This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

This is a non-edited pre-publication version of a chapter to appear in the book "*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024. DOI:10.1142/14025

Developing a Framework for Sonifying Variational Quantum Algorithms: Implications for Music Composition

Paulo Vitor Itaboraí[1,2], Peter Thomas [3], Arianna Crippa[1,4], Karl Jansen[1,2], Tim Schwägerl[1,4], María Aguado Yáñez[1,5]

[1] *Center for Quantum Technologies and Applications – Deutsches Elektronen-Synchrotron DESY*
Platanenallee 6 – 15738 – Zeuthen – Germany
[2] *Computation-based Science and Technology Research Center – The Cyprus Institute*
20 Konstantinou Kavafi Street – 2121 –Aglantzia – Cyprus
[3] *Interdisciplinary Centre for Computer Music Research – University of Plymouth*
Drake Circus - PL48AA - Plymouth - United Kingdom
[4] *Institüt für Physik – Humboldt-Universität zu Berlin*
Newtonstr. 15 – 12489 Berlin –Germany
[5] *Music Technology Group – Pompeu Fabra University*
Roc Boronat, 138 – 08018 – Barcelona – Spain

Abstract This chapter examines the Variational Quantum Harmonizer, a software tool and musical interface that focuses on the problem of sonification of the minimization steps of Variational Quantum Algorithms (VQA), used for simulating properties of quantum systems and optimization problems assisted by quantum hardware. Particularly, it details the sonification of Quadratic Unconstrained Binary Optimization (QUBO) problems using VQA. A flexible design enables its future applications both as a sonification tool for auditory displays in scientific investigation, and as a hybrid quantum-digital musical instrument for artistic endeavours. In turn, sonification can help researchers understand complex systems better and can serve for the training of quantum physics and quantum computing. The VQH structure, including its software implementation, control mechanisms, and sonification mappings are detailed. Moreover, it guides the design of QUBO cost functions in VQH as a music compositional object. The discus-

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

sion is extended to the implications of applying quantum-assisted simulation in quantum-computer aided composition and live-coding performances. An artistic output is showcased by the piece *Hexagonal Chambers* (Thomas & Itaboraí, 2023).

1.1 Introduction

The evolution of musical instruments in the 1900's and early 2000's has included electronic and digital technologies in its design and manufacturing, which has greatly impacted how music is composed and played. Modern programming languages such as SuperCollider [McCartney (2002)], Pure Data [Puckette *et al.* (1996)], and Max/MSP [Zicarelli (1998)] have enabled a flexible creation of innovative musical interfaces, allowing performers to dynamically build, modify and interact with new instruments, even while performing on stage [Roads (1996)]. This integration of digital and music technologies has introduced new forms of musical expression [NIME Proceedings]. Arguably, a similar effect can occur with quantum technologies and music.

1.1.1 *Motivation*

The use of sound for representing data is a known strategy for scientific dissemination and the comprehension of time-dependent and high-dimensional datasets. To name some examples, sonification has been used for describing astronimical data, such as the results of the acclaimed experiment on the detection of graviational waves [Zanella *et al.* (2022)]. In addition, researchers have used sonification as a tool during the design of multiscale molecular simulations in material physics [Rodrigues Miranda (2019)]. Furthermore, sensing instruments have also been traditionally implemented for identifying 'invisible' properties of our surroundings through sound, such as detection of ionizing radiation with a Geiger counter [Curtiss (1950)] or more emblematically for music, capacitive coupling of electic fields with a Theremin [Glinsky (2000)]. A more recent example shows that sonification can also be used as real-time auditory feedback in medical applications [Corredera and Tanaka (2023)].

Lastly, yet importantly, the well-known impact of sonification in music and sound art should not be overlooked [Straebel (2010)][Xenakis (1992)][Cage (1973)]. Integrating innovative technologies in electronic music instruments could create new possibilities for creating and listening to

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

music [Meyer-Eppler (1958)]. As stated by the instrument inventor Anton Walter Smetak: *"A new world requires new people and new music - and to that end, different musical instruments"* [1]

With the emerging field of quantum computing, a novel technology could be integrated to offer the possibility of going in different directions of musical expression at many levels of the artistic process.

In conclusion, creating auditory displays for comprehending, disseminating, and assisting research in quantum technologies is a problem that should be explored in depth by Quantum Computer Music academics and eventually the industry.

In another note, circuit-based quantum computing with qubits deals with the control and interaction of two-level quantum systems, usually coupled together to achieve complex behavior and physical phenomena, which could be difficult for classical machines to simulate. Likewise, a similar analogy could be made when considering acoustic musical instruments. According to Fletcher and Rossing's definition:

"In most instruments, sound production depends upon the collective behavior of several vibrators, which may be weakly or strongly coupled together. This coupling, along with nonlinear feedback, may cause the instrument as a whole to behave as a complex vibrating system, even though the individual elements are relatively simple vibrators." [Fletcher and Rossing (1998)]

This analogy provides a motivation layer for the interpretation of qubits as individual elements, which are coupled together in order to compose a complex structure that could be used to generate organized sound [Landy (2024)].

1.1.2 *Previous work*

1.1.2.1 *Interfacing Quantum Computing with Musical Interfaces*

There are a relatively small number of early studies on the topic of the sonification of quantum algorithms, either for educational or artistic purposes. In general, the literature on Quantum Computer Music is very recent. At the time of writing, the most widely adopted references can be found in two books [Miranda (2022b)][Miranda (2022)], and conference proceedings [Hamido and Itaboraí (2023b)], among other works of the same period. By surveying the works related to applications of sonification and their

[1] Quote translated from Portuguese: "Um novo mundo requer homens novos e uma musica nova - e para isso, instrumentos musicais diferentes" [Scarassatti (2001), p.xi]

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

software implementations, it is possible to verify how researchers have addressed the problem of integrating quantum computing frameworks with music programming. The approaches can be divided into two main categories: Music-Oriented and Quantum-Oriented.

A Music-Oriented Approach In frameworks centered in music programming languages, a common challenge lies in the need of a platform for a) generating quantum circuits and b) running simulations or connecting to quantum hardware. In the literature, it is possible to find implementations of microqiskit in Max/MSP [Hamido (2022)] and statevector simulators in Ableton Live [Weaver (2022)]. Additionally, there is a software application that listens to quantum circuit instructions sent via Open Sound Control (OSC) Protocol [Wright and Freed (1997)] for running circuits in quantum systems [Hamido and Itaboraí (2023a)]. The last mentioned application has been used to drive the quantum backend for the sonification of a Bloch sphere, implemented in a web-based audio application [Miranda *et al.* (2023)]. Due to the current limitations of the implementations of these simulators and qasm code generators, the applications are currently restricted to relatively simple quantum circuits and manipulations.

A Quantum-Oriented Approach Conversely, it is more common to find Quantum-Centered applications. This means that the main implementation is (typically) realised in Python, and then the challenge changes to finding separate modules and software frameworks to generate sound and music reliably. Researchers have utilized this approach to use more complex quantum circuits in sonification problems and music composition. To list some examples from the literature, quantum algorithms have been used to generate or trigger MIDI notes [Miranda and Shaji (2023)][Yáñez (2023)], musical scores [Miranda and Siegelwax (2022)], and even store and retrieve audio files [Itaboraí and Miranda (2022)][Itaboraí (2023)]. Particularly, the latter reference uses Python as a SuperCollider client [Jones (2020)], by implementing formatted OSC messages to instantiate synthesizers and control the audio engine. This is the approach chosen for the implementation presented in this work.

1.1.2.2 *Sonifying the Ising Model*

Considering the previous work published as a chapter of the preceding Quantum Computer Music book [Clemente *et al.* (2022b)], in Section 3,

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

entitled *"The sound of the Ising Model"*, the authors utilize a Variational Quantum Algorithm (VQA) such as the one described in section 1.2 to extract observables[2] from the Ising Model and create a simple sonification mapping, where the values represent frequencies of a sound spectrum mapped to a perceptual listening range.

