirk, B., Lee, D., Cwitkowitz, F., Zalkow, F., Nieto, O., Ellis, D., Mason, J., Lee, K., Steers, B., Halvachs, E., Thomé, C., Robert-Stöter, F., Bittner, R., Wei, Z., Weiss, A., Battenberg, E., Choi, K., Yamamoto, R., Carr, C., Metsai, A., Sullivan, S., Friesch, P., Krishnakumar, A., Hidaka, S., Kowalik, S., Keller, F., Mazur, D., Chabot-Leclerc, A., Hawthorne, C., Ramaprasad, C., Keum, M., Gomez, J., Monroe, W., Morozov, V. A., Eliasi, K., nullmightybofo, Biberstein, P., Sergin, N. D., Hennequin, R., Naktinis, R., beantowel, Kim, T., Åsen, J. P., Lim, J., Malins, A., Hereñú, D., van der Struijk, S., Nickel, L., Wu, J., Wang, Z., Gates, T., Vollrath, M., Sarroff, A., Xiao-Ming, Porter, A., Kranzler, S., Voodoohop, Gangi, M. D., Jinoz, H., Guerrero, C., Mazhar, A., toddrme2178, Baratz, Z., Kostin, A., Zhuang, X., Lo, C. T., Campr, P., Semeniuc, E., Biswal, M., Moura, S., Brossier, P., Lee, H., & Pimenta, W. (2023). *librosa: 0.10.1*. [Software]. https://doi.org/10.5281/zenodo.8252662

Menezes, F. (2013). *Matemática dos Afetos: Tratado de (Re)composição Musical [Mathematics of Affections: A Treatise on Musical (Re)Composition]*. Edusp.

Meyer-Eppler, W. (1958). Statistic and psychologic problems of sound. *Die Reihe*, *1*, 55–61.

Miranda, E. R. (Ed.) (2022a). *Quantum Computing in the Arts and Humanities: An Introduction to Core Concepts, Theory and Applications*. Springer. https://doi.org/10.1007/978-3-030-95538-0

Miranda, E. R. (Ed.) (2022b). *Quantum Computer Music: Foundations, Methods and Advanced Concepts*. Cham: Springer. https://doi.org/10.1007/978-3-031-13909-3

Miranda, E. R., & Basak, S. (2022). Quantum computer music: Foundations and initial experiments. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 43–67). Springer. https://doi.org/10.1007/978-3-031-13909-3_3

Miranda, E. R., & Miller-Bakewell, H. (2022). Cellular automata music composition: From classical to quantum. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 105–130). Springer. https://doi.org/10.1007/978-3-031-13909-3_6

Miranda, E. R., & Shaji, H. (2023). Generative music with partitioned quantum cellular automata. *Applied Sciences*, *13*(4). https://www.mdpi.com/2076-3417/13/4/2401

Miranda, E. R., & Siegelwax, B. N. (2022). Teaching qubits to sing: Mission impossible? *International Journal of Unconventional Computing*, *17*, 303–331. http://hdl.handle.net/10026.1/19633

Miranda, E. R., Thomas, P., & Itaboraí, P. V. (2023). Q1synth: A quantum computer musical instrument. *Applied Sciences*, *13*(4). https://www.mdpi.com/2076-3417/13/4/2386

Miranda, E. R., & Wanderley, M. M. (2006). *New digital musical instruments: control and interaction beyond the keyboard*, vol. 21. AR Editions, Inc.

Miranda, E. R., Yeung, R., Pearson, A., Meichanetzidis, K., & Coecke, B. (2022). A quantum natural language processing approach to musical intelligence. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 313–356). Springer. https://doi.org/10.1007/978-3-031-13909-3_13

Mistry, R., & Ortega, J. (2022). Experiments in quantum frequency detection using quantum fourier transform. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 275–295). Springer. https://doi.org/10.1007/978-3-031-13909-3_11

Nejad, M. Y., Heikalabad, M. M., & Rasouli, S. (2020). A blind quantum audio watermarking based on quantum discrete cosine transform. *Journal of Information Security and Applications*, *55*, 102495. https://www.sciencedirect.com/science/article/pii/S2214212619311093

Nielsen, M. A., & Chuang, I. (2002). *Quantum Computation and quantum information*. American Association of Physics Teachers.

