Reports on Progress in Physics

PURPOSE-LED PUBLISHING™

REVIEW

The advent of quantum computer music: mapping the field

To cite this article: Eduardo Reck Miranda 2024 *Rep. Prog. Phys.* **87** 086001

IOP Publishing | Reports on Progress in Physics
Rep. Prog. Phys. **87** (2024) 086001 (29pp) | https://doi.org/10.1088/1361-6633/ad627a

Review

The advent of quantum computer music: mapping the field

Eduardo Reck Miranda

Interdisciplinary Centre for Computer Music Research (ICCMR), Faculty of Arts, Design and Architecture, University of Plymouth, Plymouth PL4 8AA, United Kingdom

E-mail: eduardo.miranda@plymouth.ac.uk

Received 14 March 2024, revised 25 June 2024
Accepted for publication 12 July 2024
Published 24 July 2024

Corresponding editor: Dr Maciej Lewenstein

Abstract
Quantum computing technology is developing at a fast pace. The impact of quantum computing on the music industry is inevitable. This paper maps the emerging field of quantum computer music. Quantum computer music investigates, and develops applications and methods to process music using quantum computing technology. The paper begins by contextualising the field. Then, it discusses significant examples of various approaches developed to date to leverage quantum computing to learn, process and generate music. The methods discussed range from rendering music using data from physical quantum mechanical systems and quantum mechanical simulations to computational quantum algorithms to generate music, including quantum AI. The ambition to develop techniques to encode audio quantumly for making sound synthesisers and audio signal processing systems is also discussed.

Keywords: quantum computer music, quantum computing, music and physics

1. Introduction

Before anything else, let me clarify that this paper does not report progress in physics per se as the title of this journal might have suggested. Rather, it is about progress in *applications* of physics. More precisely, it discusses applications of quantum computing to create music.

In-depth expertise in quantum mechanics is not essential to follow this paper. However, the ability to read mathematical equations describing physical phenomena and an understanding of basic quantum computing concepts will help appreciate the systems discussed below, especially the gate-based approach to quantum computing [7].

The relationship between music and physics has been explored from time immemorial. And the practice of using computers in music is as old as computing technology itself. Therefore, the fact that musicians have started to explore quantum computers to create music should not come as a surprise. This paper introduces the emerging field of quantum computer music and discusses its various ramifications and representative examples.

But make no mistake: quantum computer music is not a genre of music. It is a field of research. By the same token note that I refrained from using the label ‘quantum music’ in this paper.

Strictly speaking, there is no such thing as quantum music. Music lives in an acoustic world perceived by the human hearing system. Most healthy adults have an average hearing range between 20 Hz and 20 kHz. People cannot hear vibrations in the subatomic realm.

However, as we shall see below it is possible to build all sorts of transducers to convert signals from one physical realm into another. Thus, to eavesdrop on the quantum mechanical world is feasible. The small print is that such eavesdropping often requires translations and various layers of subjective interpretations. But from an artistic standpoint, this is not a bad thing.

Dating from the 9th century, the Carnival of Venice still catches our imagination today. It makes me think of masked revellers in lavishing costumes partying all over town.

The premiere of the opera *Il mondo della luna* (*The World on the Moon*, in English), by composer Baldassare Galuppi, took place during the carnival of 1750. This opera could not be more appropriate for the festivity at the time. It tells the story of Ecclitico, who is in love with Clarice. But Clarice's father, Buonafede, is not convinced about Ecclitico's intentions towards his daughter.

To impress Buonafide, Ecclitico comes up with a cunning plan. He pretends to be an astronomer. As the story develops, Ecclitico shows Buonafede his 'powerful' telescope. Ecclitico hoaxes that the telescope can spy on inhabitants of the moon and invites Buonafede to have a look. While Buonafede looks through the telescope, Ecclitico move caricatures of aliens in front of its lens. Buonafede falls for the trick and approves the romance.

The ancient Greek philosophical maxim that astronomy is for the eyes what music is for the ears has always inspired musicians. *The Planets*, by Gustav Holst, written between 1914 and 1917, also comes to mind here. This is probably the most celebrated symphony inspired by astronomy ever composed. And at the pop music front, of course, there is Björk's song *Cosmogony*, from 2011, inspired by the Big Bang theory.

In truth, there is a plethora of music inspired by science, above all physics. More recently, quantum physics emerged as a natural progression in this vein. Several composers and physicists are promoting this. To cite but one example, in 2023, Reiko Yamada (a composer) and Maciej Lewenstein (a quantum physicist) hosted the Quantum Sounds Symposium at ICFO, in Spain, intending to '... gain knowledge and inspiration from some of the most recent research done in the interface between quantum physics and music composition, data sonification, interactive sound interface, sound programming and audio software development' [50]. At the occasion, they announced the release of a vinyl album with musical compositions using quantum randomness [124].

The advent of computing technology that enables the processing of large volumes of data, combined with the development of modelling and simulation methods, enabled composers to develop approaches to musical composition that are more objectively informed by science rather than merely inspired by it. This trend is yielding some unprecedented activity at the intersection between quantum physics and music.

For instance, in 2011, Alexis Kirke, a former PhD student of mine at ICCMR in the University of Plymouth, composed a piece entitled *Cloud Chamber*. This is a piece for violin that goes well beyond simply taking inspiration from quantum physics. Here the violinist interacts in real-time with cosmic ray particles in a bespoke cloud chamber on stage [60]. A cloud chamber, also referred to as a Wilson cloud chamber, is a particle detector used for visualizing the passage of ionizing radiation.

A camera above the cloud chamber recorded the particle tracks, which are synthesized into sounds. The ICCMR team developed a system, which enabled the violinist to create an electrical field that directly affected the charged particles in the chamber. This made possible real-time musical interaction between the violinist and the particles. A video recording of *Cloud Chamber* is freely available online [61].

In 2015, an MIT Media Lab group led by Joseph Paradiso, developed a platform for the sonification of particle collisions [45, 86]. They developed a method to listen to particle collision data from CERN's ATLAS detector. A prototype of this fascinating work was demonstrated at a workshop held in July 2015 at the Montreux Jazz Festival, in Switzerland.

ATLAS is one of the Large Hadron Collider (LHC) detectors used to search for new physics such as the Higgs boson or dark matter. Beams of extremely high-energy particles collide and form new particles that are scattered in all directions. The ATLAS detector collects collision data such as the trajectory and energy of the charged collision debris. The MIT sonification system takes a small subset of the collision data and relays them through algorithms designed to render the data as sequences of musical notes.

I found the ATLAS sonification work fascinating. It motivated me to further elaborate on it. In 2017, I took a short residency at MIT Media Lab to advance new methods to render ATLAS data into music. To put the outcomes into practice, I composed an opera, entitled *Lampedusa*.

The opera is set in a parallel Shakespearean Universe. The libretto is in an otherworldly language invented by David J. Peterson, the linguist who makes artificial languages for Hollywood films and TV series, such as *The Game of Thrones*' Dothraki. The plot takes place before the arrival of Prospero and Miranda in Lampedusa, allegedly the island portrayed in Shakespeare's play *The Tempest*. The opera tells the story of Sycorax, a refugee from Europe, her son, Caliban, and Ariel. Ariel is an invisible native inhabitant who objects to Caliban's ambitions of reigning over the island [65, 66, 72]. Video recordings of the three main acts of *Lampedusa* are freely available online [83–85].

The stories of *Il mondo della luna* and *Lampedusa* both allude to physics. Whereas the former refers to astronomy, the latter refers to the many-worlds interpretation of quantum mechanics. But from a creative process perspective, they are fundamentally different. To compose *Lampedusa*, I relinquished a significant proportion of my human-centred creative agency in favour of Nature. That is, the piece includes materials created by subatomic phenomena, not by me. Henceforth, this approach, which was also adopted in the aforementioned *Cloud Chamber* is referred to as *musification* of natural phenomena or models thereof [39].

While *Lampedusa* was being premiered by the BBC Singers, at the Multiverse Festival in Plymouth, UK, in 2018, I found out that Rigetti, then a start-up firm based in Berkeley, California, announced they were making their experimental quantum computers available for researchers.

Quantum computing was on my radar for a while. To begin with, it was all theoretical, weird, and difficult to grasp. There were no actual quantum computers available until relatively recently. In no time I packed my suitcase and went to Berkeley to see it all for myself. As it turned out, I became a beta-tester for pyQuil, a Python library for quantum programming that they were developing.

Thanks to Rigetti, I was able to compose, *Zeno*, my first fully-fledged composition using quantum computers. *Zeno* is a rather exploratory piece for bass clarinet and electronic sounds. During the performance, a local laptop listened to the clarinet and extracted statistical information about the incoming notes. A piece of software used this information to build quantum algorithms, which were transmitted to the quantum computer in California through the Internet. Then, the machine ran the algorithms and sent the results back to my laptop. Those results were subsequently rendered into sounds [82].

Zeno is significant for me. What I achieved with a rudimentary quantum computer sporting only a few qubits would have required sophisticated AI programming on a state-of-the-art desktop digital machine. For instance, encoding transition probabilities between musical events as quantum wavefunction amplitudes is far more straightforward to program than implementing a neural network; see the section about QuPoly below. The thought of what these new machines might afford musicians twenty of even ten years from here is mind-bogging. A video recording of *Zeno* is freely available on YouTube [127].

2. Quantum computer music in context

To better understand why it makes sense to explore emerging quantum computing technology in music, let me first contemplate some historical facts and milestones. Then, I present a selection of representative works and delve into technical details.

As early as the 1840 s, Ada King, countess of Lovelace, in England, predicted that computers would be able to compose music. Daughter of famed poet Lord Byron, she studied mathematics and logic and is likely the first-ever computer programmer. On a note about Charles Babbage's Analytical Engine, she wrote:

'*Supposing, for instance, that the fundamental relations of pitched sounds in the science of harmony and musical composition were susceptible to such expression and adaptations, the Engine might compose elaborate and scientific pieces of music of any degree of complexity or extent.*' ([73], p 21)

At about the same time, at the apex of the first Industrial Revolution, steam-powered machines controlled by stacks of punched cards were being engineered for the textile industry. Musical instrument builders promptly recognised that punch-card stacks could be used to drive automatic pipe organs. Such initiatives revealed a glimpse of an unsettling idea about the nature of the music they produced: it emanated from information, which could also be used to control all sorts of machines. The idea soon evolved into mechanical pianos and several companies began as early as the 1900 s to manufacture self-playing pianos, generally known as 'pianolas'.

Self-playing pianos enabled musicians to record their work with great fidelity: the recording apparatus could punch thousands of holes per minute on a piano roll, enough to store all the notes that a fast virtuoso could play. Because a piano roll stored a set of parameters that represented musical notes rather than sound recordings, the performances remained malleable: the information could be manually edited, the holes re-cut, and so on. This was the beginning of information technology, which gained much sophistication during the twentieth century. In many ways, the usage of stacks of punched cards to control machines paved the way for the development of programmable electronic computers.

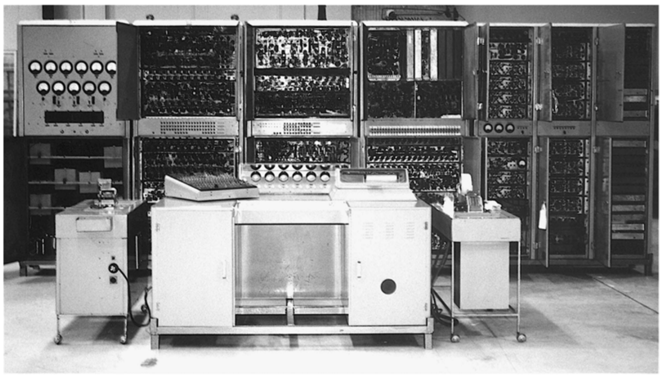

Figure 1. CSIRAC computer used to play back tunes in the early 1950 s. The loudspeaker can be seen in the right-hand door of the console. (Image published with the kind permission of Prof Paul Doornbusch.) Reproduced with permission from Prof Paul Doornbusch.

Most people hardly ever realise that composers started experimenting with computers far before the emergence of most industrial, commercial, and social demands for computers in existence today.

In fact, over the last 70 years or so, progress in computing technology and musical innovation has gone hand in hand. Computers have played a pivotal part in the development of the burgeoning music industry of today. Importantly, composers interested in exploring the potential of computing technology for their métier played an important role in these developments.

For instance, in the early 1950 s, Australia's Council for Scientific and Industrial Research (CSIR) installed a loudspeaker on their Mk1 computer to track the progress of a program using sound (figure 1). Subsequently, Geoff Hill, a mathematician with a musical background, programmed this machine to playback folk tunes [32, 119].

But one of the first significant milestones of computer music took place in 1957 at the University of Illinois at Urbana-Champaign, USA, with the composition *Illiac Suite* by Lejaren Hiller and Leonard Isaacson. Hiller, then a professor of chemistry, collaborated with mathematician Isaacson to program the ILLIAC machine to compose a string quartet [46, 51].

The ILLIAC, short for Illinois Automatic Computer, was one of the first mainframe computers built in the USA, comprising thousands of vacuum tubes. They programmed this machine with rules of harmony and counterpoint. The output was transcribed manually into a musical score for a string quartet.

Whereas Mk1 merely played back an encoded tune—like a pianola—ILLIAC was programmed with algorithms to compose music. Hence *Illiac Suite* is often cited as a pioneering piece of computer-generated music.

Various important inventions and developments have taken place since, notably the invention of the transistor and subsequently the development of the microchip. The microchip enabled the manufacturing of computers that became progressively more accessible to a wider sector of the population.

An increasing number of musicians embraced computing technology since then. Some even built bespoke computer-based devices to create sounds and music. One remarkable example is the UPIC system, built in the 1970s by Iannis Xenakis, in collaboration with engineers at CEMAMu, or *Centre d'Etudes de Mathématique et Automatique Musicales*, in Paris [69].

Xenakis is known for having pioneered the use of computers to run statistical models to generate materials for musical compositions [123]. However, his UPIC system allowed for something rather different at the time. One could draw sketches on an architect's drawing board and a computer would translate the drawings into sound. A notable composition of this time is *Mycenae-Alpha*, completed in 1978. Xenakis used the UPIC to generate sounds, which he further processed in his studio to compose the piece.

Another significant milestone took place in the early 1980 s at IRCAM with *Répons* [10]. This is an unprecedented composition by the celebrated conductor Pierre Boulez [74]. IRCAM, or *Institut de Recherche et Coordination Acoustique/Musique*, is a centre for research into music and technology founded in 1977 in Paris by Boulez himself.

Répons, for chamber orchestra and six solo percussionists, was the first significant piece of classical music to use digital computing technology to perform live on stage. The machine 'listened' to the soloists and 'responded' with synthesised sounds on the spot during a performance. To achieve this, Boulez used a novel system, called 4X System, developed by physicist Giuseppe Di Giugno and his team [29].

The UPIC and *Mycenae-Alpha*, and the 4X System and *Répons* epitomise the beginning of an era of increasingly widespread use of digital computers to develop new approaches to composition and perform interactively with musicians. They signify the beginning of our present time, where personal computers, laptops, notebooks, tablets and even smartphones are used in musical composition and performance.

Moreover, let us not forget the blooming theme of Artificial Intelligence (AI). As computers became increasingly available and user-friendly, composers started to leverage AI to make music. In the 1980 s, David Cope emerged as one of the pioneers of music AI in the USA. Cope developed software that analysed a bunch of musical scores by a given classical music composer and generated new pieces imitating the style of the said composer [22]. He is known for having programmed his system to generate 5000 chorales in the style of Bach [21].

We should bear in mind, however, that AI is merely software. And software needs computing hardware to work. Today, AI runs on a type of computer hardware architecture that dates to the 1940 s. The fundamentals of computing have not changed much since the Mk1 and ILLIAC machines [15]. Essentially, the main difference is that computers today are much smaller and have a lot more switches than their predecessors. They work with microprocessors made up of billions of tiny switches that are activated by electric signals. Values 1 and 0 reflect the on and off states of the switches.

Figure 2. Quantum computers at IBM Quantum as of 2023, available on the cloud. The actual quantum processors live inside the hanging white barrels, which are refrigerators capable of maintaining temperatures as low as −459° F. (Reprint Courtesy of IBM Corporation ©). Reproduced with permission from IBM.

Each of these switches embodies a digital bit—or bit. The bit is the smallest unit of information of a digital processor.

There has been burgeoning research, however, looking for new paradigms for computation and hardware [1, 2, 75], including quantum computation [7, 34, 93].

Quantum computing provides an alternative hardware paradigm. But it is unlikely that quantum computers will replace digital ones. Rather, they are likely to complement each other to facilitate the invention of radically different, and monumentally powerful, AI algorithms for music.

Although the industry is making terrific progress, quantum computers are not generally available yet. They are still being developed. It is often said that the stage of development of quantum computers today is comparable to the stage of development of the large computer mainframes of the 1950 s. But various software simulators are available right now. Moreover, a few companies already provide access to quantum hardware via the cloud (figure 2).

At the time of writing, various types of quantum computers are being developed, using technologies such as trapped ions, quantum dots, topological, neutral atoms, and superconducting [24]. The most popular approach uses superconducting technology and deals with information encoded as quantum bits or qubits. The qubit is to a quantum computer what a bit is to a digital one: it is a basic unit of information. Qubits are not switches. They are subject to the laws of quantum mechanics. Quantum computing harnesses quantum mechanical properties, such as superposition, entanglement, and interference to process information. This makes qubits process information fundamentally different from how digital bits do it.

Performing operations on groups of qubits in superposition enables manipulations on many possible configurations of information simultaneously [116]. Putting this into perspective, a digital processor deals with information represented by a series of bits that are switched on and off individually. For instance, with three bits one can represent numbers

0 to 7 as follows: 000, 001, 010, 011, 100, 101, 110, and 111, respectively. A digital processor deals with this data one at a time. With three qubits, however, a quantum processor represents those numbers simultaneously, which allows for processing them in parallel.

With 20 qubits, we would be able to simultaneously handle over one million pieces of information represented as quantum states in superposition. Currently, some quantum processors already work with hundreds of qubits. It is expected that they might soon have thousands or millions of qubits. However, there are several issues to be solved before these devices can leverage large amounts of qubits for general computation. And, likely, the current superconductor-based approaches to building qubit chips might not be effective in the end. The race for high-fidelity qubit technology is moving at a fast pace.