They propose an additive synthesis approach for sonifying the Ising Model using two different control signals. The first method maps the physical properties of the Ising Model - Energy eigenvalues and Magnetizations - as frequencies and amplitudes, respectively. Furthermore, the spin coupling h is used as a time variable, that evolves the system through a phase transition. In other words, in their example, for each timestamp t_n there were 16 eigenvalues E_n and 16 eigenstates $|\psi_n\rangle$ (from where the magnetizations M_n were computed) from a VQA run with a specified coupling $h(n)$.

More compellingly (for the purposes of this chapter), in the map proposed in Sec. 3.1.2 (*"Use the callback results"*), Instead of using only the output of the algorithm, where in principle the observables could have been computed by a classical algorithm they *"unfold"* the time t_n and adjust their lenses to the intermediary steps of the minimization process, and compute the eigenvalues E_n at every iteration. Hence, they infer the sonification of the VQA itself.

In sequence of [Clemente *et al.* (2022b)], [Itaboraí *et al.* (2023a)] expands on the suggestion proposed above, and includes the sonification of the quantum states $|\psi_n\rangle$ at intermediate stages of a VQA optimization, which becomes a primary point of interest on the sonification map. By prototyping a system that realizes the proposed sonification for a Variational Quantum Eigensolver (VQE) algorithm and produces audible output, the Variational Quantum Harmonizer (VQH) was conceived.

In [Itaboraí *et al.* (2023a)], there is a focus on introducing the topic, obtaining musical intuition, and also a discussion on an artistic output in the form of a music study, named *"Dependent Origination"*.

The aim of this work is to provide an overview of the second version of the VQH implementation, as well as broaden the concepts introduced, with the intention of allowing the reader to incorporate and use the VQH implementation in their own projects. Furthermore, there is a brief discussion on the use of VQH for creating music compositions as well as implications and intricacies of the artistic process for creating the music piece *"Hexagonal*

[2]i.e., any physical quantity that can be measured, such as energy, momentum, spin, etc.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

Chambers"(See Section 1.7.1).

1.2 Variational Quantum Harmonizer

The *Variational Quantum Harmonizer* is a hybrid quantum-classical sonification-based musical instrument, implemented in Python and using Qiskit [Aleksandrowicz *et al.* (2019)] for running quantum algorithms and generating data that is sonified using Supercollider [McCartney (2002)] as its main sound engine system. In other words, the VQH is a sonification tool for mapping Variational Quantum Algorithms into sound.

Variational Quantum Eigensolver The Variational Quantum Eigensolver (VQE)[Peruzzo *et al.* (2014)], (Fig.1.1) is a hybrid algorithm for simulations and optimisation problems, that integrates classical and quantum computing paradigms to determine the ground state and some excited states of a given physical system. The algorithm is based on the variational method of quantum mechanics, i.e. finding the lowest energy state via an approximation strategy. In order to apply this technique, a quantum circuit is defined with a set of parametrized and entangling gates and with an initial guess for the parameters θ. By exploiting this initial configuration, the quantum processor calculates the expectation value of an observable of the system, typically the Hamiltonian, getting the value of the cost function $E(\theta) = \langle\psi(\theta)| H |\psi(\theta)\rangle$. Then, a classical optimizer iteratively refines the initial parameters, with a new set of θ. This procedure is repeated until the optimization reaches convergence and a minimum of the cost function is found[3].

In essential terms, the VQH has two distinct levels of approach to its control interface and subsequent workflows. In a higher-level routine, the user can load datasets containing formatted results of a VQE experiment and map the data into various sonification mappings to listen to the results. Alternatively, the user is encouraged to design their own experiments from the ground up, using the interface, with the goal of obtaining sonic intuition on the way a problem of interest works, and most importantly how the VQE is performing to achieve that result. This represents the broader methodology of the VQH application as a whole, towards a system that can be used both to assist in the analysis and interpretation of quantum

[3]Notice, however, that this algorithm can also get trapped in local minima or constant plateau landscapes

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

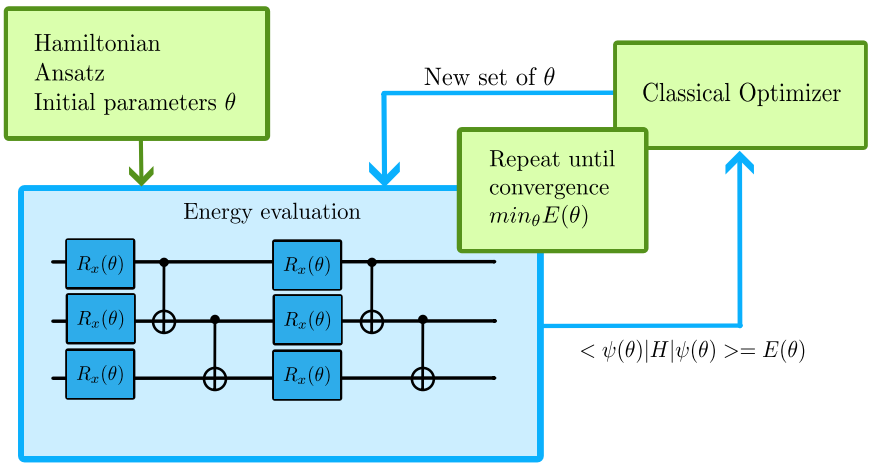


Fig. 1.1: Schematic illustration of the Variational Quantum Eigensolver (VQE) algorithm.

simulations or to compose music.

1.2.1 *The VQH prototype and overview*

For the first part of this project, an implementation that realizes the proposed pipeline (a real-time sonification tool for VQE experiments) was reached for a simple problem and also with a direct encoding as a quantum circuit and a clear mapping into sound.

In more detail, the user controls and designs parameters of a $n \times n$ square matrix that defines a Quadratic Unconstrained Binary Optimization (QUBO) problem [4].

QUBO Quadratic unconstrained binary optimization (QUBO)[Date *et al.* (2021)] is a combinatorial optimization problem, i.e. the problem of finding the best solution from all feasible solutions. It is characterised by a cost function (Eq.1.1), comprising both linear and quadratic terms. Each variable n_i uniquely contributes to the system with its linear coefficients a_i, while interacting with the other agents through the quadratic coefficients b_{ij}. Effectively solving this problem means identifying (or approximating)

[4] In practice, since the QUBO matrix is symmetric, only the upper triangle elements need to be defined

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

a configuration that minimises the cost function.

$$Q(a,b,n) = \sum_{i}^{N} a_i n_i + \sum_{i,j} b_{ij} n_i n_j \quad \text{with } n \in \{0,1\}^N \ . \tag{1.1}$$

Then, the problem is transformed into an Ising Hamiltonian 'H', by mapping the binary variables n_i to Pauli Z operators (Eqs. 1.2-1.5). This characterizes an energy landscape from where the ground state can be approached by means of a VQE-based optimization. Subsequently, at each iteration of the optimization process, the current configuration is extracted and sampled by a callback function to obtain a probability distribution that describes the current state of the system, as well as an intermediary-stage computation of the energy expectation value for the designed Hamiltonian.

$$n_i \Rightarrow \sigma_i^Z \left|n_i\right\rangle = (1-2n_i)\left|n_i\right\rangle \ , \tag{1.2}$$

$$b_{ij} \Rightarrow \beta_{ij} = b_{ij} \ , \tag{1.3}$$

$$a_i \Rightarrow \alpha_i \ = -2a_i - \sum_{j \neq i} b_{ij} \ , \tag{1.4}$$

$$H(\alpha,\beta,\sigma) = \sum_{i}^{N} \alpha_i \sigma_i^Z + \sum_{i,j} \beta_{ij} \sigma_i^Z \sigma_j^Z \ . \tag{1.5}$$

Having obtained a sample from the quantum state's probability distribution and corresponding expectation value, the first stage of sonification mapping is applied, where a decoding procedure - which will be referred to in this text as *'Basis Protocol'* - is applied to represent a qubit sequence as a group of n sounding ($|1\rangle$) or non-sounding ($|0\rangle$) notes (see Section 1.2.1.1). This protocol is carried out by obtaining the *marginal distribution* of each individual qubit, aggregating the amplitudes of the basis states that contribute for each respective note to play. As a result, a set of n relative amplitudes is achieved. By collecting these sets over many iterations, it is possible to build n streams of data to control, for instance, the amplitude of n sinusoidal oscillators with distinct frequencies (see Sec. 1.21).