Oshiro, S. (2022). Quiko: A quantum beat generation application. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 131–158). Springer. https://doi.org/10.1007/978-3-031-13909-3_7

Pang, C.-Y., Zhou, R.-G., Hu, B.-Q., Hu, W., & El-Rafei, A. (2019). Signal and image compression using quantum discrete cosine transform. *Information Sciences*, *473*, 121–141. https://www.sciencedirect.com/science/article/pii/S0020025516310970

Plesch, M., & Brukner, i. c. v. (2011). Quantum-state preparation with universal gate decompositions. *Phys. Rev. A*, *83*, 032302. https://link.aps.org/doi/10.1103/PhysRevA.83.032302

Python-OSC (2013). *Python-osc library*. [Source Code. Version]. https://pypi.org/project/python-osc/

Qi, J., & Tejedor, J. (2021). Classical-to-quantum transfer learning for spoken command recognition based on quantum neural networks. https://doi.org/10.48550/arXiv.2110.08689

Qiskit Contributors, 2023 (2023). *Qiskit: An Open-source Framework for Quantum Computing*.

Roads, C. (2002). *Microsound*. The MIT Press. https://doi.org/10.7551/mitpress/4601.001.0001

Sacchetto, M., Servetti, A., & Chafe, C. (2021). Jacktrip-webrtc: Networked music experiments with pcm stereo audio in a web browser. *Web Audio Conference*.

Sacré, D. (2015). *pyliblo: Python bindings for the liblo OSC library*. [Source Code. Version 0.10.0]. https://pypi.org/project/python-osc/

Şahin, E., & Yilmaz, İ. (2019). QRMA: quantum representation of multichannel audio. *Quantum Information Processing*, *18*(7), 209. https://doi.org/10.1007/s11128-019-2317-3

Samaruga, L., & Riera, P. (2022). A port of the supercollider's class library to python. *Proceedings of the 17th International Audio Mostly Conference*, AM '22, (p. 137–142). Association for Computing Machinery. https://doi.org/10.1145/3561212.3561250

Scarassatti, M. A. F. (2008). *Walter Smetak: O alquimista dos sons [Walter Smetak: The Alchemist of Sound]*. Perspectiva, Edições SESC SP.

Scheirer, E. D., & Ray, L. (1998). Algorithmic and wavetable synthesis in the mpeg-4 multimedia standard. *Audio Engineering Society Convention 105*. Audio Engineering Society.

Smetak, W. (1967/2019). Histórico sobre as plásticas sonoras [Historic on the sonic plastics]. In Migracielo, . (Ed.), *O Enxerto Do Takaká & Outros Textos [The Takaká Graft and Other Texts]*, (pp. 19–30). Outr&m Editorial.

Smetak, W. (1974/2019). Ensaio para o artesanato da improvisação [On the craft of improvisation]. In Migracielo, . (Ed.), *O Enxerto Do Takaká & Outros Textos [The Takaká Graft and Other Texts]*, (pp. 229–242). Outr&m Editorial.

Sounding_Qubits (2023). *Sounding Qubits: Quantum computing and musical creativity*. [Video Documentary]. YouTube. https://www.youtube.com/watch?v=wLORlYIl-QM

Srivastava, G. P. (2019). *The physics of phonons*. Routledge.

Stockhausen, K. (1957). *Klaviestücke XI*. Universal Edition. [Score]. https://www.universaledition.com/piano-piece-xi-for-piano-stockhausen-karlheinz-ue12654

Stockhausen, K. (1966). *Kontakte, Nr. 12: for electronic, piano percussion*. Universal Edition. [Score].

SuperCollider (2013). *SuperCollider 3.12.2 Documentation: Wavetable*. [Source Code Documentation, Advanced Notes section.]. https://doc.sccode.org/Classes/Wavetable.html

Tavares, G. M. (2021). *Atlas do Corpo e da Imaginação - teoria, fragmentos e imagens [Atlas of the Body and of Imagination - theory, fragments and images]*. Dublinense.