All the same, an important characteristic of quantum computers is that the results of reading out quantum states cannot be predicted with absolute certainty. There are methods to get around this with increasingly sophisticated techniques for error correction and repetition. However, quantum unpredictability is not necessarily a hindrance for musicians or creative applications in general. Composers have often embraced stochasticity; e.g. Iannis Xenakis, mentioned earlier, is a case in point. Indeed, Reiko Yamada, mentioned earlier, composes using quantum randomness [126].

Quantum computing represents a new milestone in the ever-evolving relationship between quantum physics and music. And it is paving the way for new ways of thinking about music. It is a new musical instrument.

Historically, the emergence of new musical instruments has always led to new kinds of music, which were inconceivable until then. For instance, the electronic music style popularized by the German band, Krafwerk, in the 1970 s, would have been impossible without the emergence of electronic sound synthesizers, sequencers, drum machines and vocoders [105].

Up until recently, musicians have been composing not only inspired by physics but also by rendering mathematical models and physical phenomena captured with sensors into sound and music. Now, composers can take a step beyond. Quantum computers enable musicians to leverage Nature's quantum mechanics to process and generate musical information. And there already is a community of pioneers developing technologies for this and making music with them [53].

Before we move on to examine how quantum computation is being used to create music, let us look at a precursor example of making music with a mathematical quantum mechanical model of quasi-probability distribution: the Wigner function.

3. *Studies on Wigner function*: composing with Fock and Schrödinger cat states

ICFO, the Institute of Photonic Sciences based in Castelldefels, Spain, is developing methods to understand quantum systems using sound and music. A team led by Maciej Lewenstein and Reiko Yamada developed a method to render data generated with Wigner functions into music. Yamada subsequently used their method to compose a piece for a string quartet entitled *Studies on Wigner Function* (figure 3).

It was mentioned earlier that composers have been leveraging stochasticity to compose music. However, Yamada's work is distinct because stochasticity in quantum physics differs from stochasticity in classical physics.

In classical physics, probability distributions describe the likelihood of different outcomes for observable quantities. They are non-negative and have the property that the sum of probabilities over all possible outcomes equals one. However, in quantum mechanics, certain variables, such as the position x and momentum p of a particle, cannot be described by classical probability distributions. The stochasticity of quantum systems is described using the so-called quasi-probability distributions, such as the Wigner function [63] and Husimi Q-function [25].

In quantum physics, a particle cannot have a determined position x and momentum p at the same time. According to the uncertainty principle discovered by Werner Heisenberg in 1927, if the position of a particle is localized with standard deviation Δx and the momentum with standard deviation Δp, then one obtains the inequality $\Delta x \Delta p \geq \hbar$, where $\hbar$ is the Planck constant divided by 2π [44]. To localize the particle, ideally with Δx tending to zero, Δp must tend to infinity, and vice versa [12]. Moreover, x and p are mathematically non-commutative. That is, xp is not the same as px.

It is not feasible to define a probability distribution of x and p in a meaningful way classically. Instead, one can define this with the Wigner function: $W(x,p)$. The Wigner function accounts for negative values, which are necessary to capture quantum states [103] and it allows for the representation of intrinsic and correlated randomness of the position and momentum. Yet, it is normalized like a standard probability distribution. When it is integrated over x and p it gives 1, and when it is integrated over $x(p)$, it gives a probability distribution of $p(x)$.

The data for *Studies on Wigner Function* were generated with the Wigner quasi-distribution of two types of quantum states: Fock states and Schrödinger cat states.

Fock states, named after the physicist Vladimir Fock, describe different states—or modes—of a quantum system, particularly in the context of quantum harmonic oscillators. Quantum modes play a crucial role in the field of photonic quantu Fock states are characterized by a defined number of quanta or particles in a given mode. Thus, the state $|n\rangle$ corresponds to having n particles in the mode. Similarly to other quantum states, Fock states can be in superposition. That is, a quantum system can exist in a linear combination of Fock states until a measurement is made. A Fock state is represented as a probability distribution of particles in a quantum system. For instance, $|n\rangle$ represents a Fock state with n particles where the probability of measuring n particles is given by $|\langle n||\Psi\rangle|^2$. The following expression describes the Wigner function of the nth Fock state (equation (1)):

$$W_n(x,p) = \frac{(-1)^n}{\pi} \mathrm{e}^{-\left(x^2+p^2\right)} \mathcal{L}_n\left(2\left(p^2+x^2\right)\right) \tag{1}$$

where $\mathcal{L}_n$ denotes the nth orthogonal Laguerre polynomial.

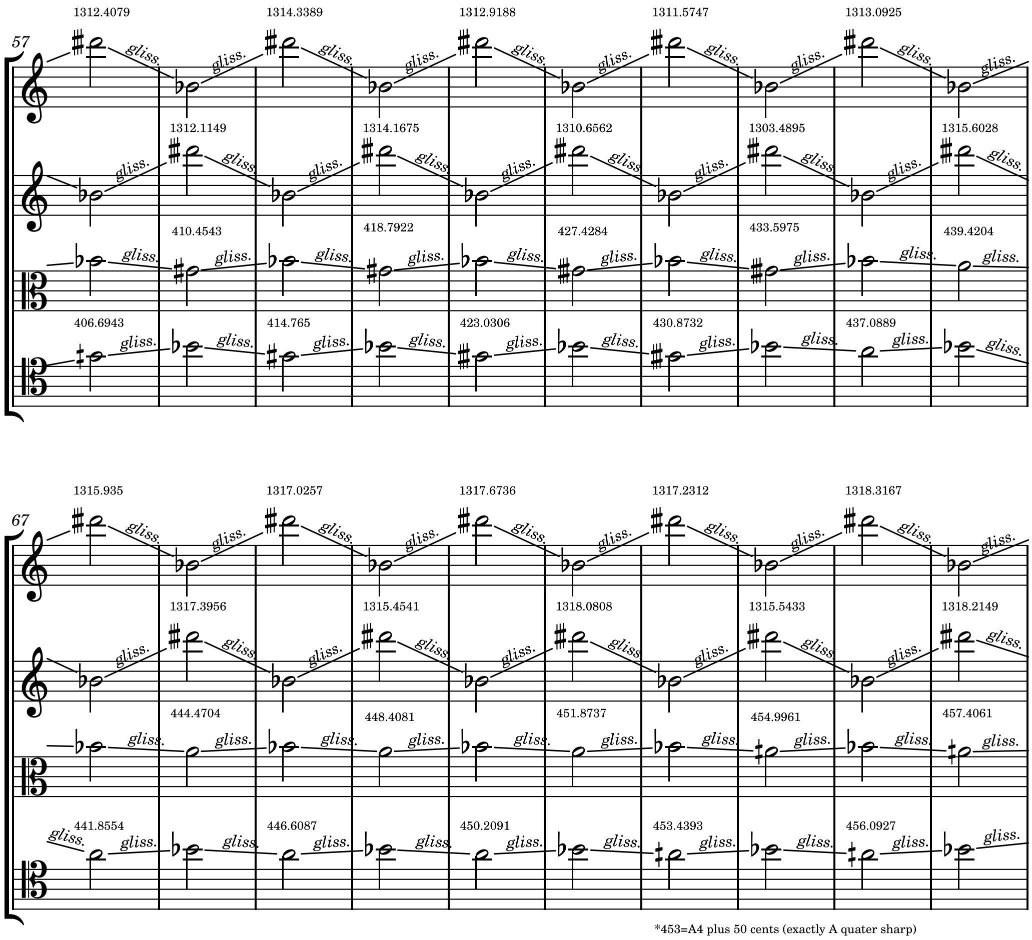

Figure 3. Excerpt of the score for Reiko Yamada's *Studies on Wigner Function*. (Printed with permission from the composer.) Reproduced with permission from Reiko Yamada.

In turn, Schrödinger cat states, named after the physicist Erwin Schrödinger, refer to a quantum superposition of at least two different macroscopic states. The term originated from the famous thought-provoking experiment proposed by Schrödinger, where a cat is placed in a sealed box with a radioactive atom, a flask of poison, a hammer, and a Geiger counter. If the atom decays, the Geiger counter detects radiation, and the hammer is released, breaking the flask. In quantum mechanics, the atom is in a superposition of decayed (fundamental) and undecayed (excited) states until it is observed. Thus, theoretically, the cat is in a superposition of $|cat\ is\ alive\rangle$ and $|cat\ is\ dead\rangle$ at the same time until an observer opens the box and look inside.

For the *Studies on Wigner Function* composition, Yamada worked with optical Schrödinger cat states. These states are superpositions of macroscopically different coherent states of a high number of photons, as predicted and observed by researchers at ICFO in high harmonic generation with above-threshold ionization processes in Xenon atoms [67].

Let us consider the examples of Fock [107] and Schrödinger cat states shown in figures 4 and 5, and the following Wigner function (equation (2)), where α and β are complex numbers associated with the arguments of the function through $\beta-\alpha = x + \mathrm{i}p$:

$$
\begin{aligned}
W(\beta) = {} & \frac{2}{\pi\left(1-\mathrm{e}^{\left(-|\delta\alpha|^2\right)}\right)}\Big[\mathrm{e}^{\left(-2|\beta-\alpha-\delta\alpha|^2\right)} \\
& + \mathrm{e}^{\left(-|\delta\alpha|^2\right)}\mathrm{e}^{\left(-2|\beta-\alpha|^2\right)} - \mathrm{e}^{\left(-|\delta\alpha|^2\right)}\mathrm{e}^{\left(-2|\beta-\alpha|^2\right)} \\
& \times\left(\mathrm{e}^{\left(2(\beta-\alpha)\delta\alpha^*\right)} + \mathrm{e}^{\left(2(\beta-\alpha)^*\delta\alpha\right)}\right)\Big]. \qquad (2)
\end{aligned}
$$

The ICFO team developed several methods to map the behaviour of equation (2) onto music. One of the methods used for *Studies on Wigner Function* associated the minimum and maximum values of a Wigner function to the pitches of musical notes.

The arguments of a Wigner function and their values were discretized as a grid. The grid limits are chosen to contain at least 99% of the quasi-probability distribution. This was done in two different ways: (a) by making the grid regular, with equidistant points, and (b) by making the grid non-regular, using intervals extracted from a Gaussian probability distribution.

Each state produces a stationary pitch. Then, if the state changes, for example, by varying the value of $\delta\alpha$, the pitch changes accordingly. The mappings were done through linear and quadratic functions obtained with regressions. These were

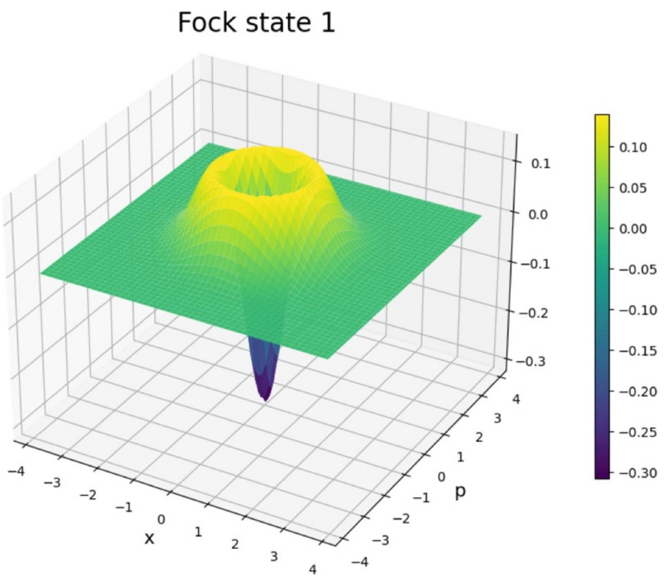

Figure 4. Wigner function of the Fock state $m = 1$. Reproduced with permission from Reiko Yamada.

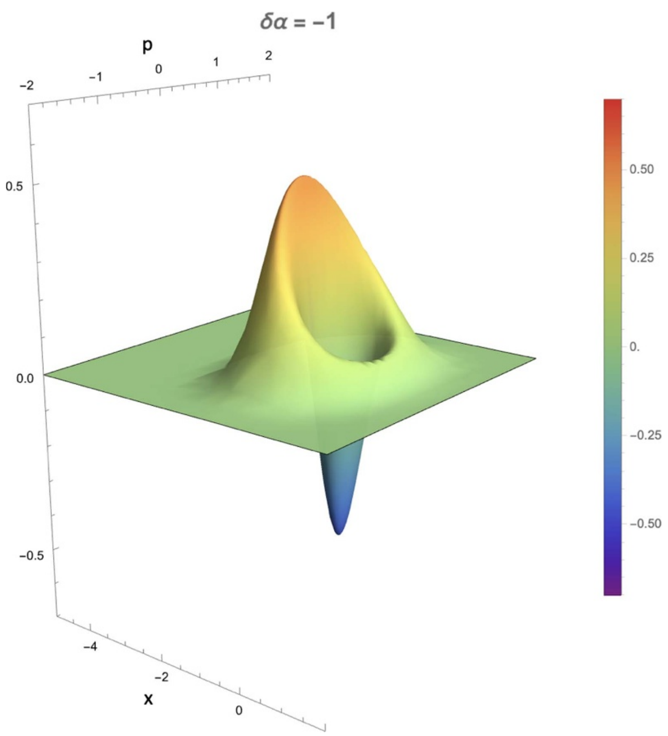

Figure 5. Wigner function of the optical kitten state with $\delta\alpha = -1$. Reproduced with permission from Reiko Yamada.

calculated so that the minimum and maximum values of the Wigner function in the range $\delta\alpha \in (0, -3)$ are mapped to 146.83 Hz and 1318.5 Hz, respectively, and the 0 value to 466.16 Hz.

To transcribe the pitch frequency values onto musical notation (figure 3), they were rounded up to the nearest quarter divisions of the 12 notes of the equal temperament music system [35]. Still, for information, the exact frequencies are indicated above each note on the score. The piece is played by making glissandi from one note to another, which generates a continuous oscillatory-like sound effect. A recording of this composition is freely available for listening on the Internet [125].

4. *Quantum sound*: auditioning transmon qubits

The Yale Quantum Institute is a pioneer in offering year-long residencies in the USA for artists to produce artworks in collaboration with quantum physicists, curated by Florian Carle. In 2019, the artist-in-residence was musician Spencer Topel. Topel collaborated with two doctoral students in quantum physics at Yale, Luke Burkhart and Kyle Serniak, to create *Quantum Sound*.

Quantum Sound is a piece of music with sounds generated from data produced by bespoke quantum hardware that Burkhart and Serniak were using at the time to develop experiments for their respective doctoral theses.

Topel leveraged data produced by their experimental setups to control software-based sound synthesisers of his design in a live performance. Some of the data were recorded beforehand, but some were relayed to control the synthesisers in real-time—directly from the quantum device during experiments—as if the act of conducting the experiments was the act of playing musical instruments during a performance.

In a nutshell, the data came from measurements of transmon qubits to probe their performance. It was mentioned earlier that the qubit is the quantum computer equivalent to the bit in a digital computer: it is the basic unit for representing information. However, this is not entirely correct. Qubits refer to *gate-based quantum computing*. Other types of quantum information processing exist, which may require different kinds of information representation. For instance, *continuous-variable quantum computing*, such as photonic computing, uses *qumodes*. Furthermore, qubits can be physically implemented in several ways. The transmon qubit discussed below is one of them.

Conceptually, a qubit is generally represented as a transparent sphere with opposite poles: $|0\rangle$ and $|1\rangle$. From its centre, a unitary vector whose length is equal to the radius of the sphere can point to anywhere on the surface. In quantum mechanics, this sphere is called *Bloch sphere* and the vector is referred to as a *state vector*. The state vector represents the state of the qubit and can be described in terms of polar coordinates using two angles, θ and φ, as shown in figure 6.

The state of a qubit lives in a two-dimensional complex vector space, referred to as two-dimensional Hilbert space. The canonical basis vectors in Hilbert space are notated as $|0\rangle$ (indicating that it is in a ground state) and $|1\rangle$ (indicating that it is an excited state). This notation, referred to as Dirac notation, provides an abbreviated way to represent a vector. For instance, $|0\rangle$ and $|1\rangle$ represent the vectors shown in equation (3):

$$|0\rangle = \begin{bmatrix}1\\0\end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix}0\\1\end{bmatrix}. \tag{3}$$

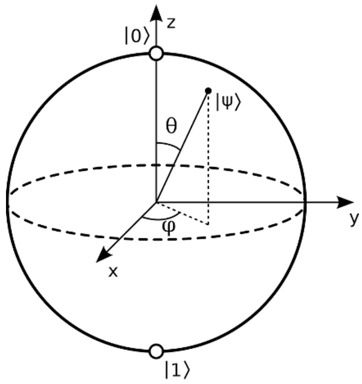

Figure 6. A Bloch sphere representing a single qubit. (Source: Smite-Meister, https://commons.wikimedia.org). This Bloch Sphere image has been obtained by the author(s) from the Wikimedia website where it was made available by Rayhem under a CC BY-SA 3.0 licence. It is included within this article on that basis. It is attributed to Rayhem.

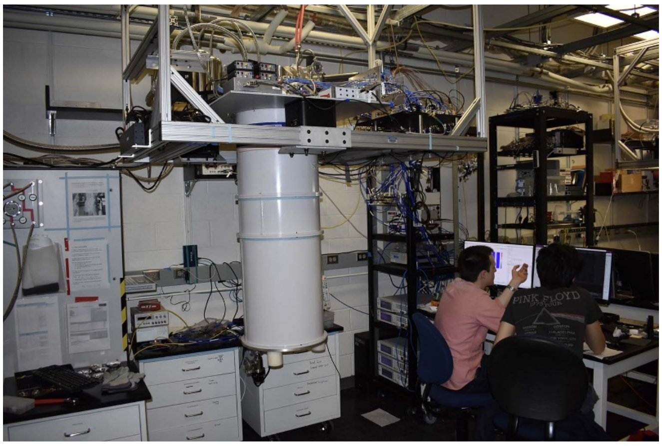

Figure 7. A dilution refrigerator and required paraphernalia to control qubits at Yale Quantum Institute's laboratory. (Image from [114] and published with the kind permission of Yale Quantum Institute.) Reproduced from [114], with permission from Springer Nature.