Marginal Distribution and the Basis Protocol This protocol defines the main sonification strategy used in this text. To establish a comparative relation: in data visualization, the content to be seen needs to be grouped and arranged into a mesh, geometry, or grid before a data attribute is mapped into color and rendered as an image. In the Basis Protocol, the

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

'sonic geometry' is the qubit marginal distribution of the sampled state. The marginalization process discards the information related to conditional probability in joint distributions, as the one of a multi-qubit system. In other words, it is the probability of a qubit being in a state $|1\rangle$ or $|0\rangle$, independently of the states of other qubits. To illustrate, consider the 3-qubit example below (Eqs. 1.6-1.9).

$$|\psi_{abc}\rangle = \sqrt{\frac{3}{4}}\,|100\rangle + \sqrt{\frac{3}{16}}\,|011\rangle + \sqrt{\frac{1}{16}}\,|101\rangle \ , \tag{1.6}$$

$$P_{marg}(q_a = |1\rangle) = 3/4 + 1/16 = 13/16 \ , \tag{1.7}$$

$$P_{marg}(q_b = |1\rangle) = 3/16 \ , \tag{1.8}$$

$$P_{marg}(q_c = |1\rangle) = 3/16 + 1/16 = 1/4 \ . \tag{1.9}$$

1.2.1.1 *Artistic Intuition: The Harp analogy*

To obtain musical intuition on the Basis Protocol, we introduce an analogy, using a figurative Harp as an object where the qubits are represented as strings. Consider a simple 3-qubit example.

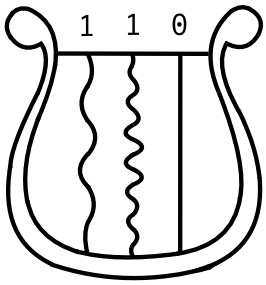


Fig. 1.2: A 3-qubit harp

In this 3-qubit Harp, the two first qubits are playing, while the third one remains silent. To configure this scenario as the ground state following the execution of the VQE algorithm, a cost function has to be designed with a QUBO coefficient matrix, so that it yields the desired outcome. There are two strategies for formulating a QUBO problem, both capable of achieving this end result. The first one involves tuning the linear coefficients a_i, assigning a value of -1 to the playing qubits and 1 to the silent one. This means that the third qubit's linear coefficient is being *penalized*, since it increases the cost function, whereas the algorithm aims at minimizing it.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

This first QUBO matrix would be then implemented as follows in Eq.1.10 [5],

$$Q(n) = -n_1 - n_2 + n_3 = \begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}. \tag{1.10}$$

The second approach builds the QUBO problem tuning the quadratic coefficients b_{ij} instead. By penalizing the interaction between the silent qubit with respect to the playing ones, and favouring the interaction between the first two, as in Eq.1.11, the desired ground state can be reached.

$$Q(n) = \begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix} \begin{pmatrix} 0 & -1 & 1 \\ -1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}. \tag{1.11}$$

Another way of visualizing the quadratic coefficients is by looking into the Ising Hamiltonian that is created from the QUBO coefficients (Eqs. 1.4-1.5). In this simple example, the Ising Model would characterize 3 spins. Translating the Harp analogy to the Ising Model, a silent note could be represented by a spin-down state ($|\downarrow\rangle$), whereas a playing note is encoded as a spin-up ($|\uparrow\rangle$). This allows for a physical interpretation of the quadractic coefficients, where the coupling determines the disposition of the spin pair with respect to each other. From equation 1.5, a negative β value would favor a ferromagnetic alignment ($\uparrow\uparrow$, $\downarrow\downarrow$), while a positive value benefits anti-ferromagnetic configurations ($\uparrow\downarrow$, $\downarrow\uparrow$)[6].

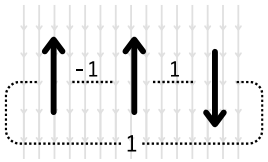


Fig. 1.3: A 3-spin Ising harp

[5] Note that the variables n can take either value 1 or 0, so that $n_i^2 = n_i$.

[6] Hence, one should be careful when designing with quadratic coefficients, since the alignment determines the direction of each spin, but not its orientation in respect to an external field. Sometimes, there is a need to introduce some linear correction factors, to ensure a desirable orientation and thus ground state. See Sec. 1.3.3.1.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

1.3 Designing Chords

Expanding from the analogy above, the first subsequent idea is to explore simple musical scales (e.g., Pentatonic, Pythagorean, Modal, Diatonic, etc..), and attempt to design cost functions in which the ground state is known, enabling us to encode, for example, a chord.

1.3.1 *Cmajor Chord*

As a starting point, let us create a 12-tone Chromatic Scale.

↑	↓	↓	↓	↑	↓	↓	↑	↓	↓	↓	↓
C	*C#*	*D*	*D#*	*E*	*F*	*F#*	*G*	*G#*	*A*	*A#*	*B*

Fig. 1.4: Spin configuration of a C Major Chord

Accordingly, 12 qubits are assigned as shown in fig 1.4. Then, the following cost function, shown in Eq. 1.12, is designed to be minimized into a *C Major* Chord using only linear coefficients for now.

$$Q(n_{Cmaj}) = -n_C - n_E - n_G + \sum_{k \notin Cmaj} n_k \quad . \tag{1.12}$$

When 1.12 is translated into a qubit-based Hamiltonian, the following set of Pauli operators is achieved (Eq. 1.13[7,8]).

$$\begin{aligned} 2\big[& - ZIIIIIIIIIII \\ & + IZIIIIIIIIII \\ & + IIZIIIIIIIII \\ & + IIIZIIIIIIII \\ & - IIIIZIIIIIII \\ & + IIIIIZIIIIII \\ & + IIIIIIZIIIII \\ & - IIIIIIIZIIII \\ & + IIIIIIIIZIII \\ & + IIIIIIIIIZII \\ & + IIIIIIIIIIZI \\ & + IIIIIIIIIIIZ \big] \end{aligned} \tag{1.13}$$

[7] To keep the same convention from the harp analogy, the qubits are being ordered in Big-endian (left-to-right) form in equation 1.13, contrary to Qiskit's Little-endian notation.

[8] Also note that the signs are swapped compared to the QUBO coefficients of Eq. 1.12, as $Z|\downarrow\rangle = |\downarrow\rangle$ and $Z|\uparrow\rangle = -|\uparrow\rangle$ (Eq. 1.2). This ensures that $|\uparrow\downarrow\downarrow\uparrow\downarrow\downarrow\uparrow\downarrow\downarrow\downarrow\downarrow\rangle$ is the ground state for Eq. 1.13

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

We use the COBYLA [Powell (1994)] optimizer and Qiskit's EfficientSU2 ansatz to perform a VQE minimization on this energy profile.

As mentioned in the section above, at each iteration, the current state is sampled. To monitor which notes are playing at that stage, the marginal distribution of the sample is computed. The progression of this marginal distribution across 150 iterations of the optimization is depicted in Fig. 1.5.

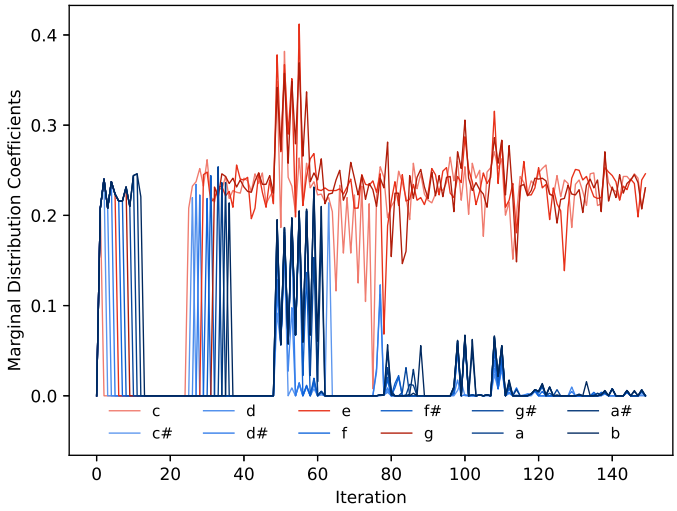


Fig. 1.5: Marginal Distribution Evolution for C_{maj}

The coefficients are then mapped into an additive synthesis with 12 oscillators (tuned according to the chromatic scale), and their respective amplitudes are controlled over time. A spectrogram of the resulting sound is shown in Figure 1.6

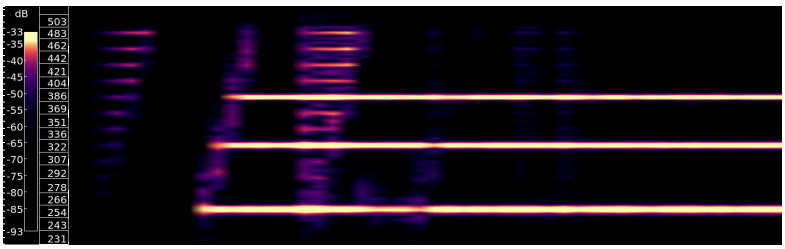


Fig. 1.6: Spectrogram of C_{maj} sonified with additive synthesis

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

One can see how the optimizer managed the minimization process and steered the configuration space toward the anticipated solution - a sounding C major.