Topel, S., Serniak, K., Burkhart, L., & Carle, F. (2022). Superconducting qubits as musical synthesizers for live performance. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 447–464). Springer. https://doi.org/10.1007/978-3-031-13909-3_18

Venegas-Andraca, S. E. (2005). *Discrete quantum walks and quantum image processing*.

Venegas-Andraca, S. E., & Bose, S. (2003). Storing, processing, and retrieving an image using quantum mechanics. *Quantum Information and Computation*, *5105*, 137–147.

Wang, J. (2016). QRDA: quantum representation of digital audio. *International Journal of Theoretical Physics*, *55*(3), 1622–1641. https://doi.org/10.1007/s10773-015-2800-2

Wang, Z., Xu, M., & Zhang, Y. (2022). Review of quantum image processing. *Archives of Computational Methods in Engineering*, *29*(2), 737–761. https://doi.org/10.1007/s11831-021-09599-2

Weaver, J. L. (2022). Quantum music playground tutorial. In Miranda, E. R. (Ed.), *Quantum Computer Music: Foundations, Methods and Advanced Concepts*, (pp. 197–222). Springer. https://doi.org/10.1007/978-3-031-13909-3_9

Wright, M., & Freed, A. (1997). Open SoundControl: A new protocol for communicationg with sound synthesizers. *International Computer Music Conference (ICMC)*, vol. 1997, (pp. 101–104). http://hdl.handle.net/2027/spo.bbp2372.1997.033

Xenakis, I. (1992). *Formalized Music: Thought and Mathematics in Composition*. Harmonologia series. Pendragon Press.

Xu, G., Xu, X., Wang, X., & Wang, X. (2019). Order-encoded quantum image model and parallel histogram specification. *Quantum Information Processing*, *18*(11), 346. https://doi.org/10.1007/s11128-019-2463-7

Yan, F., Gao, S., Iliyasu, A. M., & Chen, K. (2022). Probability amplitude-encoded multichannel representation for quantum audio signals. *Quantum Information Processing*, *21*(3), 95. https://doi.org/10.1007/s11128-022-03435-7

Yan, F., Iliyasu, A. M., Guo, Y., & Yang, H. (2018). Flexible representation and manipulation of audio signals on quantum computers. *Theoretical Computer Science*, *752*, 71–85. https://doi.org/10.1016/j.tcs.2017.12.025

Yan, F., Iliyasu, A. M., Le, P. Q., Sun, B., Dong, F., & Hirota, K. (2013). A parallel comparison of multiple pairs of images on quantum computers. *International Journal of Innovative Computing and Applications*, *5*(4), 199–212. https://www.inderscienceonline.com/doi/abs/10.1504/IJICA.2013.062955

Yan, F., Iliyasu, A. M., & Venegas-Andraca, S. E. (2016). A survey of quantum image representations. *Quantum Information Processing*, *15*(1), 1–35. https://doi.org/10.1007/s11128-015-1195-6

Yan, F., Venegas-Andraca, S. E., & Hirota, K. (2023). Toward implementing efficient image processing algorithms on quantum computers. *Soft Computing*, *27*(18), 13115–13127. https://doi.org/10.1007/s00500-021-06669-2

Yilmaz, İ., & Şahin, E. (2018). Security of neqr quantum image by using quantum fourier transform with blind trent. *International Journal of Information Security Science*, *7*(1), 20–25.

Zhang, X.-M., Li, T., & Yuan, X. (2022). Quantum state preparation with optimal circuit depth: Implementations and applications. *Physical Review Letters*, *129*(23). https://doi.org/10.1103%2Fphysrevlett.129.230504

Zhang, Y., Lu, K., Gao, Y., & Wang, M. (2013). NEQR: a novel enhanced quantum representation of digital images. *Quantum Information Processing*, *12*(8), 2833–2860. https://doi.org/10.1007/s11128-013-0567-z

Ziemer, R. E., & Tranter., W. H. (2014). *Principles of Communications*. John Wiley & Sons.