Mathematically, the state $|\psi\rangle$ of a qubit is written as a linear combination of the basis vectors as follows: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. This linear combination expresses a state of *superposition*.

In a nutshell, qubits process information in a state of superposition. But they will return binary numbers (0 s and 1 s) when we read them. To put it bluntly, the vector will end up pointing to the north or south. The coefficients α and β, referred to as *amplitudes*, represent tendencies towards the north or the south, respectively. In quantum computing terminology, the act of reading qubits is referred to as *projective measurement*, or simply, a *measurement*.

Normally, quantum computers are programmed by applying sequences of operations to qubits. Quantum computing programming tools provide several operators, referred to as *gates*, which act on qubits. For instance, the $\mathbf{RX}(\pi)$ gate rotates the state vector of a qubit by 180 degrees around the *x*-axis of the Bloch sphere geometry. Thus, if the qubit vector is pointing to $|0\rangle$, then this gate flips it to $|1\rangle$, or vice-versa. The $\mathbf{RX}(\pi)$ gate is often referred to as the NOT gate and notated simply as $\mathbf{X}$.

A transmon qubit is implemented with two components: a capacitor and a non-linear inductor. The latter is in the form of a Josephson junction, abbreviated as JJ. A JJ, named after its inventor Brian Josephson [113]. It is made by sandwiching a thin layer of a nonsuperconducting material—establishing a weak link—between two layers of superconducting material, such as aluminium. Electrons in a JJ can tunnel through the weak link between the two superconducting layers. This tunnelling produces subatomic quantum mechanical phenomena on an ordinary atomic scale, which can be harnessed to implement transmon qubits.

Transmon qubits can be constructed using traditional lithography techniques, like those used in semiconductor circuit manufacturing. But to display quantum behaviour, the device must be cooled at temperatures as low as -457° F, which can be achieved inside a dilution refrigerator [9, 128] (figure 7). It is a popular approach in the development of superconducting quantum computers [92]. For instance, IBM Quantum primarily uses transmon qubits. Other approaches include the Cooper-pair Box [27] and Phase Qubits [8].

In an ideal world, the amplitudes α and β discussed above indicate whether the qubit is more likely to return 0 (when the vector is pointing to a region above the equator line) or 1 (when it is pointing below the equator line) when it is measured. However, physical qubits, as we know them today, are very much prone to error. The transmon implementation is no exception. The Holy Grail of quantum computing research is to find ways to mitigate this problem. This is the rationale behind the research conducted by Serniak and Burkhart.

Whereas Serniak was interested in understanding a particular type of qubit error, caused by a phenomenon referred to as non-equilibrium quasiparticle excitations [106], Burkhart conducted experiments to study high-fidelity qubit control and the entanglement between two quantum systems. That is, how they interact—or interfere—with one another. Two entangled qubits can no longer be considered individually. They create a quantum system of their own.

The experiments with the transmon qubit involved creating and measuring quantum states with input DC (Direct Current) microwave signals typically in the range of 1–10 GHz [101]. Qubit states were measured using a paired harmonic oscillator, referred to as *transmon-cavity system*, also made with superconducting material. Here, a qubit state is encoded in the amplitude and phase of the microwave signal that interacts with the harmonic oscillator. By monitoring the oscillator Serniak and Burkhart were able to read the state of the qubit.

Single measurements could be achieved within a few hundred nanoseconds. Figure 8 shows an example of qubit measurements concatenated at a sampling rate of 1 MHz. The up and down jumps in the signal represent the qubit state changes, ground or excited. This is the type of signal that Topel leveraged to control his sound synthesisers. For instance, peaks above a pre-defined threshold could trigger notes to generate a rhythmic pattern or other musical events. The amplitude

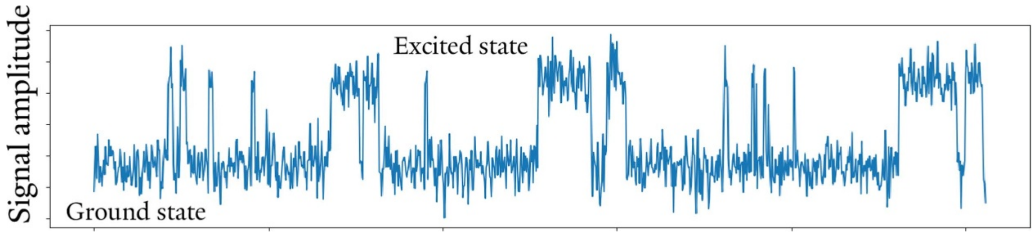

Figure 8. Qubit state dynamics. (Image from [114] and published with the kind permission of Yale Quantum Institute.) Reproduced from [114], with permission from Springer Nature.

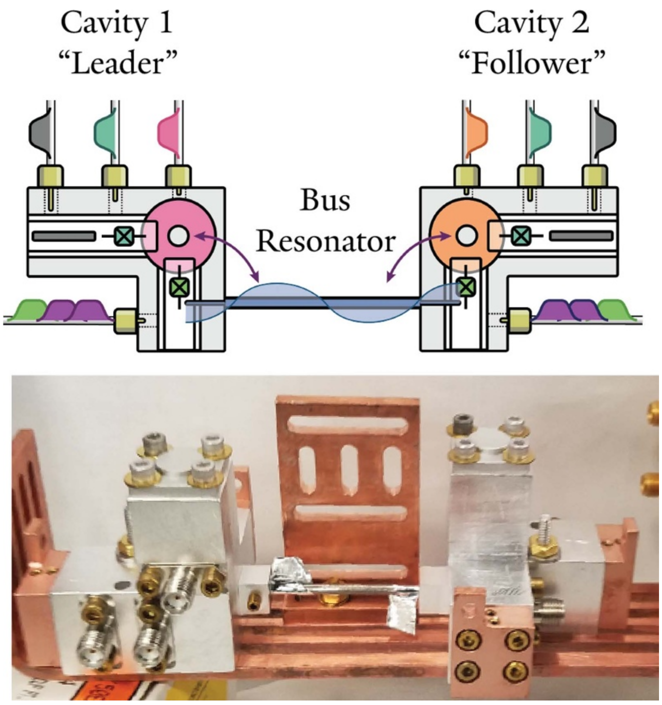

Figure 9. The diagram at the top and the photograph shows the experimental setup developed by Burkhart to study how the qubit in ‘Cavity 1’ interfered with the qubit in ‘Cavity 2’ via high-fidelity control and feedback. (Image from [114] and published with the kind permission of Yale Quantum Institute.) Reproduced from [114], with permission from Springer Nature.

of the signal was also used directly to control sound synthesis parameters, such as frequencies or amplitudes for sound generators.

In the case of the work of Burkhart, the experimental device consisted of two separated transmon qubit units coupled via an auxiliary cavity mode, referred to as *bus resonator*, to enable entanglement [14] (figure 9). Still, each unit could be controlled independently and yield separate readouts. Topel used data for making sounds, which were generated as follows:

- (a) by applying $\mathbf{RX}(\pi)$ gates to a single qubit conditioned on fast feedback of its own state (figure 8)
- (b) by applying $\mathbf{RX}(\pi)$ gates to one qubit conditioned to the fast feedback of the state of the other qubit (figure 10)

In case (b), one qubit acted as a leader and the other as a follower. In entanglement, the state of the follower is controlled by the state of the leader. This can be seen in the plot at the top of figure 10(Good Follower). But errors occurred when the angle of the $\mathbf{RX}(\boldsymbol{\theta})$ gate varied continuously from $\theta = 0$ to $\theta = 2\pi$. Not surprisingly, in this case, the relationship between the follower and the leader was inconsistent, as shown in the plot at the bottom of figure 10 (Bad Follower).

To sonify the signals acquired from the aforementioned experiments Topel down-sampled the qubit state dynamic signals to an audio rate with a DC offset to centre the signal at 0 V, scaled to 2 V pole-to-pole. These processed signals acted as audio oscillators operating in the audible range (kHz) with the qubit-transition information used as a primary control voltage and the quantum noise signal extracted for modulation of the oscillator as a secondary effect. A modulation index parameter allows the user to control the amount of modulation applied, which uses the quantum noise encoded in the signal to generate its waveform. An index equal to zero would result in no modulation effect applied, with an increasing index corresponding to an increasing modulation effect. For more details, see [114].

Quantum Sound was premiered on 14 June 2019, at the 24th International Festival of Arts and Ideas, in New Heaver, CN, USA. The performance was broadcast live on WPKN 89.5 FM radio. The piece was published by Yale Quantum Institute as a limited edition album on vinyl and a video recording of the piece is freely available online [98].

5. *Spinnings*: The Q1Synth instrument

In 2021, I developed Q1Synth with my post-graduate students at ICCMR, namely Peter Thomas and Paulo Itaboraí [77]. At the time of writing, it still is an ongoing project and there are different incarnations of it. Below is an introduction to one of them.

In a similar vein to the sonification of qubits discussed in section 4, Q1Synth is a musical instrument that produces sounds from qubit state vectors and measurements. However, Q1Synth is not tied to specific hardware. Furthermore, it operates at a higher level of abstraction: it departs from an abstract visual representation of a qubit rather than from operations at the hardware level.

Q1Synth works with only one qubit. The system is presented on a computer screen (figure 11) or mobile device (figure 12) as an artistic rendering of a Bloch sphere [34] portraying a qubit (figure 11). The performer plays the instrument

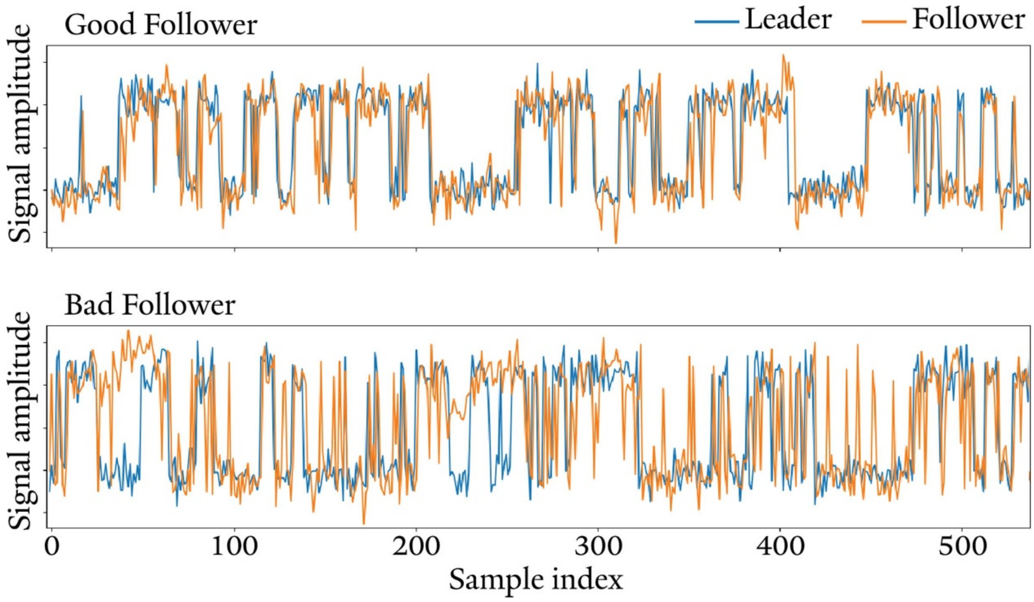

Figure 10. Entangled qubit state dynamics. (Image from [114] and published with the kind permission of Yale Quantum Institute.) Reproduced from [114], with permission from Springer Nature.

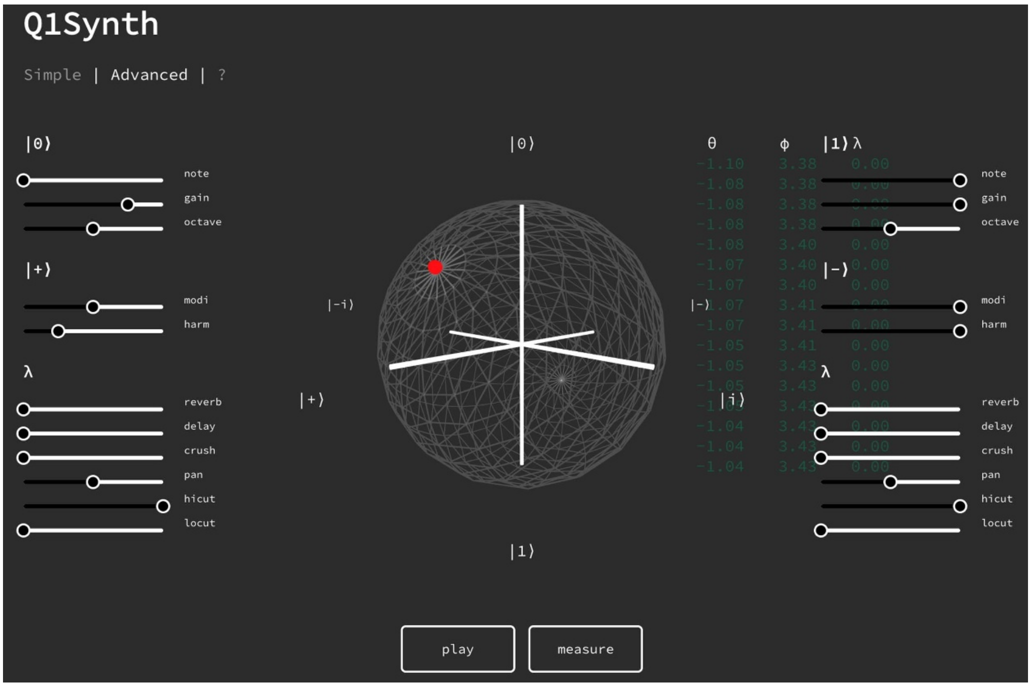

Figure 11. Q1Synth is presented on a computer screen as an artistic representation of a Bloch sphere.

by rotating this sphere and measuring the portrayed qubit using a mouse or a touchscreen. Alternatively, an external MIDI controller can be used, e.g. a VR glove.

To play the instrument, one presses the button 'play'. This puts the qubit in rotation mode. While the qubit is rotated, a continuously changing sound is produced. The instrument has a 'measure' button. When the performer presses this button, the instrument generates a program to create a quantum state corresponding to the current position of the state vector of the qubit, represented by a red dot. Then, it sends the program to a quantum computer over the cloud for processing (figure 13). The quantum computer subsequently runs the program and returns the measurement result (c_0). Then, the vector moves to either north or south, depending on the result of the measurement. We took the liberty of making the vector move in slow motion and this is accompanied by a respective sound. Once the vector reaches the destination the sound ends. Pressing the 'play' button again recommences the process, and so on. Currently, Q1Synth connects to an IQM quantum computer based in Finland.

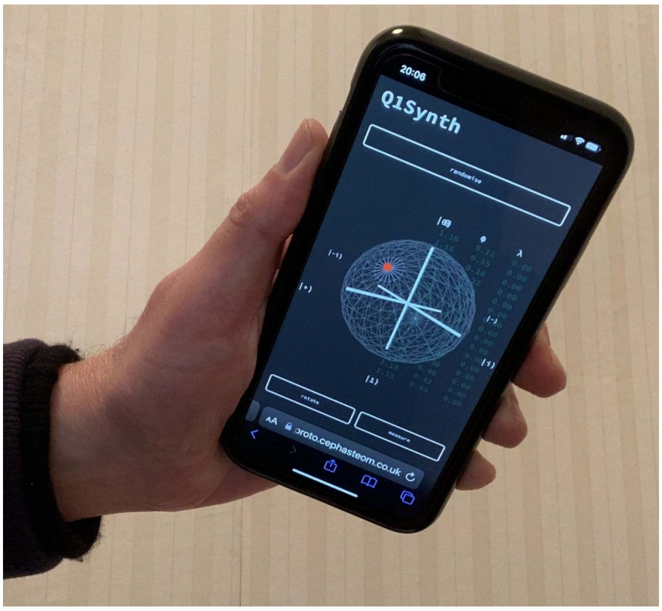

Figure 12. Q1Synth can also be used on a mobile device.

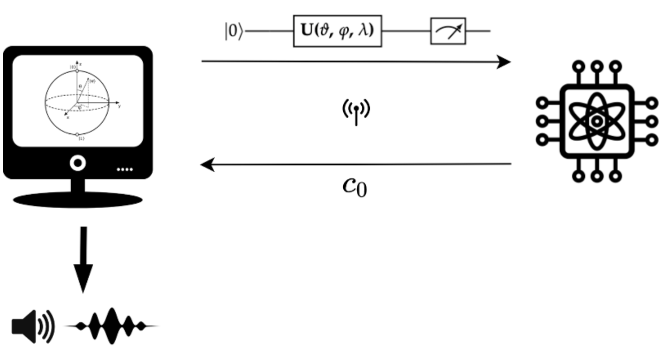

Figure 13. For measurement, Q1Synth builds a quantum program and sends it to an IQM quantum computer over the cloud.

The main technical difference between Q1Synth and Yale's work is that the latter involved sonification of data resulting from direct manipulation of physical qubits in real-time. Conversely, the performer does not rotate a physical qubit in real-time with Q1Synth. Rather, what is rotated is the Bloch sphere representation of a qubit. The system connects to a physical qubit only when the command for measurement is activated.

The coordinates of the Bloch sphere are processed in terms of Euler angles $(\theta, \varphi, \lambda)$. They describe rotation angles necessary to position the state vector around the sphere, starting from the north pole. The angle θ will determine the inclination (or latitude) of the vector, whereas φ is responsible for its azimuth (or longitude). In Q1Synth, the angle λ does not change the position of the red dot, but it will influence the orientation of the sphere. The λ is referred to as the *phase* of the vector.

Quantum computing programming tools typically provide a generic rotation gate **U**, with three *Euler angles*. Thus, any rotation gate can be specified in terms of **U**. For instance the gate **X** mentioned earlier is equivalent to $\mathbf{U}(\frac{\pi}{2}, -\frac{\pi}{2}, \frac{\pi}{2})$.

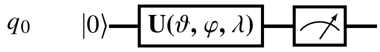

Figure 14. Q1 Synth generates a quantum circuit, which applies a **U** gate to a single qubit q_0. The dial at the end represents the measurement operator. Typically, the qubits start in the ground state $|0\rangle$.