1.3.2 *Chord Progressions*

In the example above, the experiment was designed in a way that it starts in "silence", meaning the optimization begins with all qubits in the $|0\rangle$ state. However, it is technically possible to choose any other valid state as the initial condition for the VQE, *including one that encodes a different note or chord*. Consequently, a problem could be designed in a way that an arbitrary initial chord C_1 transitions to a new chord C_2.

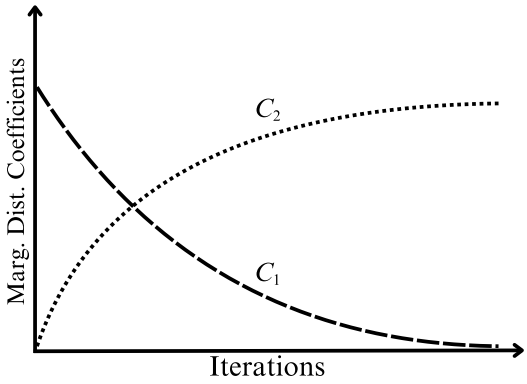


Fig. 1.7: Changing the initial condition

Expanding this idea, the next step would be to concatenate different VQE experiments by using the results of a previous run as the initial conditions for a new one, thus creating long chord progressions, timbral transitions, or more complex sonic transformations. These would then be controlled by the changes in the energy profile of the system, characterized by a time-dependent Hamiltonian (Fig. 1.8).

1.3.2.1 *Example: I-IV-V-I Progression*

Consider the following cost functions Q_{1-4} below (Eqs. 1.14-1.17).

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

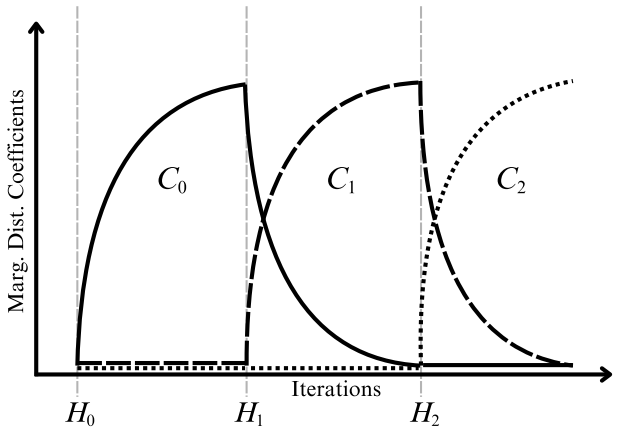


Fig. 1.8: Creating chord progressions

$$Q_1(n^I) = -n_C - n_E - n_G + \sum_{k \notin Cmaj} n_k \,, \tag{1.14}$$

$$Q_2(n^{IV}) = -n_F - n_A - n_C + \sum_{k \notin Fmaj} n_k \,, \tag{1.15}$$

$$Q_3(n^V) = -n_G - n_B - n_D + \sum_{k \notin Gmaj} n_k \,, \tag{1.16}$$

$$Q_4(n^I) = Q_1(n^I) \,. \tag{1.17}$$

Then, 4 experiments are prepared with 512 iterations, and identical setup (COBYLA, EfficientSU2). However, the initial point of Q_{n+1} is the solution of Q_n. By concatenating the iterations of all runs, we are left with a classic I-IV-V-I chord progression 1.9.

It is possible to note that, by changing the initial point from "silence" to a different state, the minimization process will necessarily have another path, which might produce the appearance of "intrusive" notes (such as the A# in the last chord), or the persistence of previous notes where a decrease in the note's amplitude may not provide significant changes in the minimization task of the current chord.

1.3.2.2 *Adiabatic Progressions*

As an additional option, it is possible to consider smoothing down the chord transition. For instance, one could consider a Hamiltonian that is a

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

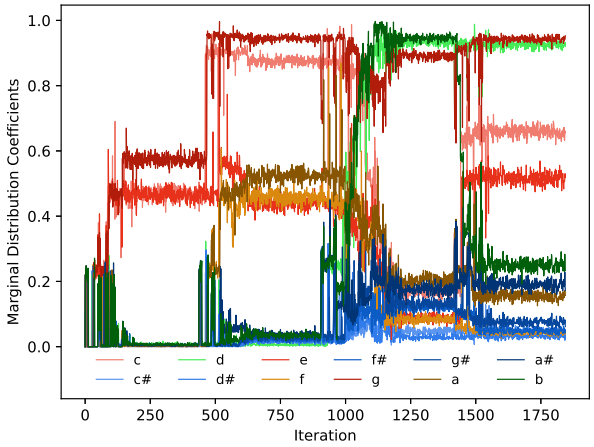


Fig. 1.9: Marginal Distribution Evolution of a I-IV-V-I Progression

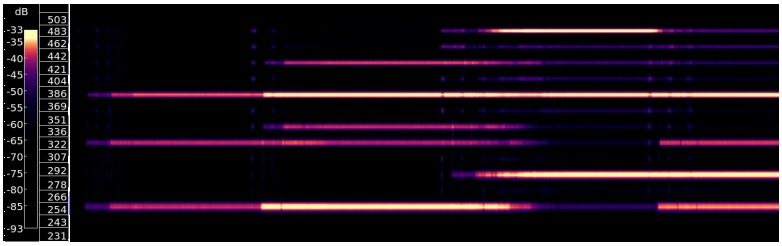


Fig. 1.10: Spectrogram of I-IV-V-I progression with additive synthesis

combination of one chord and another, and use this Hamiltonian as halfway point. Ultimately, with enough intermediary steps, the aim would be to slow the progression of the chord until it evolves *adiabatically*. In fact, this is a technique used in the literature to attempt to improve the VQE minimization, called *Adiabatically Navigated VQE* [Matsuura *et al.* (2020)]. In other words, we design a time-dependent Hamiltonian that guides the solution from $H_{initial}$ to H_{final} (Eq. 1.18).

$$H(t) = (1-t)H_{initial} + tH_{final} \ . \tag{1.18}$$

For example, one could start with a $|C_{maj}\rangle$ configuration and make an adiabatic transition to $|B7/D\#\rangle$ (Fig. 1.11)

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

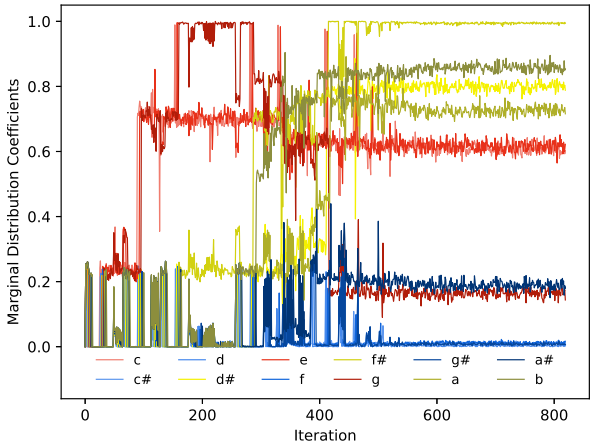


Fig. 1.11: Adiabatic Chord Progression from $|C_{maj}\rangle$ to $|B7/D\#\rangle$

1.3.3 *Degenerate Solutions*

In sequence, let us analyze a different cost function (Eqs. 1.19-1.20), where some quadratic terms are used. The terms, when translated into an Ising problem, indicate a coupling between nearest-neighbor spins in a linear lattice, using periodic boundary conditions.

$$b_{kl} = \begin{cases} 1, & k \in Chord;\ l \notin Chord \\ & k \notin Chord;\ l \in Chord \\ -1, & \text{otherwise} \end{cases} \tag{1.19}$$

$$a_k = -\frac{1}{2}\sum_{k \neq l} b_{kl} \tag{1.20}$$

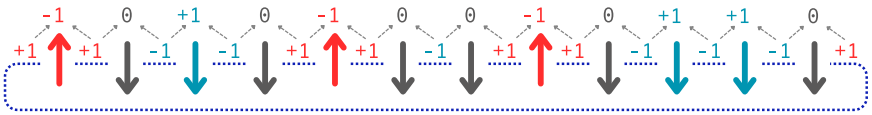


Fig. 1.12: Spin configuration of a C_{maj} with nearest-neighbour coupling

It is possible to verify that this cost function, together with the linear coefficients a_k, are designed so that the solution has two equally good configurations that minimize it: $|\uparrow\downarrow\downarrow\uparrow\downarrow\downarrow\uparrow\downarrow\downarrow\downarrow\downarrow\rangle$ and $|\downarrow\uparrow\uparrow\uparrow\downarrow\uparrow\uparrow\downarrow\uparrow\uparrow\uparrow\uparrow\rangle$ (Eq.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

1.21). In result, every time the experiment is run, the algorithm can reach a state where *Cmaj* is playing, or where all notes are playing except the *Cmaj* constituting notes are playing. The resulting state can also be influenced by the choice of the classical optimizer, initial conditions and ansaetze. To keep the same terminology as from the previous work, this state could be named *"anti-Cmaj"*.

$$\begin{array}{l} -\ ZIIIIIIIIIII \\ +\ IIZIIIIIIIII \\ -\ IIIIZIIIIIII \\ -\ IIIIIIIZIIII \\ +\ IIIIIIIIIZII \\ +\ IIIIIIIIIIZI \end{array} \quad + \quad \begin{array}{l} +\ ZZIIIIIIIIII \\ -\ IZZIIIIIIIII \\ -\ IIZZIIIIIIII \\ +\ IIIZZIIIIIII \\ +\ IIIIZZIIIIII \\ -\ IIIIIZZIIIII \\ +\ IIIIIIZZIIII \\ +\ IIIIIIIZZIII \\ -\ IIIIIIIIZZII \\ -\ IIIIIIIIIZZI \\ -\ IIIIIIIIIIZZ \\ +\ ZIIIIIIIIIIZ \end{array} \tag{1.21}$$

This can already be a powerful tool for music composition, where the cost function becomes a compositional object, and the behaviour of a specific run influences the result of a given music session (see Sec. 1.7.1).

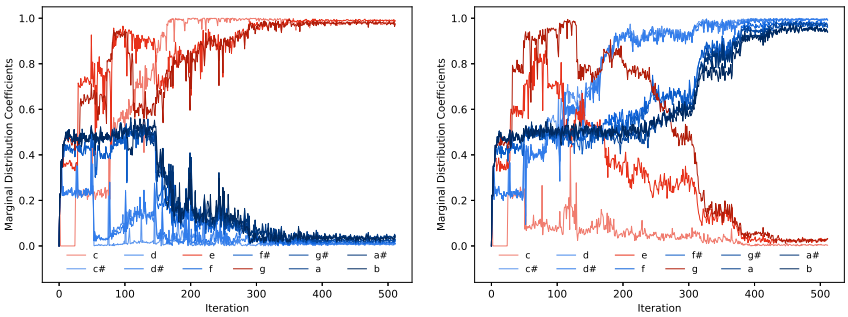
Chord flips Interestingly, when there is degeneracy in the solution (i.e., more that one state reaches minimum energy), it might happen that occasionally the VQE minimization starts approaching one solution, but suddenly encounters another minima, and switches its trajectory. One example of this behaviour can be seen in 1.13c, where the solution starts by favouring *Cmaj*, but "flips" entirely into an *anti-Cmaj* after around 300 iterations.

1.3.3.1 *Breaking Degeneracy*

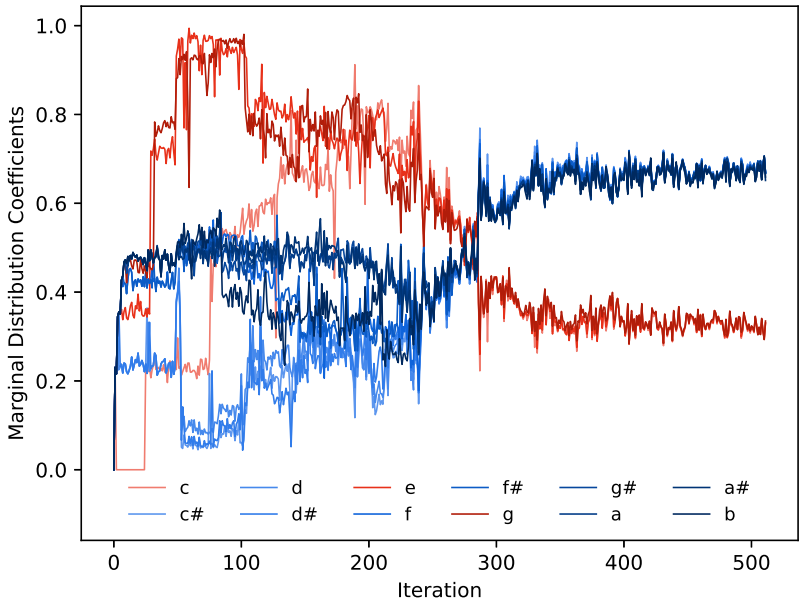
Occasionally it might be desirable to consider QUBO coefficients that are highly interconnected, but that still have a single ground state solution. This can be achieved by using corrective linear terms. As an example, starting from equation 1.19, one could introduce a constant $\bar{a} \in [-1, 1]$ to change the energy symmetry of the system (Eq. 1.22). In this case, it can be seen that positive values of $\bar{a}$ will benefit C_{maj}, and vice-versa. [9].

[9]Notice that this is an empirical result, and not a mathematical proof that *guarantees* symmetry/degeneracy breaking of the ground state.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9



(a) Converging to C_{maj} (b) Mixed solution



(c) Chord Flip from C_{maj} to $antiC_{maj}$

Fig. 1.13: Different results for a cost function with degenerate ground state

$$a_k = -\frac{1}{2}\sum_{k\neq l} b_{kl} + \bar{a} \quad . \tag{1.22}$$

To illustrate, consider the 4-spin example below, where both nearest neighbour and non-neighbour coupling terms are combined to make the second note flip its direction (Fig. 1.14). Then, a QUBO matrix is designed

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

Table 1.1: Designing QUBOs with linear and quadratic coefficients

	$\langle H\rangle$ (Eqs. 1.19 & 1.22)							
$\bar{a}$	$\|\downarrow\downarrow\downarrow\downarrow\rangle$	$\|\downarrow\downarrow\downarrow\uparrow\rangle$	$\|\downarrow\downarrow\uparrow\downarrow\rangle$	$\|\downarrow\downarrow\uparrow\uparrow\rangle$	$\|\downarrow\uparrow\downarrow\downarrow\rangle$	$\|\downarrow\uparrow\downarrow\uparrow\rangle$	$\|\downarrow\uparrow\uparrow\downarrow\rangle$	$\|\downarrow\uparrow\uparrow\uparrow\rangle$
4	-32	-12	-12	0	-28	0	0	20
3	-24	-8	-8	0	-24	0	0	16
1	-8	0	0	0	-16	0	0	8
0	0	4	4	0	-12	0	0	4
-1	8	8	8	0	-8	0	0	0
-3	24	16	16	0	0	0	0	-8
-4	32	20	20	0	4	0	0	-12
$\bar{a}$	$\|\uparrow\uparrow\uparrow\uparrow\rangle$	$\|\uparrow\uparrow\uparrow\downarrow\rangle$	$\|\uparrow\uparrow\downarrow\uparrow\rangle$	$\|\uparrow\uparrow\downarrow\downarrow\rangle$	$\|\uparrow\downarrow\uparrow\uparrow\rangle$	$\|\uparrow\downarrow\uparrow\downarrow\rangle$	$\|\uparrow\downarrow\downarrow\uparrow\rangle$	$\|\uparrow\downarrow\downarrow\downarrow\rangle$
4	32	20	20	0	4	0	0	-12
3	24	16	16	0	0	0	0	-8
1	8	8	8	0	-8	0	0	0
0	0	4	4	0	-12	0	0	4
-1	-8	0	0	0	-16	0	0	8
-3	-24	-8	-8	0	-24	0	0	16
-4	-32	-12	-12	0	-28	0	0	20

*Single ground states are denoted in blue.
**Degenerate Ground states are denoted in purple.

to satisfy equations 1.19 and 1.22. Table 1.1 shows the energy spectrum of the system for different values of $\bar{a}$. Notice that, as $\bar{a}$ increases, the linear solution $|\downarrow\downarrow\downarrow\downarrow\rangle$ starts to gain importance, until it becomes the predominant term.