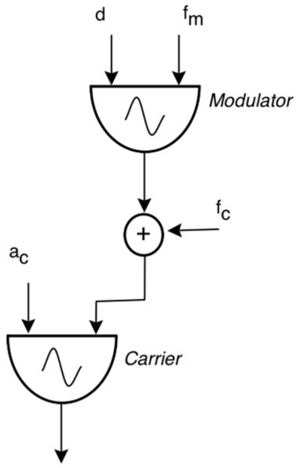

Figure 15. A simple FM synthesis architecture with two oscillators: a modulator and a carrier.

A quantum computing program is often depicted as a circuit with quantum gates operating on qubits (figure 14). When the measuring command is detected, Q1Synth builds a quantum circuit that implements a **U** gate. The coordinates of the sphere at the moment the 'measure' button is activated define the angles for the **U** gate.

At the time of writing, Q1Synth uses three types of sound synthesis techniques [90]: frequency modulation (FM), additive synthesis and granular synthesis. Let us look in some detail at how the FM method works.

In FM, the frequency of a waveform referred to as the *carrier*, is altered with a modulating signal, referred to as the *modulator* [17]. The most basic FM synthesiser comprises two sine wave oscillators: one acting as the modulator and the other as the carrier signal, respectively (figure 15). In this case, whereas d defines the amplitude and f_m the frequency of the modulator, a_c defines the amplitude of the carrier. The sinusoidal output from the modulator is added to the frequency of the carrier, f_c. The carrier is the signal that we hear directly. The modulator is heard only indirectly because its output is added to the base frequency of the carrier. When the frequency of the modulator is in the audio range, numerous additional partials, or sidebands, are added to the spectrum of the carrier's wave.

The system uses the angles (θ, φ and λ) of the Bloch sphere to interpolate parameter values for the FM algorithm. The user can define which synthesis parameters each angle will control. For instance, the angle θ (inclination) could be assigned to control f_c and a_c, the angle φ (azimuth) to control f_m and d, and the angle λ (phase) to control an added vibrato affect.

Q1Synth provides sliders on either side of the Bloch sphere to specify the range of values for each parameter (figure 11).

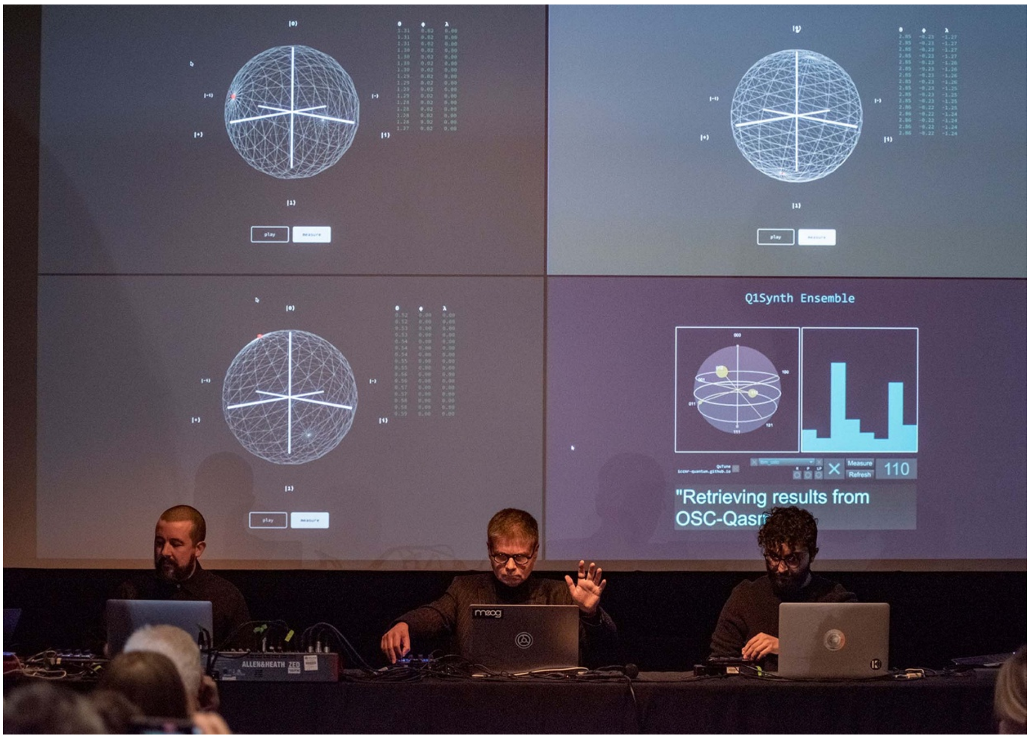

Figure 16. Premiere of *Spinnings* at the Goethe-Institut London, starring Peter Thomas (on the left), this author (in the middle) and Paulo Itaboraí (on the right).

Each sphere axis is assigned a pair of slider banks, enabling the values at each extreme to be interpolated as the sphere is rotated. For example, when the state vector is pointing north, ($|0\rangle$ position), the values for f_c and a_c will correspond to the sliders on the left side of the screen. When the state vector is pointing south ($|1\rangle$ position), these values will correspond to the sliders on the right. When the sphere is between these poles the system interpolates between each slider pair, according to the angle of inclination.

To render the movement of the vector into sound after a measurement, Q1Synth performs a complex, multidimensional modulation, interpolating between all parameters simultaneously until the red dot reaches its final destination, at which point the sound fades out. All interpolations are linear.

I used Q1Synth to compose *Spinnings*, which received its premiere at the Goethe-Institut London on 6 December 2022. The piece uses three Q1Synth instruments (figure 16), each of which is played by a different performer. In this case, the performers create quantum states and measurements with three qubits. A recording of *Spinnings* and an accompanying video are freely available online [109, 111].

6. *Qubism*: quantum generative music

Qubism is a sonata-like piece in three movements for chamber orchestra and electronics, composed in 2022. It was premiered by London Sinfonietta, at Kings Place, London, in June 2023 17. I composed this piece using two generative music systems developed with collaborators (e.g. Brian Siegelwax and Hector Miller-Bakewell) and my team at ICCMR. One, called *QuPoly* generates musical improvisations using a bespoke quantum AI system programmed to learn musical rules from examples. The other, called *PQCA Music*, generates musical forms using Partitioned Quantum Cellular Automata (PQCA).

The works reviewed heretofore resulted from some form of musification of data from quantum phenomena or software models thereof. Conversely, *Qubism* resulted from leveraging computation with quantum mechanics to generate music.

Generally, composers working with computers today either use AI to learn to imitate the style of existing music or use models that harness specific pattern-generating algorithms to create original music.

The great majority of research in AI for musical composition has been focused on developing systems for imitating the style of existing music [81]. For instance, David Cope mentioned earlier [22] uses a technique, originally developed for linguistics, known as Augmented Transition Networks (ATN) [121]. More recently, [64] proposed deep learning neural networks [41, 43] to compose tunes imitating the ones used for training the network.

Conversely, abstract pattern-generating algorithms are suitable for generating novel music, rather than imitating existing music. These algorithms were not necessarily developed for music to begin with. But they often embody processes believed to express compositional processes of some sort. The art of generating music with such models depends on the design of effective methods to map their outcomes into music. Normally, this design is considered as part of the creative process of

Figure 17. Premiere of *Qubism* by London Sinfonietta in June 2023.

composing and it is usually bespoke for specific compositions. Various composers have programmed computers to generate music using set theory, probabilistic methods, chaos, fractals, and so on [31, 49, 70, 122, 123]. In the same vein, Artificial Life (or A-Life) models have been developed to generate music [87, 88]; e.g. cellular automata and genetic algorithms [47, 56].

Emerging quantum computing technology is certain to open new avenues for both approaches. Whereas much research is being conducted into harnessing quantum computing for machine learning [104], which will, of course, be useful for the *AI approach* music, the *abstract generative approach* is bound to enjoy a fertile renaissance. *Qubism* inaugurates the exploration of these approaches in the realm of quantum computing. The piece has three movements. A recording of the premiere is freely available online [99].

6.1. QuPoly

QuPoly extracts sequencing—or transition—rules from input music and represents them in terms of quantum circuits. Then, it runs those quantum circuits to generate musical responses (figure 18). Once the system has learned some rules, it can generate as many new tunes, of virtually any length, as required. Essentially, the learning component of QuPoly is done classically but the generative component is done quantumly.

First, let us examine what the rules extracted by the system look like. Consider the system extracted the following rules from a short tune:

- $C_3 \Longrightarrow D_3(25\%) \vee G\sharp_3(25\%) \vee C_4(25\%) \vee D_4(25\%)$
- $D_3 \Longrightarrow C_3(30\%) \vee E_3(70\%)$
- $E_3 \Longrightarrow D_3(25\%) \vee F\sharp_3(25\%) \vee A\sharp_3(25\%) \vee C_4(5\%) \vee D_4(20\%)$
- $F\sharp_3 \Longrightarrow E_3(100\%)$
- $G\sharp_3 \Longrightarrow C_3(30\%) \vee A\sharp_3(70\%)$
- $A\sharp_3 \Longrightarrow E_3(33\%) \vee G\sharp_3(33\%) \vee C_4(34\%)$
- $C_4 \Longrightarrow C_3(30\%) \vee A\sharp_3(70\%)$
- $D_4 \Longrightarrow C_3(20\%) \vee E_3(80\%)$.

The second rule states that there is a 30% chance that note D_3 would be followed by note C_3 and a 70% chance that it would be followed by note E_3. The symbol '$\vee$' stands for 'or' and the percentage figure in parenthesis next to the notes is their weight coefficient, expressed here in terms of probability of occurrence. Thus, if we give QuPoly one of the above initial notes (on the left side of the arrow), the system will evaluate the respective rule and produce a new note. This new note is subsequently used to pick a rule to generate another note, and so on.

Figure 19 depicts the quantum circuit for the second rule. This circuit prepares three qubits to produce a quantum state representing the probability distribution of the rule. To enable this, the system represents each distinct note with a binary code. The larger the lexicon of notes, the higher the number of digits required to encode the lexicon. In the present example, the lexicon has eight notes. Thus, three digits are sufficient, as follows:

- $C_3 \Longrightarrow 000$
- $D_3 \Longrightarrow 001$
- $E_3 \Longrightarrow 010$
- $F\sharp_3 \Longrightarrow 011$
- $G\sharp_3 \Longrightarrow 100$
- $A\sharp_3 \Longrightarrow 101$
- $C_4 \Longrightarrow 110$
- $D_4 \Longrightarrow 111$.

Therefore, in terms of the binary representation, the second rule is expressed as $001 \Longrightarrow 000(30\%) \vee 010(70\%)$. In this case, the respective circuit in figure 19 prepares the qubits in such a way that they would most likely produce either 000 or 010 when they are measured.

The circuit is read from the left to the right. The various gates **RY** tell the system how to rotate the qubits q_0, q_1 and q_2 along the meridian axis. Upon measurement, the circuit should ideally output either 000 with 30% probability or 101 with 70%. The ordering of the qubits is $q_2q_1q_0$.

For *Qubism*, there are moments on the score indicating when QuPoly listens to an instrument and generates responses. Figure 20 shows a passage illustrating this interaction. In this case, the system listens to a phrase played on the violin and generates a response lasting for circa 20 seconds, which is played on synthesised percussion. I connected QuPoly to an

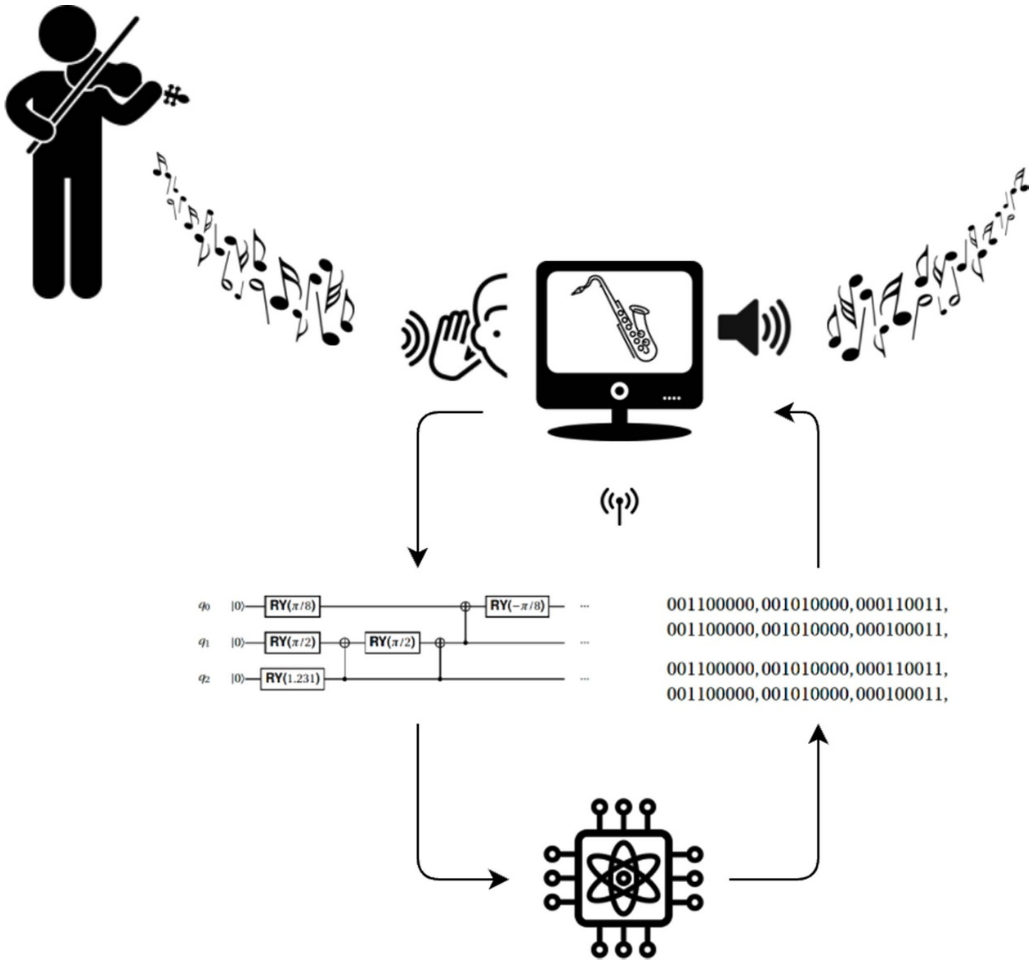

Figure 18. The QuPoly system.

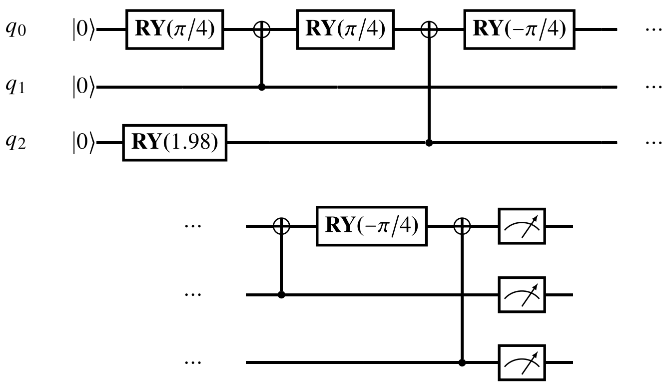

Figure 19. Circuit for the rule $100 \Longrightarrow 000(30\%) \vee 101(70\%)$.

IBM Quantum machine on the cloud to run the circuits for the premiere of *Qubism*.

As a previous attempt at the design of quantum algorithms to generate music with transition rules, I cite the *Basak-Miranda algorithm* [78], which leveraged a property of quantum mechanics known as constructive and destructive interference to compute the rules. This algorithm is based on the well-known Grover's algorithm [42], which has become a favoured example to demonstrate the advantage of quantum computing for searching for information in databases. However, the Basak-Miranda algorithm is limited to using equal weight distribution between the possible next notes. For instance, two possible notes would have a $50\% \times 50\%$ distribution between them by default. Yet another approach, referred to as *quantum annealing*, is discussed in section 7.

6.2. PQCA music

Cellular Automata, or CA, are discrete dynamical systems often described as counterparts to partial differential equations, which are suitable for modelling continuous dynamical systems. CA have been used to model phenomena in a variety of fields, including image-processing [95], ecology [48], biology [37], sociology [36], vocal production [89] and indeed music [47].

Stanislaw Ulam and John von Neumann conceived the notion of CA in the 1960 s [16]. They were looking into developing an abstract model that could capture the behaviour of biological organisms and the process of self-replication. In practice, a CA model is implemented as an arrangement of identical units, referred to as *cells*, that influence one another. This arrangement usually forms either a one-dimensional

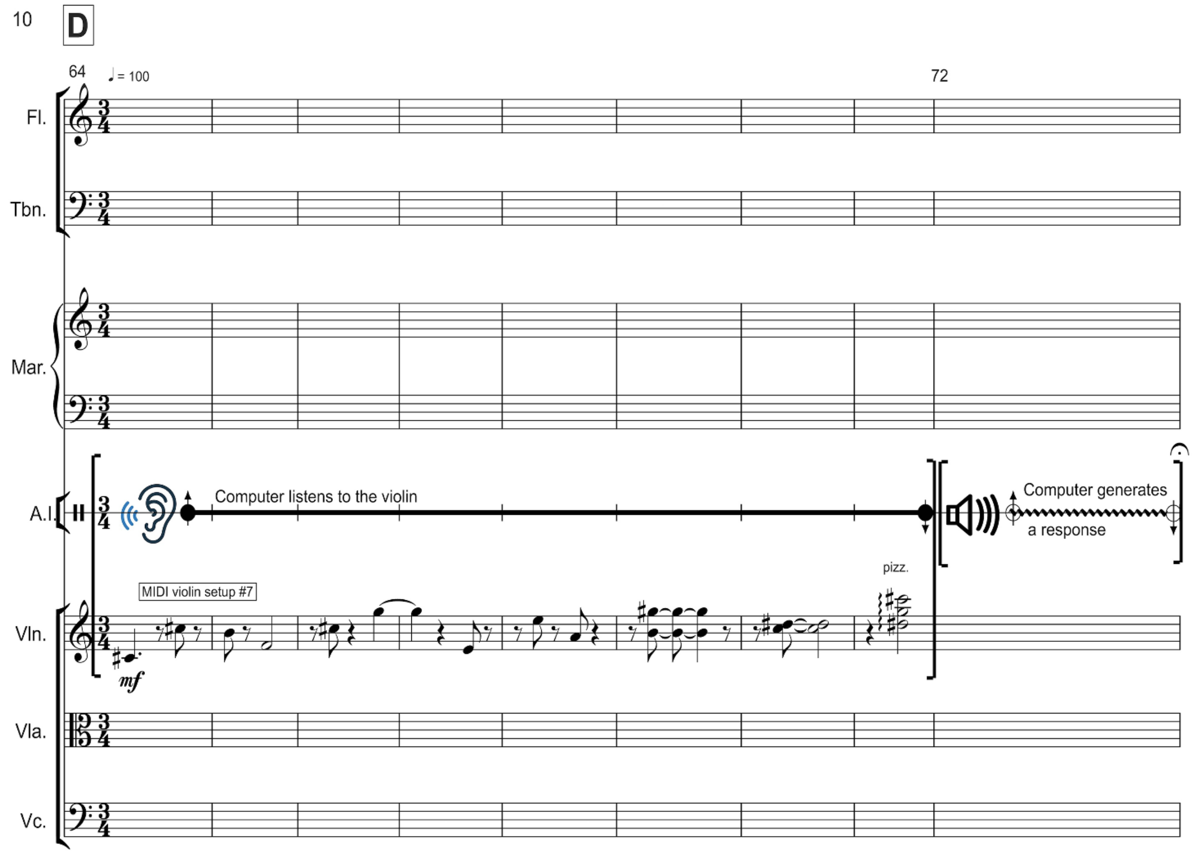

Figure 20. An excerpt from the score for *Qubism*.

array or a two-dimensional matrix of integer variables. An algorithm expressing transition rules operates on all cells simultaneously to alter their values.

Transition rules take into account the values of cells that are next to or surrounding each cell. The number of possible values for the cells can represent objects, properties, or processes, depending on what is being modelled. The functioning of a CA-based system is normally monitored on a computer screen as a sequence of changing checkered patterns, following the tick of a virtual clock, like in an animated film.

Quantum cellular automata (QCA), as its name suggests, are CA for quantum computers. They were introduced as an alternative paradigm for quantum computation and were shown to be universal: QCA can function as a quantum Turing machine. Please, refer to [38] for a theoretical discussion about QCA and universal computation.

In QCA, the cells are implemented as qubits. However, implementing a QCA with currently available quantum hardware is not trivial. The difficulty lies in updating all qubits in the same cycle of the automaton. This requires memory. For instance, if we attempt to implement a quantum version of any classical CA algorithm, we would need to store a copy of the original state of a qubit to update the others. However, this is not allowed with a quantum computer because of the no-cloning theorem, which states that an arbitrary unknown quantum state cannot be copied [93]. Various approaches have been proposed to get around this problem [38]. One of them is referred to as PQCA.

1	2	3	4	5	6	7	8	9	10	11	12

Figure 21. A bar of twelve PQCA cells.

For a rigorous theoretical discussion about PQCA, in particular, to learn more about their non-classical behaviour, please refer to [4, 52]. Only the practical aspects of implementing PQCA for modelling musical composition will be discussed below.

A CA at a certain time t is characterized by two properties: its current state and an update step that establishes the values of all its cells at time $t+1$. Thus, an update circuit must, in theory, be applied to all qubits of the automaton simultaneously at each time step t.

In PQCA, one or more *partition* schemes are applied to split an arrangement of cells into tessellating *supercells*. Then, *update frames* are defined using the partitions and *local circuits*. Those *update frames* subsequently form a *global update circuit*. As an example to picture this, consider a one-dimensional PQCA characterised by a bar of twelve cells. This is depicted in figure 21.

Figure 22. At the top is a bar of twelve cells partitioned into six supercells, two cells long each. At the bottom is a shifted version of the partition.

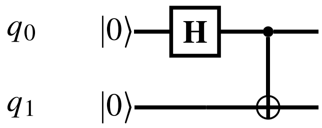

Figure 23. A simple local updating circuit for a supercell comprising two qubits.

Figure 22 shows two examples of partition schemes. At the top is a partition consisting of six supercells, with two cells long each. Let us notate this as follows:

$$\{[1,2],[3,4],[5,6],[7,8],[9,10],[11,12]\}. \tag{4}$$

And this is bottom is a shifted version of the partition above (equation (4)):

$$\{[2,3],[4,5],[6,7],[8,9],[10,11],[12,1]\}. \tag{5}$$

Now, let us define a local circuit for the one-dimensional PQCA considering the partitions shown in figure 22. A local circuit needs to have the same number of qubits as the number of cells in the target supercells. In this case, the supercells contain two cells each. Therefore, the local update circuit must work with two qubits. Figure 23 shows a two-qubit circuit, with a Hadamard gate, **H**, and a controlled NOT gate, **CX**. The **H** gate puts qubit q_0 in a state of equal superposition, which means that there is an equal chance of returning 0 or 1 upon measurement. The **CX** conditions the value of qubit q_1 to the value of q_0. Thus, assuming that both qubits start as $|0\rangle$, if q_0 is measured equal to 1, then q_1 will also be equal to 1. Otherwise, q_1 returns a 0. This local circuit is then tessellated through the supercells forming an update frame.

Figure 24 shows an example of a global update circuit using two update frames, one based on the partition shown at the top of figure 22 and the other using a shifted version of the partition, which is shown at the bottom of the figure. For this example, the same local circuit applies for both update frames; but there could be two distinct local circuits if desired.

An example showing four cycles of the PQCA from given initial cell values—or qubit states—is shown in figure 25. The automaton applies the global update circuit to the current states of the qubits, and the measured output is used to arm the qubits with states for the next cycle, and so on.

The art of generating music with CA hinges on the methods to convert their outputs into patterns of musical notes. It is here that composers can experiment with different mapping designs [79, 91]. An instance of a mapping designed for *Qubism* is presented below; please see [76] for more details, including the use of a two-dimensional PQCA.

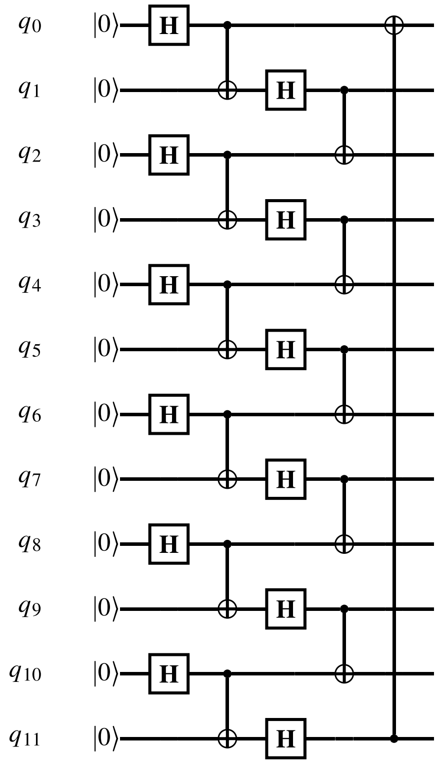

Figure 24. An example of a global quantum circuit for a one-dimensional PQCA.

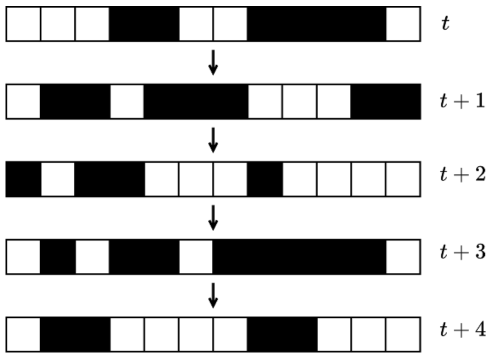

Figure 25. An example of five cycles of a one-dimensional PQCA. Note: the colours of the squares stand for the measured values of the automaton. A black cell represents the number 1 and a white one the number 0. Reproduced from [79], with permission from Springer Nature.

The one-dimensional PQCA for *Qubism* consisted of a bar of 18 cells. Here, the mapping scheme generates clusters of musical notes. Each cycle of the PQCA produces an eighteen digits long string, which is converted into either a cluster or a rest; i.e. silence.

A cluster is defined as a group of notes that are played simultaneously. In the case discussed here, it can have up to 12

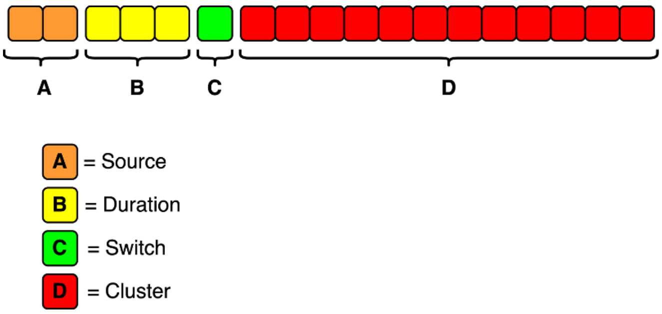

Figure 26. Coding scheme for representing a cluster of musical notes. Reproduced from [79], with permission from Springer Nature.

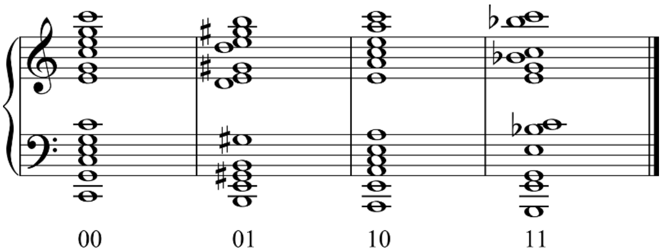

Figure 27. A set of four sources of pitches to build clusters.

notes. Conversely, a rest is when a cycle does not generate any notes at all.

To encode musical information, the string is split into four sections (figure 26). Each section forms a code representing a property of the cluster, as follows (digit order is from the left to the right of the string):

- Code A (digits 1 and 2) = defines the source of the notes. There are four different sources to choose from (figure 27).
- Code B (digits 3, 4 and 5) = defines the duration of the cluster. There are eight different durations to choose from (figure 28).
- Code C (digit 6) = rest switch.
- Code D (digits from 7 to 18) = defines the notes of the cluster; these notes are picked from the source defined by Code A.

The notes for a cluster are picked from one of four sources, according to Code A. Hence this code requires two digits to encode four options. The composition *Qubism* used two sets of sources, one of which is shown in figure 27.

There are eight options for the duration of a cluster, which is defined by Code B (figure 28). It requires three digits to encode eight options. And Code C establishes if the respective cluster is active (Code C = 1) or constitutes a rest (Code C = 0).

Note that in figure 27 the sources are already displayed as clusters of twelve pitches. Formed by the last twelve digits of the string (from 7 to 18), Code D defines which notes in the selected source will constitute the respective cluster.

The positions of digits equal to 1 in the string (from the left to the right) correspond to the positions of the notes to be picked from the source (from the bottom to the top). For instance, let us consider Code D = {011001001001}. This instructs the system to pick the second, third, sixth, ninth and twelfth notes to form the cluster, as illustrated in figure 29. However, if Code C happens to be equal to 0, then the system would output a rest, respective to the duration of the cluster instead.

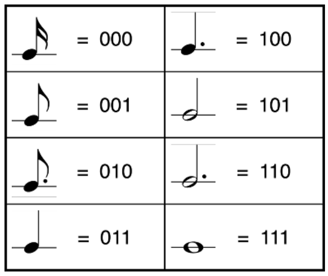

Figure 28. Codes for note durations.

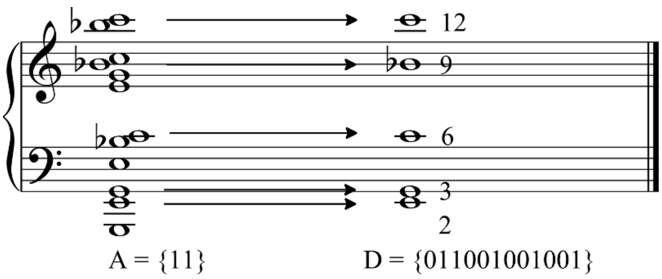

Figure 29. An example of a cluster generated by the bitstring [111111011001001001].

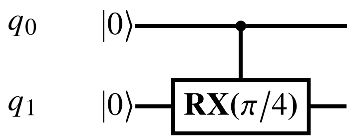

Figure 30. The local circuit for the first update frame of the example uses a conditioned **RX** gate.

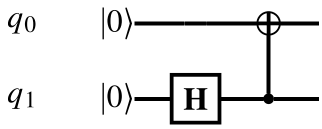

Figure 31. The local circuit for the second update frame of the example.

Let us examine an example using the sources of pitches and duration codes shown in figures 27 and 28. In this case, the PQCA bar of 18 cells was partitioned into nine pairs of super-cells. Similarly to the instance in figure 22, here there are two partition schemes also. One of which is a shifted version of the other by one cell. Two update frames were defined, one for each of the partitions, respectively. Figures 30 and 31 show the respective local circuits for the update frames. And the global update circuit is depicted in figure 32.

The qubits in a quantum circuit typically start in the ground state $|0\rangle$. However, here I wanted to initialise the cells of

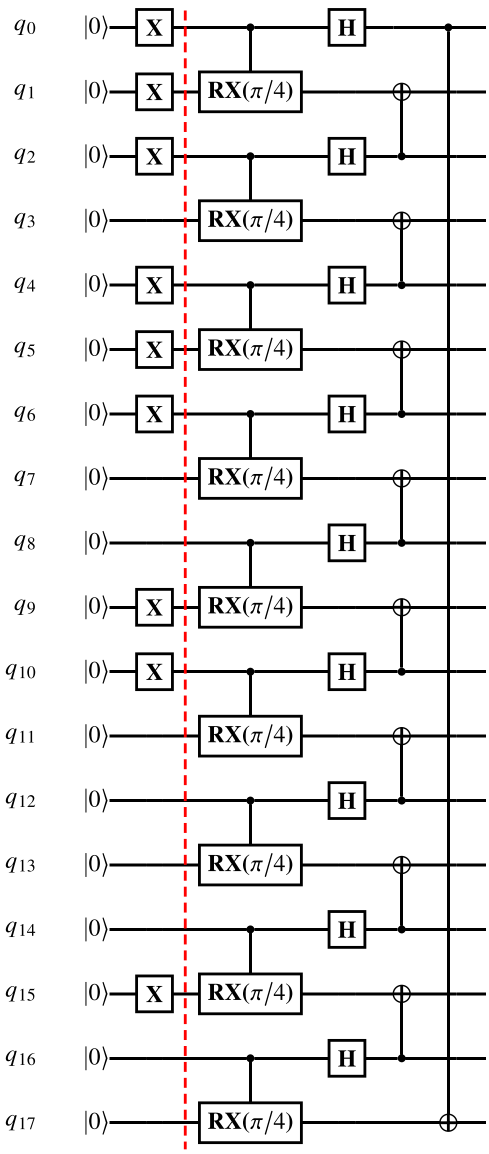

Figure 32. The resulting global update circuit. On the left side of the red dash-line are the gates to initialise the qubits.

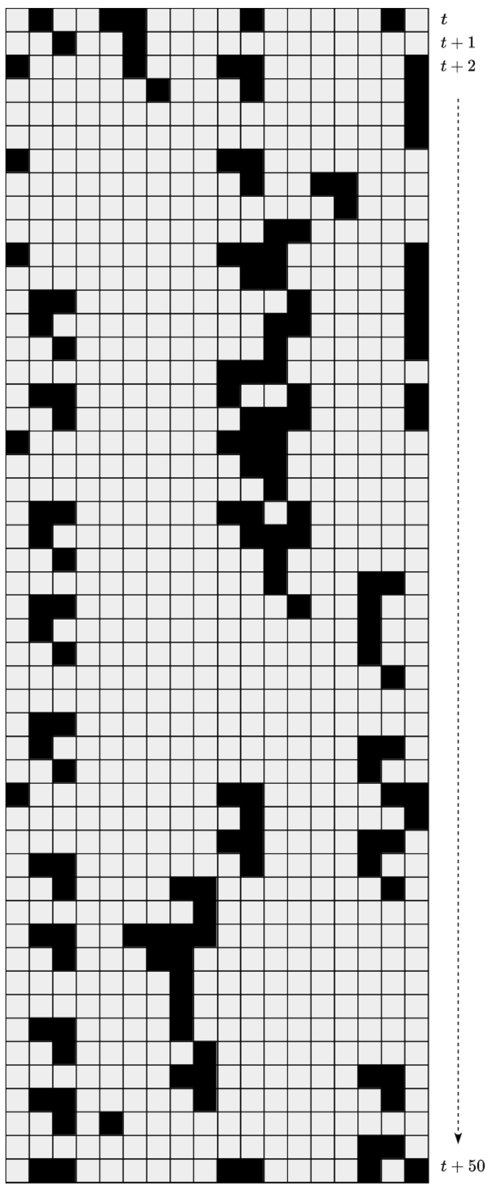

Figure 33. The resulting PQCA cellular pattern.

the PQCA with random values. To this end, I implemented a simple quantum dice to generate random values to initialise them. As an example, consider the following initial cell values: $[1,1,1,0,1,1,1,0,0,1,1,0,0,0,0,1,0,0]$. In this case, the system needes to adapt the global update circuit before processing it. When a cell is to value one, the system adds an **X** gate at the beginning of the circuit to flip the respective qubit to $|1\rangle$. Figure 30 shows the circuit for the first cycle. On the left side of the dashed line are the gates to initialise the qubits. Such an update must be done for every cycle of the PQCA to arm the qubits with the results from the last measurement. Figure 33 shows an example of a cellular pattern resulting from 50 cycles and figure 34 depicts an example of a musical rendering. (Note: this musical example was not rendered from the pattern in figure 33.) Obviously, the musical patterns in figure 34 are raw materials, which I subsequently orchestrated as shown in figure 35.