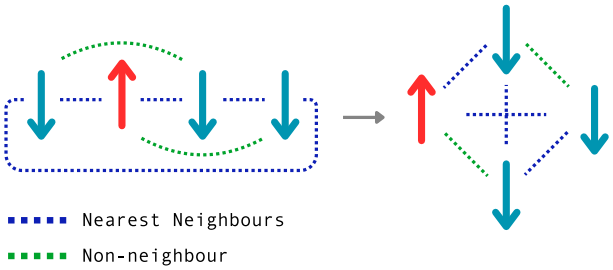


Fig. 1.14: Increasing the coupling

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

1.4 VQH implementation

A Real-Time Approach A distinctive feature of the implementation of VQH is on its integration, and interfacing between a quantum-centered environment and a music-centered one.

We proposed to realize this communication with a more integrated, modular, and dependency-inverted implementation. The aim is to design the VQH in a way that enables sonifications to be experimented on-the-fly, during a musical performance. Moreover, it could also enable researchers on Quantum Simulation problems to sonify their research while an experiment is running, providing them with a real-time auditory display of an experiment.

To that end, the Variational Quantum Harmonizer became structured as a *modular musical instrument*, or in a broader sense of the word, an *interface for music expression.*

Modularity The VQH implementation is modular and object-oriented, enabling independent users to define classes in separate python scripts - as specified in the source code documentation [Itaboraí *et al.* (2023b)] - to add customization and new features, such as different quantum hardware platforms and simulators, complex sonification mappings, new VQA implementations, optimization and simulation problems, encoding/decoding techniques, synthesis engine and music programming language integration.

1.4.1 *VQH Blueprint*

Based on the classification / characterization system for Digital Musical Instruments proposed by [Miranda and Wanderley (2006)], it is possible to represent the VQH by separating the elements related to Control, Mapping and Synthesis[10]. In a first layer of this illustration (Fig. 1.15), it is possible to note that the VQH has one control interface, two mapping stages, and one synthesis engine. In principle, the synthesis side can be implemented in any known synthesis platform or music programming language (See sec. 1.6).

In more detail, the terminology employed in the components of the VQH is explained: When considering a general VQA algorithm, the main input

[10]Note that in a final implementation of a musical interface, there might be no clear separation between, control, mapping and synthesis, as each component might serve in several roles across the system

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

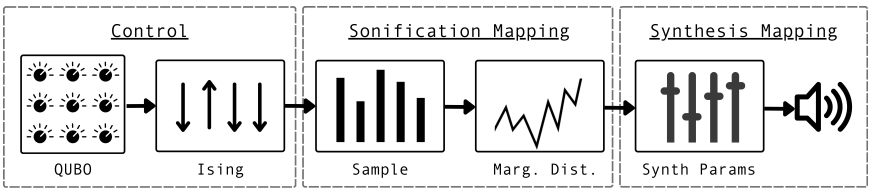


Fig. 1.15: VQH as a musical interface

of the system is the Hamiltonian of a system (***Problem***), where the energy expectation values to be minimized are extracted from (***Control***). This Hamiltonian is described as a list of operators decomposed as Pauli tensor products. The first mapping stage deals with polling data from the iterations of the VQA (***Generation***), which could be running with a QASM Simulator or in a real device (***Platform***) and decoding the information that will be used as a data stream for sonification (***Protocol***). Then, this data is transformed in the next stage into specific synthesis parameters, according to the chosen sonification strategy (***Mapping***). Finally, a synthesizer listens to these parameter streams and plays sounds (***Synthesis***).

Additionally, there is a higher level interface, where the user controls more global parameters of the system, general VQE configurations, editing parameters and variables, as well as general controls for running experiments and triggering sound (***Interface***). Further, there is a lower-level control, where it is possible to design the problem that is being optimized (e.g., the Pauli operators) and switch between the sonification mapping strategies, as well as choose how the sonification data is read and parsed to a synthesis engine (***Core***).

A visualization of the implementation can be seen in the diagram below (Fig. 1.16). Note that a box was inserted in the diagram to indicate that the data can also be plotted and visualized as an image by an external software (see Sec. 1.7.1.3 and Fig. 1.28 for an example). Generally, it is a conjunction of audio and image (and other sensorial media) that constitute the global concept of data visualisation, compehension and new insight on a dataset. In so, the VQH aims to be flexible for the introduction of other visualization techniques apart from sound.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

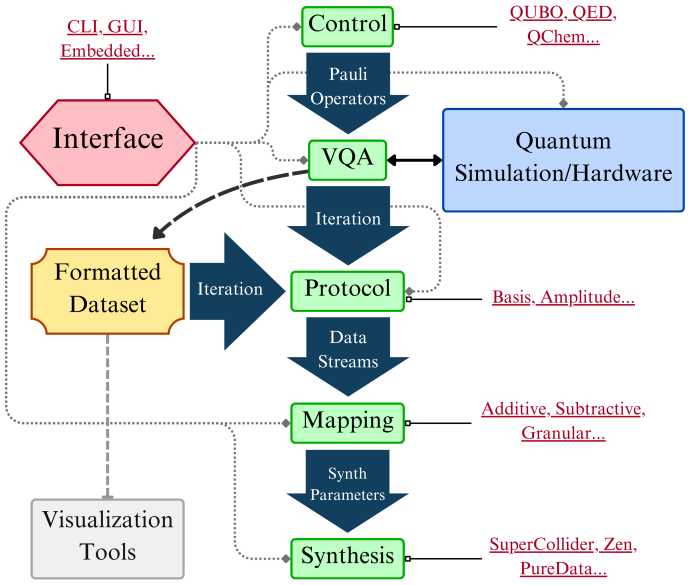


Fig. 1.16: Generalized VQH Structure

1.4.2 *Control*

1.4.2.1 *Controlling the QUBO Problem*

As seen in Sec. 1.2, the main problem studied so far within the VQH is the QUBO problem. The QUBO is transformed into an Ising Hamiltonian, by changing the binary variable basis n_i, which has values 0 and 1, into a Pauli Z basis, having eigenvalues ± 1. In the previous section 1.3, it was shown how the user can design cost functions using the harp analogy and interact with linear and quadratic QUBO coefficients to obtain specific results. When more complex couplings are used, knowing the ground state of your problem becomes more difficult.

Creating Superposition More generally, we can consider adding a transverse magnetic field to our Hamiltonian, controlled by a magnitude parameter h_x, which can introduce phase transitions and quantum effects

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

to the Ising model (Eq. 1.23)[11].

$$H(\sigma) = \sum_i \alpha_i \sigma_i^Z + \sum_{i,j \neq i} \beta_{ij} \sigma_i^Z \sigma_j^Z - h_x \sum_i \sigma_i^X \tag{1.23}$$

The introduction of a transverse field can lead to systems in which the ground state is a superposition state. As a result, with a sufficiently large field, it can be seen from equation 1.23 that the ground state of the Ising problem may be alternatively written in terms of the σ^X basis ($|\rightarrow\rangle$,$|\leftarrow\rangle$). In this case, a different mapping protocol could be designed, where measurements in the σ^X basis are also considered.