7. *Superposition*: composing with quantum annealing

Quantum annealing is a computational technique that leverages the adiabatic principle in quantum mechanics to solve optimization problems [112]. The adiabatic principle describes what happens when the total energy of a quantum system—referred to as its Hamiltonian—is subjected to gradual changes [11, 40]. Allegedly, this technique runs best on purpose-built quantum annealers.

D-Wave Systems is a well-known company offering quantum annealing technology commercially and for research [33]. However, the D-Wave machines are primarily designed for quantum annealing. They are not intended for other purposes.

In 2016, Kirke composed a piece entitled *Superposition* for a soprano and electronic sounds generated with quantum

Figure 34. A resulting rhythmic sequence of clusters.

Figure 35. An excerpt from the musical score for *Qubism*.

annealing. It was premiered by singer Juliette Pochin at the Port Eliot Festival, in Cornwall, UK [59]. For more information about the system developed at ICCMR for this composition, please refer to [58].

Other projects aimed at generating music with quantum annealing running on a D-Wave machine include the *Algorhythms* system developed by Chuharski [18] and the research developed by Arya *et al* [6].

In simple terms, the adiabatic principle predicates the following: consider a quantum system in a stationary state of an initial Hamiltonian. If the system is subject to slow perturbation over time, then it would slowly adapt its probability density. Its Hamiltonian would change smoothly and the system would evolve gradually.

Probability density is the probability of finding a particle at a particular location in a given space. With slow perturbation, the probabilities of finding the system in its different energy eigenstates evolve smoothly without any sudden transitions. Conversely, rapid perturbations can lead to deviations from the adiabatic principle, resulting in non-adiabatic transitions, decoherence, and energy dissipation.

Quantum annealing relies on such smooth change of the Hamiltonian of a quantum system to solve problems. A problem is encoded in the ground state of a physical system, typically a quantum system represented by a bunch of qubits. Then, the Hamiltonian of the system is changed in such a way that its ground state evolves from one that encodes the problem to one that represents the solution.

Quantum annealing is suitable for *propositional satisfiability problems* and other combinatorial searching problems. A propositional satisfiability problem is the problem of determining whether a set of logic expressions is true. The issue of selecting suitable options in musical sequencing rules, such as the ones discussed in section 6.1, can be thought of as a propositional satisfiability problem [6, 18].

Quantum annealing can be implemented using the Hamiltonian of quantum Ising models [108]. Ising models can represent the physics of magnetic materials based on the molecules within them. As well as electrical charge, each of these molecules has a property known as spin. Ising models describe the behaviour of interacting spins, incorporating quantum mechanical principles, such as superposition and entanglement, into the description of the system [102]. Their spin can be -1 or $+1$.

The Hamiltonian $H(s)$ of a collection of molecules with spin s_i can be modelled by equation (6), where $s_i \in \{-1,+1\}$ denotes the spin of molecule i and h_i represents an external magnetic field. $J_{(i,j)}$ represents the interactions between each molecule and its nearest neighbour. In a nutshell, quantum annealing is aimed at finding values of s_i that minimise the Hamiltonian for given h_i and $J_{(i,j)}$

$$H(s) = \sum_i h_i s_i + \sum_{i<j} J_{(i,j)} s_i s_j. \tag{6}$$

To implement equation (6) on a quantum computer, s_i can be substituted by the Pauli-Z operator σ_i^z acting on qubit i (equation (7)). The Pauli-Z operator σ_i^z would be the equivalent of an $\mathbf{RZ}(\boldsymbol{\theta})$ gate in gate-based computing

$$H(s) = \sum_i h_i \sigma_i^z + \sum_{i<j} J_{(i,j)} \sigma_i^z \sigma_j^z. \tag{7}$$

Now, let us imagine two Hamiltonians, H_0 and H_1, and interpolate slowly between them. Assume that the first, H_0, is a transverse field Hamiltonian (equation (8)), where σ_i^x is the Pauli-X operator acting on qubit i; this would be the equivalent of an $\mathbf{RX}(\boldsymbol{\theta})$ gate in gate-based computing. The ground state of H_0 is in equal superposition of all basis states.

$$H_0 = \sum_{i=1} \sigma_i^x. \tag{8}$$

And the second, H_1, is the classical Ising model in equation (7). Then, a time-dependent Hamiltonian can be defined by transitioning between H_0 and H_1, with $t \in [0,1]$ (equation (9)):

$$H(t) = (1-t) H_0 + t H_1. \tag{9}$$

Thus, at $t = 0$, one obtains only the transverse field (H_0). At $t = 1$ one obtains the classical Ising model (H_1). One can transition from the ground state of H_0 to the ground state of H_1 by slowly changing t from 0 to 1, ensuring that the system stays in the lowest energy solution all the way through. Quantum annealing requires this process to be repeated several times. The solution to the problem should be the run that yielded the lowest energy.

Let us examine an example of using quantum annealing to generate sequences of musical notes. The task is to pick notes from a given set: $\{C_3, G\sharp_3, D\sharp_3, F_3, C\sharp_3, F\sharp_3, E_3, A\sharp_3\}$. However, a system for musical composition normally works with rules governing how its components—in this case, musical notes—should be organised. These rules establish constraints, defining that certain elements are more suitable than others to appear at specific moments in a sequence. In music AI circles [81] such rules are often represented as a Markov network with conditional transition probabilities [26].

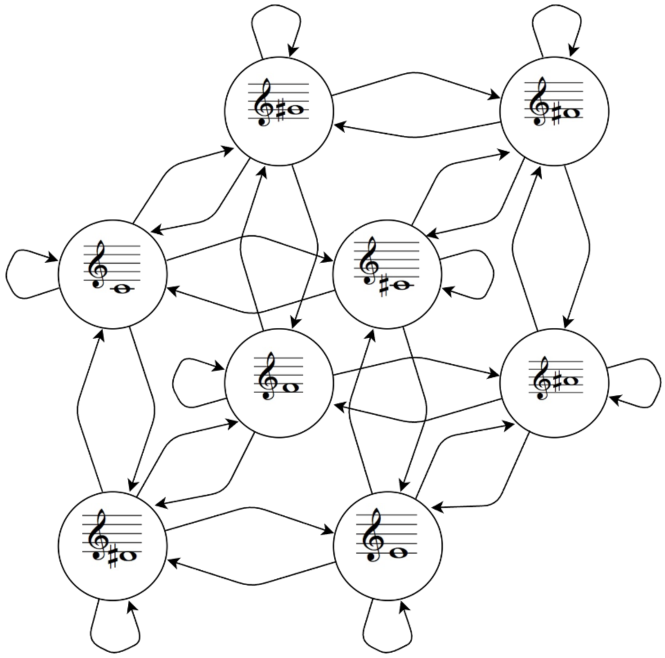

Figure 36. A Markov network representation of musical rules. Not shown in the figure are the different probability coefficients for each transition represented by the edges.

An approach to using rules to generate music with quantum computers was introduced in section 6.1. Here the problem is defined as a random walk optimisation problem. It is tackled with a technique called markov random fields, or MRF, which is akin to the Ising model. Incidentally, the Ising model is regarded as a special case of MRF [57].

Indeed, musical rules comparable to those depicted in section 6.1, can be thought of in terms of a Markov network. They can be represented as graphs with vertices and edges (figure 36). Nonetheless, for quantum annealing, they must be converted into MRF [57]. Quantum annealing needs relationships and dependencies between vertices instead of edges with directionality.

Converting a Markov network into an MRF entails removing the directionality of the edges and turning it into an undirected graph model, where edges represent pairwise interactions (figure 37). In MRF, each edge $[\gamma,\delta]$ can be one of the following states $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$. Each of these states is associated with a potential energy.

In short, the system creates a distribution of potentials for each note in the MRF. Then it runs the system for several times to sample that distribution for the lowest energy state. The note

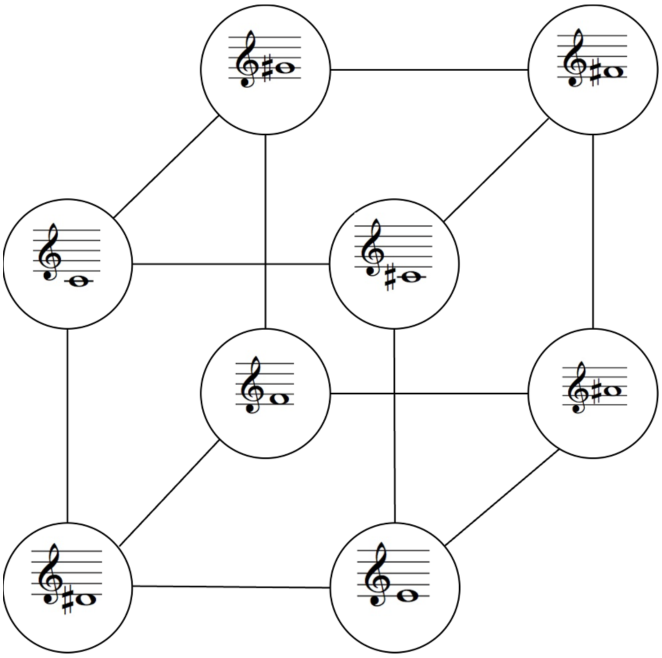

Figure 37. MRF conversion of the Markov network representation in figure 36.

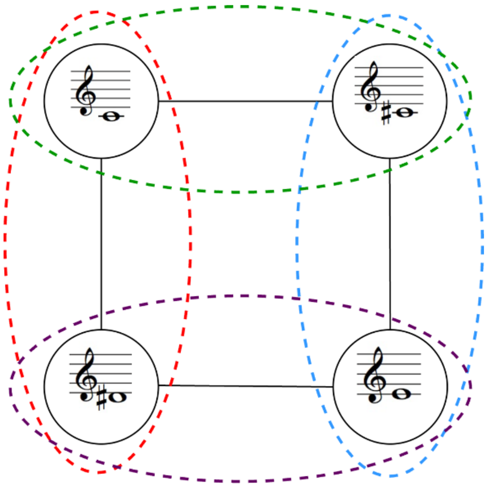

Figure 38. A partial version of the MRF from figure 37 illustrating pairwise vertex dependencies.

Figure 39. An example of a generated sequence.

with the lowest potential is picked as the next note to be played. This process is repeated as many times and the number of notes required in the sequence. An example focusing on three notes from figure 38 is given as follows:

- $(C_3, C\sharp_3) = [(0,0) = 10], [(0,1) = -8], [(1,0) = -3], [(1,1) = 10]$
- $(C_3, D\sharp_3) = [(0,0) = 10], [(0,1) = -9], [(1,0) = -2], [(1,1) = 10]$.

In the above example, desired states were hard coded with lower potentials to begin with. State 1 corresponds to the note being played. The edges (0, 0) and (1, 1) were purposely set with high potential to evade cases where the current note would be repeated or two notes would be played simultaneously. An example of a generated sequence, considering the MRF. in figure 37 is shown in figure 39. In this case, the system does not deal with rhythm. All notes have identical duration.

In terms of assigning potentials, this can be done in many ways. Naturally, these should reflect the constraints of musical rules governing how the note components should be organised. But transition weights could also be randomised at every time step. Specific transitions could be forbidden, or removed, to prevent specific notes from following some others.

8. Towards quantum audio

To be handled by digital electronic devices, audio needs to be represented digitally. The process of converting an analogue audio signal into a digital one is referred to as *audio sampling* [13].

Audio sampling is the action of taking snapshots of a continuous signal and storing the captured amplitude values as time-indexed samples (figure 40). Conventionally, those snapshots are taken in equally spaced time lapses. The speed of the lapses is referred to as the *sampling rate* of the digital audio. It establishes how many samples are taken per second. High-quality digital audio systems use a sampling rate of 48 000 snapshots per second (i.e. 48 kHz). A hypothetical signal lasting for one second is shown in figure 40, where it was sampled 50 times.

Furthermore, the time-indexed samples need to be *quantised*. But this has nothing to do with quantum mechanics. Rather, quantisation here has to do with the fact that the sampled amplitudes need to be restricted to a prescribed discrete scale. Quantisation is, in some ways, like sampling, but it operates alongside the vertical axis. Furthermore, quantisation is an approximation and therefore can lead to distortion of the signal if the resolution is not adequate.

In the example shown in figure 40, there are seven quantised values to round the amplitudes up or down. For practical purposes, a quantised signal is often shifted to make it fall entirely onto the positive domain. On the right side of figure 40, two different binary encodings for the amplitudes are shown: unsigned (i.e. transposed to the positive domain) and signed encoding, respectively.

The digitised samples are normally stored in an array. In the example in figure 41, the index t_n of the array corresponds to time. It specifies where the respective samples—or audio amplitudes a_n—are stored. To play back the sound, the system scans the array to read the samples and sends them to a digital-to-analogue converter (DAC) to reconstruct the original analogue audio.

The systems discussed above do not handle audio per se. For instance, the Yale team (section 4) and Q1Synth (section 5)

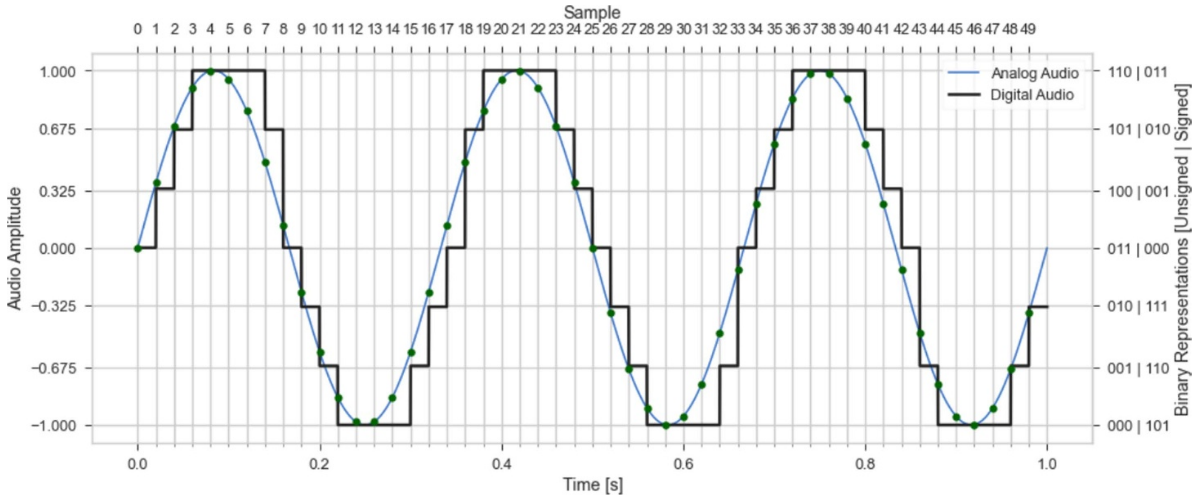

Figure 40. An analog sinusoidal wave and its respective quantisation. Reproduced from [54], with permission from Springer Nature.

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 41. Audio array visualization.

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

Figure 42. Digital to Quantum.

dealt with control parameters for processing sounds, which are encoded digitally, rather than sounds encoded quantumly.

However, it is not straightforward to represent sounds quantumly. Existing methods are tied to a digital representation. In other words, it is a Digital-to-Quantum conversion, rather than Analog-to-Quantum.

There are several technical limitations to encoding audio with qubits. For instance, at the time of writing, random access memory (RAM) technology to store data quantumly is underdeveloped. There is no such thing as a 'quantum random access memory' (QRAM) as yet. Even so, a few approaches for quantum audio representation have been proposed already [54]. Instead of storing the audio samples, methods are being developed to represent audio in terms of circuits.

An intuitive understanding of quantum audio representation is proposed in figure 42. The Dirac notation translating the time-index t_n as quantum states suggests how audio samples may be represented as quantum states (figure 42).

Most approaches to quantum audio representations proposed to date largely employ, in one way or another, a quantum register to create a superposition of time indexes for audio samples. This is referred to as a *time register*. Thus, each state will be an index, indicating a position in time, similar to a classical array. Then, any information associated to this state will encode the respective sample s (equation (10)). There are different approaches to doing this.

$$s|t_k\rangle \tag{10}$$

Let us assume a digital audio A, with $N = 2^n$, $n \in \mathbb{Z}^*$ samples, each of which is quantised to $[-2^{q-1}, -2^{q-1} + 1, \ldots, 2^{q-1} - 1]$. That is, there are 2^q possible values. Those samples could be represented in a quantum state $|A\rangle$ by creating a superposition of all of its possible states t (encoding time), weighted by their respective probability amplitudes (encoding the sample value), as shown in equations (11) and (12)

$$a_n(t_n) \longrightarrow \alpha_i |t_i\rangle. \tag{11}$$

In equation (12), think of $|A\rangle$ as doing the job of an array in digital sampling.

$$|A\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_2|2\rangle + \alpha_3|3\rangle + \alpha_4|4\rangle + \alpha_5|5\rangle + \alpha_6|6\rangle + \alpha_7|7\rangle. \tag{12}$$

Let us look closely at an approach to quantum audio representation, referred to as Quantum Probability Amplitude Modulation, or QPAM [54].

Suppose an audio signal of lenght N, with n (or $\lceil \log N \rceil$) qubits. QPAM maps each possible sample value of the audio onto a probability amplitude (equation (13))

$$|A\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle. \tag{13}$$

More generally, the squared sum of all probability amplitudes of the system should be equal to 1. As it was already mentioned earlier, given an arbitrary qubit $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$,

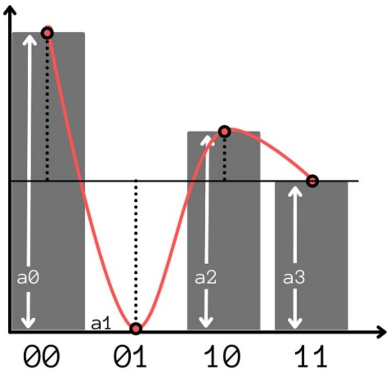

Figure 43. The black dotted line shows the digital audio sample, and the white line the histogram bins.

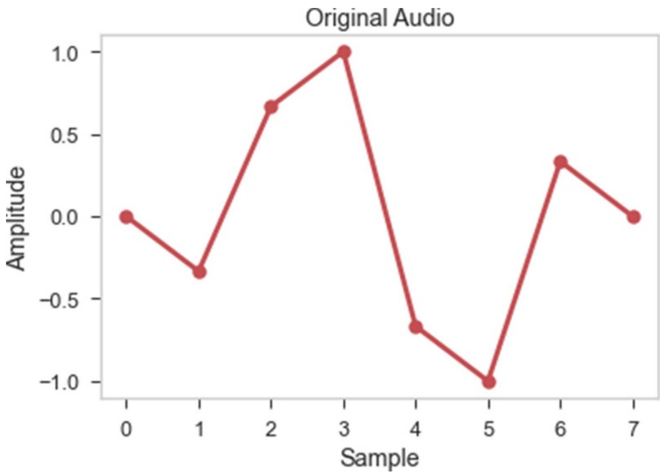

Figure 44. Hypothetic audio, represented using seven samples. Reproduced from [54], with permission from Springer Nature.

the probability of measuring the state $|0\rangle$ is $|\alpha|^2$ and the state $|1\rangle$ is $|\beta|^2$ with the proviso that $|\alpha|^2 + |\beta|^2 = 1$.

Mind, however, that although the audio samples a_n are amplitudes, these are not the same as the probability amplitudes α_i of quantum states. Thus, audio amplitudes need to be mapped onto quantum probability amplitudes. A schematic QPAM visualisation of a hypothetical signal is depicted in figure 43.

Consider the hypothetic snippet of audio depicted in figure 44. The amplitudes of the samples are numbers ranging between -1 and $+1$. But their sum would not necessarily add to 1. Therefore, the amplitudes need to be normalised. One way of doing this is to follow these steps:

- Step 1: Add 1 to all amplitudes a_n.
- Step 2: Divide the amplitudes by 2.
- Step 3: Divide the amplitudes again, by the sum of all of the amplitudes.
- Step 4: Take the square root of the results.

The amplitudes of our example in figure 44 are listed before the down arrow in figure 45. The normalised values are shown

0.0	−0.3	0.7	1.0	−0.7	−1	0.3	0.0

↓

0.3	0.2	0.5	0.6	0.1	0.0	0.4	0.3

Figure 45. Normalization process.

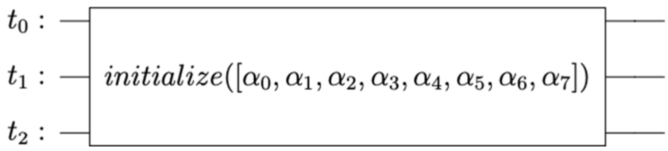

Figure 46. Initialising qubit states for QPAM quantum audio.

after the down arrow. Firstly, the sample amplitudes were shifted to the positive domain by adding 1. At this point, the amplitudes ranged between 0 and 2. Next, they were scaled down to fit the range between 0 and 1. Then, all their values are summed and every value is divided by the result of the sum. At this point, they are normalised: the sum of all amplitudes should be equal to 1. The final step is to turn these figures into probability values by taking their square root, as shown in equation (14).

$$\alpha_i = \frac{1}{\sqrt{g}}\sqrt{\frac{(a_i+1)}{2}}; \qquad g = \sum_k \frac{(a_k+1)}{2}. \tag{14}$$

As the information is stored in the probability amplitudes α_i, the qubits are prepared by setting them in the desired state of superposition.

Fortunately, most quantum computing programming languages, such as Qiskit [97], provide a command to initialise qubit states. Thus, the preparation of QPAM audio requires, first, the conversion of digital samples into probability amplitudes. Then, these probability amplitudes are used to create a unitary quantum gate (figure 46), which is decomposed into a set of machine-readable instructions. The preparation of an arbitrary n-qubit state normally requires $\mathcal{O}n$ simple operations. This implies that QPAM preparation needs $\mathcal{O}(\lceil \log N \rceil)$ quantum gates.

Figure 47 portrays a histogram from measuring qubits initialised to represent the audio shown in figure 44. This example was generated with Qiskit Aer simulator. And figures 47 and 48 show the respective audio reconstruction.

QPAM is referred to as a probabilistic representation because of its reconstruction mechanism. The reconstruction of QPAM audio requires the preparation of several identical quantum versions of it and the statistical analysis of many measurements. As QPAM information is stored in the probability amplitudes, the reconstructed signal is an approximation of the original one. Its accuracy is determined by the amount of measurements taken. The analysis returns a histogram of the measurements, from which the probability of measuring each state is calculated (e.g. figure 47). When considering how the

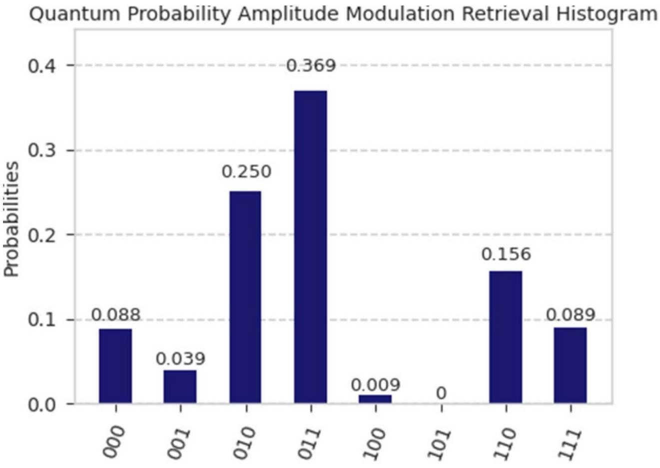

Figure 47. A hypothetical QPAM representation. Reproduced from [54], with permission from Springer Nature.

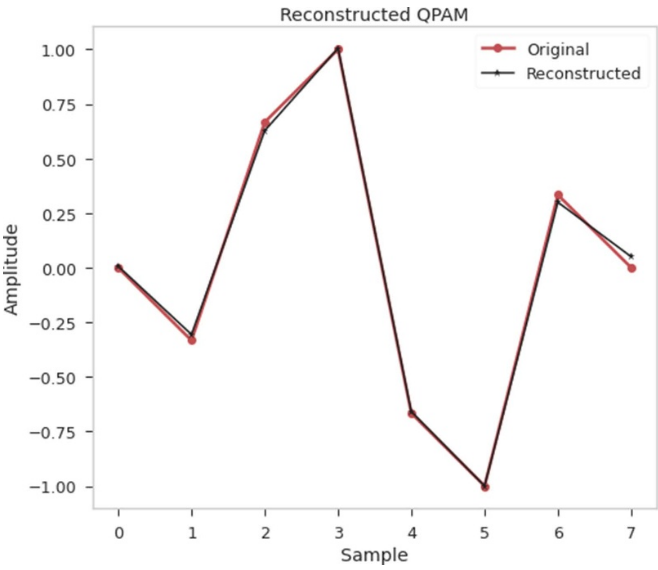

Figure 48. Original and reconstructed audio from the QPAM representation shown in figure 47.

system was prepared in the first place, the visual representation of the histogram already suggests the shape of the original signal. Next, the histogram values need to be converted to the range of the original audio, using an inverted version of equation (14) for a_i, as shown in equation (15).

$$a_i = 2g|\alpha_i|^2 - 1. \tag{15}$$

In equation (15), bear in mind that α_i is a complex number. Therefore, $\alpha_i^2 \neq |\alpha_i|^2$. The probability p_i of $|i\rangle$ being measured is given by $p_i = |\alpha_i|^2$. However, after completing the measurements, a histogram bin will not return p_i, but rather an approximation $\tilde{p}_i$. Therefore, to be more precise, equation (15) needs to be adapted, as shown in equation (16)

$$a_i = 2g\tilde{p}_i - 1. \tag{16}$$

The ability to represent audio quantumly paves the way for the development of quantum audio processors and sound synthesisers. Let us envisage an audio effect unit proposed by Paulo Itaboraí at ICCMR, referred to as *Geiger counter fffect*, which uses QPAM [54].

As already discussed earlier in this paper, when a system of qubits is measured, the outcome is probabilistic rather than deterministic. That is if the same quantum state is measured multiple times, a different result may be obtained for each measurement. To circumvent this, the same quantum circuit is run and measured many times to obtain a statistically robust result. The number of times a circuit is repeatedly measured for this process is referred to as *shots*.

Normally, thousands of shots are necessary to ascertain a result. However, this requirement can be twisted here to obtain sound effects. The quantum dithering effect is achieved by intentionally underestimating the number of shots necessary to retrieve a sufficient approximation of the audio in question. A few hypothetical examples using four qubits are shown in figure 49. Each of those histograms yields a different sound.

The bottom of figure 50, in red, portrays an example of the Geiger counter effect applied to the sound portrayed at the top, in blue. With subtle underestimations, the presence of a faint white noise is added to the original sound. This example was produced with 15 qubits.

Overall, quantum audio representation methods are of two categories, according to how audio amplitude information is encoded and reconstructed through measurements: coefficient-based and state-based. The former includes QPAM, discussed above, and SQPAM (Single Qubit Probability Amplitude Modulation). The latter includes QSM (Quantum State Modulation) and MQSM (Multichannel Quantum State Modulation). More information is available in [54].

9. Other initiatives and concluding remarks

The works reviewed in this paper are representative of the various approaches that have emerged so far to leverage quantum computing for music, namely:

- (a) Musification of quantum mechanical models running on classical hardware; e.g. section 3.
- (b) Musification of physical quantum mechanical systems; e.g. section 4.
- (c) Deployment of quantum algorithms to compose music or sound; e.g. section 6.2.
- (d) Deployment of quantum AI to generate music or sound; e.g. sections 6.1 and 7.
- (e) Development of quantum encoding and quantum audio signal processing; e.g. section 8.

There are a few other initiatives that were not detailed here due to lack of space. But with very few exceptions they would fall into one of the approaches listed above or a combination thereof. For instance, the system Quantum Music Playground, an educational tool for learning quantum computing concepts

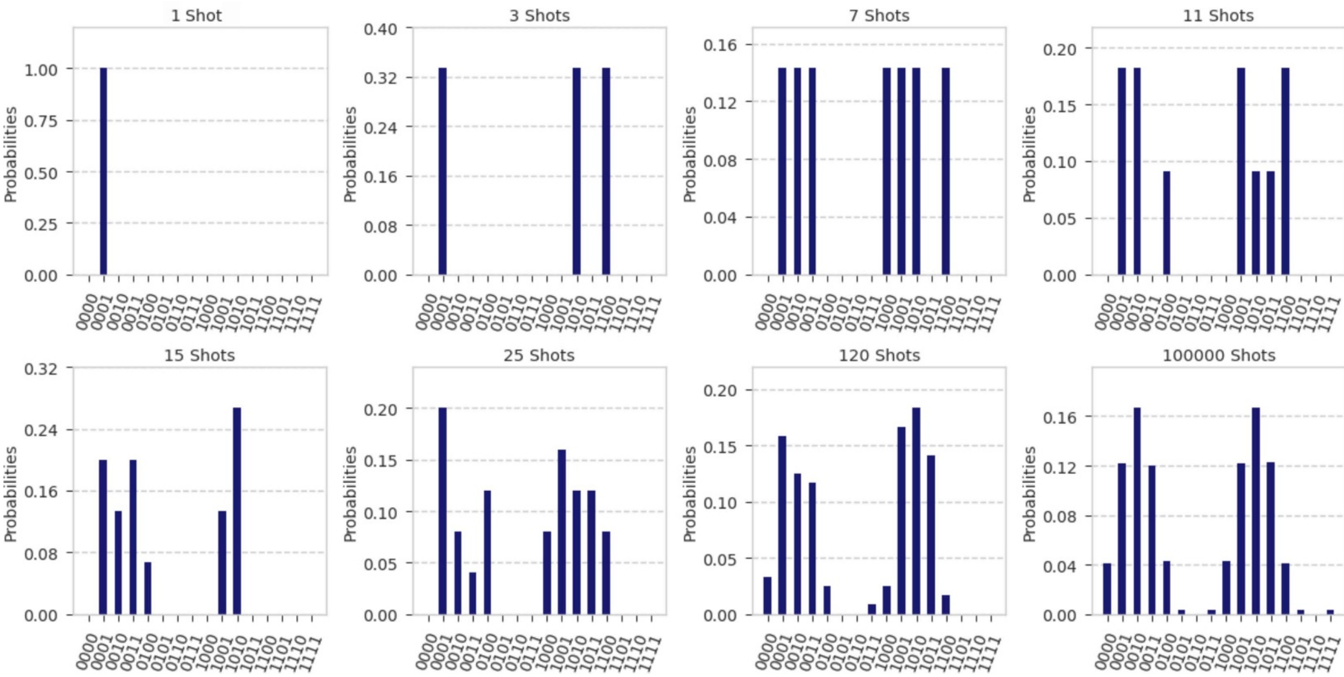

Figure 49. Histograms obtained by running and measuring the same circuit with different shorts: 1, 3, 7, 11, 15, 25, 120 and 100 000 shots, respectively.

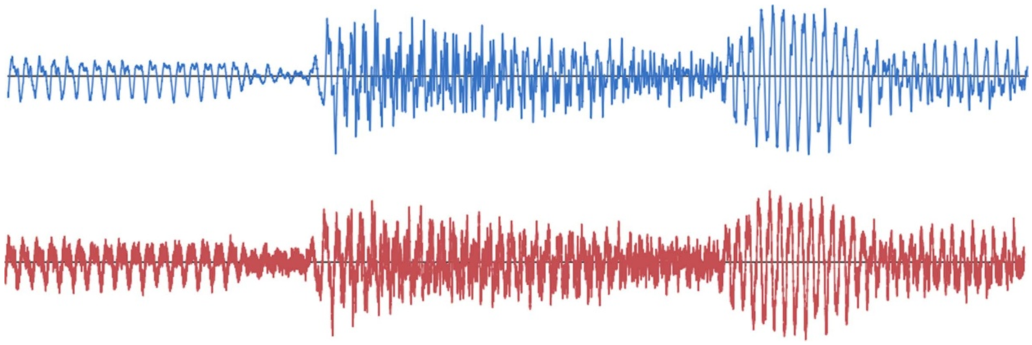

Figure 50. Quantum dithering effect applied on a 15 qubit QPAM audio.

through music developed by James Weaver at IBM Quantum, could be classed as approach (c) [120].

Another example of approaches (a) and (b) comes to mind. In 2015 two pianists, Sonja Lončar and Andrija Pavlović, collaborated with Dragan Novković at the School of Electrical and Computer Engineering of Applied Studies, Belgrade, and Vlatko Vedral, from the Department of Physics in the University of Oxford, to develop musical compositions inspired by quantum mechanics [68, 94]. Of particular interest, is a piece entitled *Bose–Einstein Condensate*. As its title suggests, it explores a physical phenomenon called Bose–Einstein Condensate (BEC) [3]. They developed methods to synthesise sounds using data from BEC experiments and physical simulations. Then, Lončar and Pavlović uploaded those sounds into a sampler and performed with them alongside two pianos.

Notwithstanding the importance of approaches (a) and (b) to pave the way for the dawn of quantum computer music, they do not make use of a *quantum computer* per se. By definition, they would not fall within the domain of Quantum Computer Music. However, what if the model is programmed in a virtual quantum computer running on an ordinary digital machine? An example of this is the work of Karl Jansen and his team at Desy, Germany. They developed a system that generates sounds from a Variational Quantum Eigensolver (QVE) model using a quantum computer simulator running on an ordinary laptop [19]. In principle, their model can run on quantum hardware just as well. Therefore, I would say that this is a very fine piece of Quantum Computer Music research. In truth, this is common practice in research circles these days: a proof-of-concept is developed and tested on a simulator first, and then it is scaled up for real quantum hardware.

Much work into developing quantum AI for music is underway. This does not come as a surprise as it draws from burgeoning activity in quantum machine learning research more generally [100, 117]. For instance, in addition to the works discussed in sections 6.1 and 7, I recently collaborated with Bob

Coecke and his team at Quantinuum to explore their Quantum Natural Language Processing (QNLP) technology in music.

Music and natural language share many properties [55, 62]. Research into QNLP is developing computational linguistic models that are deemed more suitable for quantum computers than classical ones [20]. To begin with, we implemented a quantum machine learning classifier. It was able to classify a corpus of carefully annotated music into two classes: rhythmic or melodic. Then, we built a proof-of-concept generative music system, named *Quanthoven*, which called upon the classifier to generate music. *Quanthoven* recombined snippets extracted from the music in the original training corpus to produce new rhythmic and melodic pieces [80]. A few examples were released on SoundClick [71].

Last but not least, there have been some attempts at combining quantum theory with music theory. For instance, chords are thought of as musical notes in 'superposition'. A few scholars have proposed the use of quantum terminology to express musical concepts. A notable work on this front was developed by Volkmar Putz and Karl Svozil: they proposed a theory of music based on quantum mechanical concepts of superposition and entanglement [96]. A more recent initiative suggested combining principles of quantum computing with musical harmony [30]. These works are conjectural. But they are inspirational. They may prove handy for developing representations of music for applications in (c), (d) and (e).

Back in 2017, when I heard in the news that qubits were being implemented using superconducting technology and Josephson junctions, the *Il mondo della luna* opera immediately came to mind. It looked as though a few lunatics were hoaxing potential bona fide investors to fund new start-up companies. Quantum computers cannot possibly exist, but theoretically, I thought. But no! There were no fake caricatures over the cloud. The qubits were real.

Quantum computing technology is developing at a fast pace and its impact on the music industry is inevitable. As I mentioned in the introduction, historically, the emergence of new musical instruments has always led to new kinds of music, which were inconceivable until then. Will there be new kinds of music in the future that would not be possible without quantum computing music technologies? It is too early to speculate. The journey has just begun. We will only know after we develop and try such exciting new technologies. The reader is kindly invited to jump on the bandwagon.

Data availability statement

No new data were created or analysed in this study.

Acknowledgments

This paper was written in the context of the project 'Quantum Computing Applications in Music and Art: Methods, Systems and Demonstrative Musical Compositions', funded by Quantinuum. I am deeply thankful to Ilyas Khan, the founder of Quantinuum, for his unwavering support and belief in the potential of quantum computing for music and creativity.