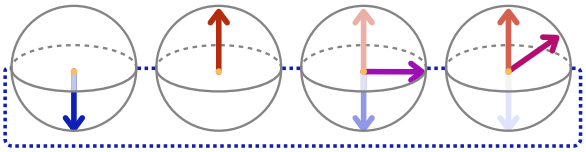


Fig. 1.17: Creating Superposition with transverse fields

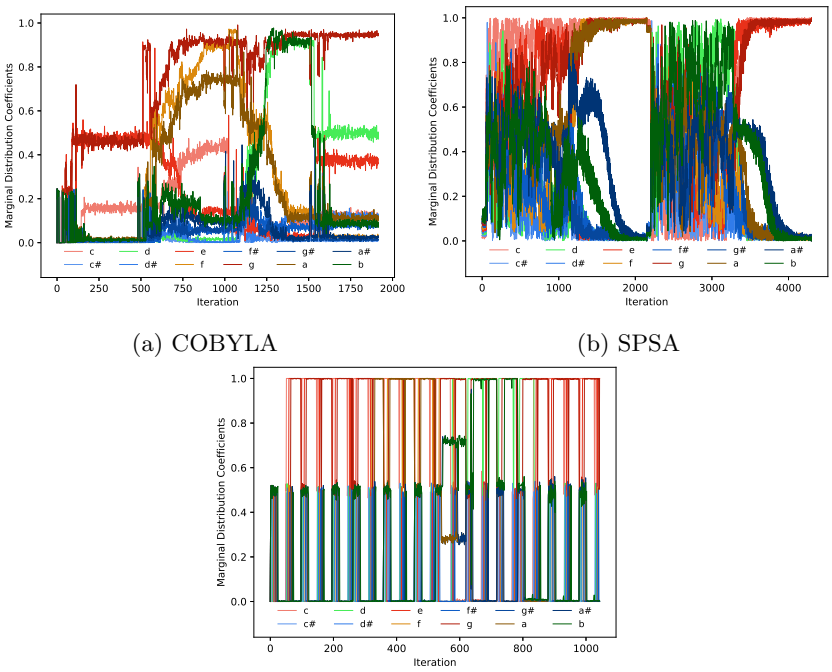
1.4.2.2 *Controlling the VQE*

One of the most relevant components of a VQA is the classical optimizer. As seen in 1.2, while the quantum side of the algorithm is responsible for evaluating the energies for a given parameterized state, the classical optimizer is the element that guides the changes in the configuration space. Thus, changing the optimizer can have a significant impact in the trajectory towards finding a result. Since the VQH is sensitive to iteration-by-iteration data, this also translates into noticeable perceptual changes in the sonification.

Artistic Intuition on the Classical Optimizers In Fig. 1.18, a comparison between three optimizers is made for a chord progression (Sec. 1.3.2). Restating the interpretation mentioned in [Itaboraí *et al.* (2023a)], the COBYLA algorithm (1.18a) seems to "perturb" each note individually while analyzing the changes in the cost function. Using the harp analogy for the basis encoding, it looks like the algorithm is "sweeping" through

[11] This parameter was also used for creating a musical gesture for *hexagonal Chambers* (Sec. 1.7.1), where the transverse field increased progressively towards the end of the piece

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9



(a) COBYLA (b) SPSA

(c) NFT

Fig. 1.18: Comparison between Optimizers

the strings (or muffling them), while deciding which tones "sound better" according to our hamiltonian listener.

On the other hand, SPSA [Spall (1992)] (Fig. 1.18b), has a more noisy profile, where the marginal distributions show a slower convergence. This means that usually many notes could be sounding simultaneously, until non-desired notes progressively fade out. This case could be particularly useful for constrained-randomness applications in composition, where parameters vary within moving boundaries during a complex gesture.

The NFT [Nakanishi *et al.* (2020)] algorithm (1.18c), in contrast, generates a periodic pattern, which is useful for creating rhythmic pulses.

1.4.2.3 *Implementation*

QUBO Matrix In the VQH software, the QUBO coefficients are stored and read from a comma separated matrix form in a CSV file. As shown in Figure 1.19, there is a header line that provides labels and metadata

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

to build the pauli operators and to facilitate the sonification mapping. To facilitate the creation of chord progressions, it is possible to define several Hamiltonians in the CSV file, in sequence. Figure 1.19 shows a 4-qubit example of a chord transition, using Eq. 1.22 with $\bar{a} = 0.5$.

Chords: 0010 → 1100

```
q1,   s0,   s1,   s2,   s3,
s0,  1.5,   -1,    0,   -1,
s1,   -1,  0.5,    1,    0,
s2,    0,    1, -0.5,    1,
s3,   -1,    0,    1,  0.5,
q2,   s0,   s1,   s2,   s3,
s0,  0.5,   -1,    0,    1,
s1,   -1,  0.5,    1,    0,
s2,    0,    1,  0.5,   -1,
s3,    1,    0,   -1,  0.5,
```

Fig. 1.19: Text-Based control of a chord transition (Eq. 1.22)

VQE Configuration Other relevant configurations for the VQE experiment are found in a separate JSON file (Fig. 1.2). If there is more than one QUBO matrix stored in the file (`sequence_length`), the VQH will parse the result of one experiment as the initial point for the next.

1.4.2.4 *Text-based and MIDI-based control*

Initially, the user could control the QUBO coefficients by directly editing and saving the CSV file using a preferred text editor. Similarly, the VQE configuration can be edited manually.

This approach can be useful for *Live Coding* performances (see Sec. 1.5.4), where typically there is an aesthetic demand for displaying text editors to the audience, sharing how the programmer-artist implements the sonification and interacts artistically with the code during the performance.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

Table 1.2: VQE configuration file on current version of VQH

Argument Name	Argument Description
`reps`	Number of Layers for the Ansatz
`entanglement`	Type of Entaglement*
`optimizer_name`	Classical Optimizer
`sequence_length`	Number of Hamiltonians
`size`	Size of the problem in qubits
`description`	Experiment Metadata
`iterations`	VQE iterations per Hamiltonian (list)
`nextpathid`	Experiment's unique idenfifier

*Specific for the EfficientSU2 Ansatz.

Extending this control, it is also possible to assign MIDI Control Change values in VQH to indirectly control the QUBO coefficients and VQE parameters. For instance, one could use a MIDI knob grid to change coefficients and a button/pad matrix to control configuration settings, change the mapping and toggle the sounds (Fig. 1.20).

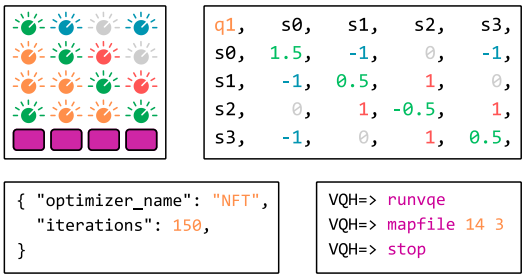


Fig. 1.20: Example of a MIDI Controller assigned to VQH parameters

1.4.3 *CLI Interface*

The current VQH version can be operated through a Command-Line Interface, implemented as an encapsulated prompt environment, with predefined functions for managing the interfacing between all the components of the instrument.

Inside the VQH file structure and a python environment with all dependencies installed, it is possible to run an *Experiment Session* by running the main script `VQH.py` and selecting the script arguments, as described below.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

Within a specific *Session*, the user can encapsulate a series of experiments, metadata and output data in a dedicated folder, which can be retrieved and used at a later time.

1.4.3.1 *Script Arguments*

```
python VQH.py [SESSIONPATH] [PLATFORM] [PROTOCOL]
```

- `SESSIONPATH` - The name of the folder where the VQE experiments and all generated data will be stored for this session. If the folder already exists, more data will be appended to the same folder.
- `PLATFORM` - Quantum Provider used. This is where the connection to quantum computing services is made possible. If a user has access to cloud services, their credentials can be used to connect to remote providers and execute jobs there. Alternatively, a local simulator can be used.
- `PROTOCOL` - Note encoding schemes/protocols for interpreting quantum states as notes, determining the global layer of sonification mapping. At the time of writing, the only implemented protocol was the `basis` protocol (See Sec. 1.2.1.1).