ORCID iD

Eduardo Reck Miranda https://orcid.org/0000-0002-8306-9585

References

[1] Adamatzky A 2021 *Handbook of Unconventional Computing Vol. 1: Theory* (World Scientific)
[2] Adamatzky A 2021 *Handbook of Unconventional Computing. Vol. 2: Implementations* (World Scientific)
[3] Andrews M R, Kurn D M, Miesner H-J, Durfee D S, Townsend C G, Inouye S and Ketterle W 1997 Propagation of Sound in a Bose-Einstein Condensate *Phys. Rev. Lett.* **79** 553–6
[4] Arrighi P and Grattage J 2010 Partitioned quantum cellular automata are intrinsically universal (arXiv:1010.2335)
[5] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10
[6] Arya A, Botelho L, Canete F, Kapadia D and Salehi O 2022 Music composition using quantum annealing *Quantum Computer Music: Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)
[7] Bernhardt C 2020 *Quantum Computing for Everyone* (The MIT Press)
[8] Blatter G, Geshkenbein V B, Fauchere A L, Feigelman M V and Ioffe L B 2001 Quantum computing with superconducting phase qubits *Physica* C **352** 105–9
[9] Bluefors n.d. Dilution refrigerator systems for quantum technology and research (available at: https://bluefors.com/home) (Accessed 14 December 2021)
[10] Boulez P Répons (Original 1981 version, world premiere) (available at: www.youtube.com/watch?v = lX2YDLQlLbI) (Accessed 9 January 2024)
[11] Born M and Fock V A 1928 Beweis des Adiabatensatzes *Z. Phys.* A **51** 165–80
[12] Braginsky V B and Khalili F A 1992 *Quantum Measurement* (Cambridge University Press)
[13] Broughton S A and Bryan K 2008 *Discrete Fourier Analysis and Wavelets: Applications to Signal and Image Processing* (Wiley)
[14] Burkhart L D 2020 Error-detected networking for 3D circuit quantum electrodynamics *PhD Thesis* Yale University, USA
[15] Burks A W, Goldstine H H and von Neumann J 1982 Preliminary discussion of the logical design of an electronic computing instrument *The Origins of Digital Computers* ed B Randell (Springer) pp 399–413
[16] Burks A W ed 1971 *Essays on Cellular Automata* (University of Illinois Press)
[17] Chowning J and Bristow D 1986 *FM Theory and Applications: by Musicians for Musicians* (Yamaha Music Foundation)
[18] Chuharski J M 2022 Adiabatic quantum computing and applications to music *Quantum Computer Music: Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)
[19] Clemente G, Crippa A, Jansen K and Tüysüz C 2022 New directions in quantum music: concepts for a quantum keyboard and the sound of the ising model *Quantum Computer Music: Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)
[20] Coecke B 2021 Compositionality as we see it, everywhere around us (arXiv:2110.05327v2)
[21] 5000 Works in Bach Style David Cope Website (available at: http://artsites.ucsc.edu/faculty/cope/5000.html)

[22] Cope D 1996 *Experiments in Musical Intelligence* (A-R Edition)
[23] Crosson E J and Lidar D A 2021 Prospects for quantum enhancement with diabatic quantum annealing *Nat. Rev. Phys.* **3** 466–89
[24] Dargan J 2023 What types of quantum computers exist in 2024? *The Quantum Insider* (available at: https://thequantuminsider.com/2023/06/06/types-of-quantum-computers/) (Accessed 6 June 2023)
[25] Datseris G and Fleiscchmann R 2019 Husimi function for electrons moving in magnetic fields (arXiv:1912.04622)
[26] Davis M H A 1993 *Markov Models and Optimization* (Routledge)
[27] Devoret M H, Wallraff A and Martinis J M 2008 Superconducting qubits: a short review (available at: https://arxiv.org/pdf/cond-mat/0411174.pdf)
[28] Devoret M H and Schoelkopf R J 2013 Superconducting circuits for quantum information: an outlook *Science* **339** 1169–74
[29] Di Giugno J, Kott J and Gerzo A 1981 Progress report on the 4X machine and its use *Proc. Int. Computer Music Conf.* (North Texas State University) (available at: https://quod.lib.umich.edu/i/icmc/bbp2372.1981.029/1/–progress-report-on-the-4x-machine-and-its-use?page = root;size = 150;view = pdf)
[30] Dobrian C and Hamido O C 2024 Intro to quantum harmony: chords in superposition (arXiv:2404.13140)
[31] Dodge T and Jerse T A 1985 *Computer Music: Synthesis, Composition and Performance* (Schirmer Books)
[32] Doornbusch P 2004 Computer music journal an overview of CSIRAC *Comput. Music J.* **28** 10–25
[33] D-Wave 2021 D-wave QPU architecture: topologies (available at: docs.dwavesys.com/docs/latest/c_gs_4.html) (Accessed 5 Febuary 2024)
[34] Easttom C 2021 *Quantum Computing Fundamentals* (Addison-Wesley)
[35] Enciclopedia Britannica Equal temperament (available at: www.britannica.com/art/equal-temperament) (Accessed 13 March 2024)
[36] Epstein J M and Axtell R 1996 *Growing Artificial Societies: Social Sciences From the Bottom Up* (The MIT Press)
[37] Ermentrout G B and Edelstein-Keshet L 1993 Cellular automata approaches to biological modeling *J. Theor. Biol.* **160** 97–133
[38] Farrelly T 2020 A review of quantum cellular automata *Quantum* **4** 368
[39] Filella N B 2018 Data sonification in creative practice *PhD Thesis* University of Plymouth
[40] Grant E K and Humble T S 2020 Adiabatic quantum computing and quantum annealing *Physics* **2020** 32
[41] Graupe D 2016 *Deep Learning Neural Networks* (World Scientific)
[42] Grover L K 1997 Quantum mechanics helps in searching for a needle in a haystack *Phys. Rev. Lett.* **79** 325–8
[43] Hernandez-Olivan C and Beltran J R 2021 Music composition with deep learning: a review (arXiv:2108.12290)
[44] Hilgevoord J 2016 The uncertainty principle *Stanford Encyclopedia of Philosophy*
[45] Hill E, Chertson J, Goldfarb S and Paradiso J A 2016 ATLAS data sonification: a new interface for musical expression and public interaction *Proc. 38 Int. Conf. on High Energy Physics* (available at: https://cds.cern.ch/record/2227055/files/ATL-OREACH-PROC-2016-005.pdf)
[46] Hiller L and Isaacson L M 1959 *Experimental Music: Composition With an Electronic Computer* (McGraw-Hill)
[47] Hoffmann P 2002 Towards an automated art: algorithmic processes in Xenakis' compositions *Contemp. Music Rev.* **21** 121–31
[48] Hogeweg P 1988 Cellular automata as a paradigm for ecological modeling *Appl. Math. Comput.* **27** 81–100
[49] Hsu K J and Hsu A J 1990 Fractal geometry of music *Proc. Natl Acad. Sci.* **87** 938–41
[50] ICFO 2023 *Quantum Sounds Symp. (Barcelona, Spain, 6 June–8 June 2023 Castelldefels)* (available at: www.icfo.eu/event/3284/quantum-sounds-symposium-/)
[51] ILLIAC Suite Online illinois distributed museum (available at: https://distributedmuseum.illinois.edu/exhibit/illiac-suite/) (Accessed 9 January 2024)
[52] Inokuchi S and Mizoguchi Y 2003 Generalized partitioned quantum cellular automata and quantization of classical CA (arXiv:quant-ph/0312102)
[53] Miranda E R and Jansen K 2023 *2nd Int. Symp. on Quantum Computing and Musical Creativity (Berlin, 5 October–6 October 2023)* (available at: https://indico.desy.de/event/38609/page/4475-information)
[54] Itaborai P V and Miranda E R 2022 Quantum representations of sound: from mechanical waves to quantum circuits *Quantum Computer Music: Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)
[55] Jäncke L 2012 The relationship between music and language *Front. Psychol.* **3** 123
[56] Kelemen J and Sosik P 2001 *Advances in Artificial Life - Lecture Notes in Artificial Intelligence* (Springer)
[57] Kindermann R and Snell J L 1980 *Markov Random Fields and Their Applications* (American Mathematical Society)
[58] Kirke A and Miranda E R 2017 Experiments in sound and music quantum computing *Guide to Unconventional Computing for Music* ed E R Miranda (Springer)
[59] Kirke A 2016 Quantum computer singing performance (available at: https://alexiskirke.com/2016/07/07/d-wave-quantum-computer-performance/) (Accessed 12 March 2024)
[60] Kirke A, Miranda E, Chiaramonte A, Troisi A R, Matthias J, Fry N, McCabe C, Radtke J and Bull M 2013 Cloud chamber: a performance with real-time two-way interaction between subatomic particles and violinist *Leonardo* **46** 84–85
[61] Cloud Chamber Premiere at Peninsula arts contemporary music festival, Plymouth, UK (available at: www.youtube.com/watch?v = iGcYyZRYV78)
[62] Koelsch S 2011 Toward a neural basis of music perception - a review and updated model *Front. Psychol.* **2** 110
[63] Kowalski K and Rembielinski J 2016 The Wigner function in the relativistic quantum mechanics *Ann. Phys., NY* **375** 1–15
[64] Kuang J and Yang T 2021 Popular song composition based on deep learning and neural networks *J. Math.* **2021** 7164817
[65] Lampedusa: Behind the curtain University of Plymouth, UK (www.plymouth.ac.uk/students-and-family/arts-institute-public-programme/lampedusa-behind-the-curtain) (Accessed 9 January 2024)
[66] Lampedusa - Opera in 3 Acts University of Plymouth, UK (https://vimeo.com/331262882) (Accessed 9 January 2024)
[67] Lewenstein M, Ciappina M F , Pisanty E, Rivera-Dean J, Lamprou T and Tzallas P 2021 Generation of optical Schrödinger cat states in intense laser-matter interactions *Nat. Phys.* **17** 1104–8
[68] Lončar S and Pavlović A 2018 Hybrid duo *Muzikologija* **24** 111–21
[69] Lohner H 1986 The UPIC system: a user's report *Comput. Music J.* **10** 42–49
[70] Loy G 2011 *Musimathics: The Mathematical Foundations of Music* (The MIT Press)
[71] Ludovico Quanthoven Album on soundclick (available at: www.soundclick.com/ludovicoquanthoven) (Accessed 5 March 2024)

[72] Miranda E R 2019 Lampedusa Opera, Libretto (available at: https://neuromusic.soc.plymouth.ac.uk/papers/Lampedusa_Programme.pdf) (Accessed 9 January 2024)
[73] Manabrea L F 1843 *Sketch of the Analytical Engine Invented by Charles Babbage* Translated by Ada Lovelace. R. and J. E Taylor
[74] Mawhinney S and Boulez P 2001 Composer in Interview: Pierre Boulez *Tempo* **216** 2–5
[75] McQuillan I and Seki S (eds) 2019 Unconventional computation and natural computation *18th Int. Conf., UCNC 2019* (Lecture Notes in Computer Science (LNCS), (Springer)
[76] Miranda E R and Shaji H 2023 Generative music with partitioned quantum cellular automata *Appl. Sci.* **13** 2401
[77] Miranda E R, Thomas P and Itaboraí P V 2023 Q1Synth: a quantum computer musical instrument *Appl. Sci.* **13** 2386
[78] Miranda E R and Basak S T 2022 Quantum computer music: foundations and initial experiments *Quantum Computer Music: Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)
[79] Miranda E R and Miller-Bakewell H 2022 Cellular automata music composition: from classical to quantum *Quantum Computer Music Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)
[80] Miranda E R, Yeung R, Pearson A, Meichanetzidis K and Coecke B 2022 A quantum natural language processing approach to musical intelligence *Quantum Computer Music Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)
[81] Miranda E R (ed) 2021 *Handbook of Artificial Intelligence for Music Foundations, Advanced Approaches and Developments for Creativity* (Springer International Publishing)
[82] Miranda E R 2020 Creative quantum computing: inverse FFT, sound synthesis, adaptive sequencing and musical composition (arXiv:2005.05832)
[83] Miranda E R 2019 Lampedusa, Act 1 (available at: https://vimeo.com/332486103) (Accessed 9 January 2024)
[84] Miranda E R 2019 Lampedusa, Act 1 (available at: https://vimeo.com/332486858) (Accessed 9 January 2024)
[85] Miranda E R 2019 Lampedusa, Act 1 (available at: https://vimeo.com/335205464) (Accessed 9 January 2024)
[86] Miranda E R 2016 The music of particle collisions *Nat. Phys.* **12** 721
[87] Miranda E R 2011 *A-Life for Music: Music and Computer Models of Living Systems* (A-R edns)
[88] Miranda E R and Biles J A (eds) 2007 *Evolutionary Computer Music* (Springer)
[89] Miranda E R 2002 Generating source streams for extralinguistic utterances *J. Audio Eng. Soc.* **50** 165–72
[90] Miranda E R 2002 *Computer Sound Design: Synthesis Techniques and Programming* 2nd edn (Focal Press)
[91] Miranda E R 1991 Cellular Automata Music: An Interdisciplinary Project *J. New Music Res.* **22** 3–21
[92] Morten K, Schwartz M E, Braumüller J, Krantz P, Wang J, Gustavsson S and Oliver W D 2019 Superconducting qubits: current state of play *Annu. Rev. Condens. Matter Phys.* **11** 369–95
[93] Nielsen M A and Chuang I L 2011 *Quantum Computation and Quantum Information* 10th Anniversary edn (Cambridge University Press)
[94] Novković D, Peljević M and Malinović M 2018 Synthesis and analysis of sounds developed from the Bose-Einstein condensate: theory and experimental results *Muzikologija* **24** 95–109
[95] Preston K and McDuff M J B 1984 *Modern Cellular Automata: Theory and Applications* (Springer International Publishing)
[96] Putz V and Svozil K 2022 Quantum music, quantum arts and their perception *Quantum Computing in the Arts and Humanities* ed E R Miranda (Springer)
[97] Qiskit Documentation (available at: www.ibm.com/quantum/qiskit) (Accessed 5 March 2024)
[98] Quantum Sounds (available at: www.youtube.com/watch?v = aOVW59VhaNk&t = 2s) (Accessed 3 January 2024)
[99] Qubism 2023 Musical composition with quantum computer (available at: https://on.soundcloud.com/GPAVz) (Accessed 27 January 2024)
[100] Raj P, Kumar A, Dubey A K, Bhatia S and Manoj S O (eds) 2023 *Quantum Computing and Artificial Intelligence* (De Gruyter)
[101] Roth T E, Ma R and Chew W C 2021 An introduction to the transmon qubit for electromagnetic engineers (arXiv:2106.11352)
[102] Sachdev S 2011 *Quantum Phase Transitions* (Cambridge University Press)
[103] Schleich W P 2001 *Quantum Optics in Phase Space* (Wiley)
[104] Schuld M and Petruccione F 2021 *Machine Learning With Quantum Computers* (Springer)
[105] Schutte U 2020 *Kraftwerk: Future Music From Germany* (Penguin)
[106] Serniak K 2019 Nonequilibrium quasiparticles in superconducting qubits *PhD Thesis* Yale University, USA
[107] Shoupu W 2009 Atomic fock states and quantum computing *PhD Thesis* University of Texas at Austin, USA
[108] Singh S P 2019 The ising model: brief introduction and its application *Solid State Physics - Metastable, Spintronics Materials and Mechanics of Deformable Bodies - Recent Progress* ed S Sivasankaran, P K Nayak and E Günay (IntechOpen Limited)
[109] Sounding Qubits Video documentary. The Goethe-Institut (available at: www.goethe.de/prj/lqs/en/eve/sou.html) (Accessed 9 January 2024)
[110] Spall J C 1998 Implementation of the simultaneous perturbation algorithm for stochastic optimization *IEEE Trans. Aerosp. Electron. Syst.* **34** 817–23
[111] Spinnings Musical composition for Q1Synth trio E. R. Miranda, P. Thomas and P. Itabora'i (available at: www.soundclick.com/music/songInfo.cfm?songID = 14529239) (Accessed 9 January 2024) SoundClick
[112] Strategist Q 2020 What is quantum annealing and how does it differ from gate-based quantum computers? *Quantum Zeitgeist* (available at: https://quantumzeitgeist.com) (Accessed 8 June 2022)
[113] Tafuri F Ed 2019 *Fundamentals and Frontiers of the Josephson Effect* (Springer)
[114] Topel S, Serniak K, Burkhard L and Carle F 2022 Superconducting qubits as musical synthesizers for live performance *Quantum Computer Music: Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)
[115] Toumi A, Yeung R and de Felice G 2021 Diagrammatic differentiation for quantum machine learning *Electron. Proc.Theor. Comput. Sci.* **343** 132–44
[116] von Neumann J 2018 *Mathematical Foundations of Quantum Mechanics* ed N Wheeler (Princeton University Press)
[117] Viggiano G and Brin D 2023 *Convergence: Artificial Intelligence and Quantum Computing* (Willey)
[118] Vool U and Devoret M 2017 Introduction to quantum electromagnetic circuits *Int. J. Circuit Theory Appl.* **45** 897–934
[119] The music played by CSIRAC (available at: https://cis.unimelb.edu.au/about/csirac/music/music-played) (Accessed 9 January 2024)
[120] Weaver J 2022 Quantum music playground tutorial *Quantum Computer Music: Foundations, Methods and Advanced Concepts* ed E R Miranda (Springer)

[121] Woods W A 1970 Transition network grammars for natural language analysis *Commun. ACM* **13** 591–606
[122] Worral D 1996 Studies in metamusical methods for sound image and composition *Organised Sound* **1** 183–94
[123] Xenakis I 1971 *Formalized Music* (Indiana University Press)
[124] Yamada R 2022 Interpreting quantum randomness Phonos: Barcelona, Spain
[125] Yamada R 2022 String Quartet. Recording of a musical composition composed with the Wigner function (available at: https://shorturl.at/abjE7) (Accessed 16 January 2024)
[126] Yamada R, Grandi S, Muñoz-Gil G, Barbiero L, Aloy A and Lewenstein M 2021 Applications of quantum randomness: from rabi oscillations to fourier axis controlling the musical timbre *Int. J. Music Sci. Technol. Art* **3** 17–25
[127] Zeno Musical composition with quantum computing (available at: www.youtube.com/watch?v = zeDM4tspdh4&t = 3s) (Accessed 7 January 2024)
[128] Zhao Z and Wang C Ed 2019 *Cryogenic Engineering and Technologies: Principles and Applications of Cryogen-Free Systems* (CRC Press)