Example

```
$ python VQH.py Example local basis
```

In the session defined above, all generated data will be stored inside a folder named `Example_Data/`, each experiment has its own unique identifier '`<id>`', and its respective results will be found in the subfolder `Example_Data/Data_<id>`. Moreover, the session will perform jobs using a local provider (`qiskit_aer`).

1.4.3.2 *VQH Prompt*

After startup, the user is presented with a prompt "`VQH=>`", where internal commands can be used for operating the sonification.

- `VQH=> runvqe` - This command triggers a VQE experiment. When called, the command reads the current configuration files, the protocol being used, as well as the QUBO coefficients defined in the `h_setup.csv` file. Then the minimization algorithm is executed, following the specification, and the sonification data is saved in a

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

folder with an assigned identifier '`<id>`'. If more than one matrix was defined, it will run VQE for each Hamiltonian using the previous results as initial point, and concatenating them for saving the final dataset.

- `map [type]` - This command triggers the sonification for the last experiment that was run and loaded into memory. The [type] flag denotes the sonification method and synthesis engine used (see Sections 1.5 and 1.6).
- `mapfile [<id>] [type]` - Triggers the sonification using data from a previous experiment stored in the session folder, using the `<id>` to select the dataset.
- `stop` - Sends a "panic" message to stop all sounds. Necessary for some sonification mappings where the generated sound is continuous.
- `quit` or just `q` - Exits the program.

1.5 Sonification Mapping

After exploring the VQH implementation, and exploring different cost function designs and how the datasets are generated using the VQH interface, the focus is switched for the sound generation. This is a typical sonification mapping problem.

1.5.1 *Sonification Data*

At this stage of the VQH pipeline, it will be considered that an experiment session was run, and a dataset was generated and stored in a folder. Inside this folder, a set of files can be found.

Firstly, there are three files that contain metadata for the reproduction of the experiment. Respectively, they contain a copy of the QUBO coefficients, the VQE parameters, and a list of operators used. Secondly, there is a `.json` file containing raw results - a list of sampled distributions for each iteration. Third, there are files containing post-processed according to the chosen protocol, such as marginal distributions, observables, amplitudes, etc[12].

More specifically, after using the basis protocol, there should be three files of post-processed data:

[12]Notice that since the raw results are saved, the user can retrieve a previous experiment and sonify it using a different protocol.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

- A stream of Marginal Distribution Coefficients of a sampled statevector for each iteration ($c_n(t)$).
- Energy Expectation Value for each iteration ($E_0(t)$).
- The basis state with the highest sampled probability. ($|\psi_{max}(t)\rangle$)

Then, the mappings described below were proposed to map the data streams into synthesis control parameters.

1.5.2 *Simple Mappings*

1.5.2.1 *Additive*

One of the most direct and natural approaches one thinks about when considering a mapping problem is to use the simplest constituents of sound: Amplitude and frequency.

Subsequently, it is possible to consider the MDC as an ideal candidate for controlling relative amplitudes (or defining frequency variation). Furthermore, this framework is directly correlated with the harp analogy presented previously 1.2.1.1, where each qubit is assigned to an individual note. In addition, there are a myriad of options when it comes to the choice of frequencies for each note, which can fall into any desired musical scale.

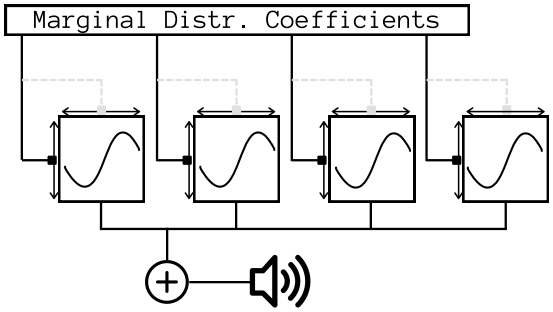


Fig. 1.21: Additive Synthesis Mapping

More generally, any method that consists of adding fundamental sounds, such as pure tones, oscillators, and sound banks, to create notes, chords, timbres, clusters and noises is considered to be *additive synthesis*. As a result, controlling the relative amplitudes of these sounds can change the

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

desired timbre and characteristics of the perceived sound.

1.5.2.2 *Frequency Modulation*

Under a similar additive structure, it is possible to use the MDC streams to control frequency variation. For each coefficient $c_n(t)$, there is a frequency $f_n(t)$.

In a preliminary approach, one could consider a linear (or logarithmic) mapping between the minimum and maximum values for the coefficients (c_{min}, c_{max}) and an audible range of frequencies (f_{min}, f_{max}), as exemplified below.

$$f_n(t) = \frac{(f_{max} - f_{min})}{(c_{max} - c_{min})}(c_n(t) - c_{min}) + f_{min} \quad \text{(linear)} \tag{1.24}$$

$$f_n(t) = f_{min} . \left(\frac{f_{max}}{f_{min}}\right)^{\frac{(c_n(t) - c_{min})}{(c_{max} - c_{min})}} \quad \text{(logarithmic)} \tag{1.25}$$

Inharmonicity Approach Another FM approach to be considered for sonification is proposed. Consider a tone generated from a harmonic series. For a given partial n, with frequency nf_0 it is possible to define a proportional shift $c_n(t)$ away from the harmonic equilibrium. In other words, it introduces inharmonicities to the sounds, which can potentially create complex metallic timbres (Eq. 1.26).

$$f_n(t) = (n - \frac{(c_n(t) - c_{min})}{(c_{max} - c_{min})})f_0 \tag{1.26}$$

1.5.2.3 *Subtractive*

Instead of accumulating tones, an alternative method could be to employ a subtractive or filtering strategy. One could generate a white noise (or start from any spectrally rich sound), and filter it down using a parallel set of Resonant (Band-Pass) Filters (RF).

Similar to the additive approach, it is possible to center each RF in frequencies that follow a determined musical scale. Then, each component of the MDC ($c_n(t)$) can be assigned as the gains of each filter.

To increase the complexity of the generated sound, the energy expectation value ($E_0(t)$) will be included, to control a second parameter of the synthesis. For instance, it could control the RF quality/resonant factor Q.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

A low value of energy could represent a high Q, and vice-versa. As a result, as the expectation value decreases, the sound approximates a pure tone, inferring a perceptual result similar to the harp analogy. Conversely, as the result diverges from the ground state, it approaches the original sound.

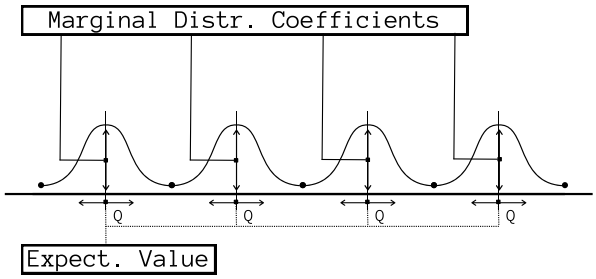


Fig. 1.22: Subtractive Synthesis Mapping

Rotary motors An alternate mapping for the expectation values, being a one-dimensional control signal, would be to leave apply a global pitchshift to the sounds - in other words, modulate the RF central frequencies. Depending on the intensity of the modulation and the RF resonant factor, the perceptual result can resemble a rotary motor experiencing variable friction, such as a drilling machine.

1.5.3 *Intermediary Mappings*

1.5.3.1 *Arpeggios*

For some applications, it might be desirable to listen to each marginal distribution coefficient more individually for a different distinction between relative amplitudes. This can be achieved by designing a *temporal expansion* around a VQE iteration. For example, instead of listening to a chord or cluster, the sound could be distributed in time as an *arpeggio*.

A suggested method for arpeggiation is demonstrated. First, in a fixed iteration i the collected coefficients $c_n(i)$ are mapped as intensities of individual percussive notes $c_n(i)p_n$. Then, the notes are sorted by amplitude, from the quietest to loudest note. These notes are played in a rapid succes-

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

sion, achieving an arpeggio. In addition, the time expansion rate (speed) of the arpeggio could be controlled by the expectation value. The closer from the ground state, the more compact the arpeggio becomes, until all notes ultimately become simultaneous again.

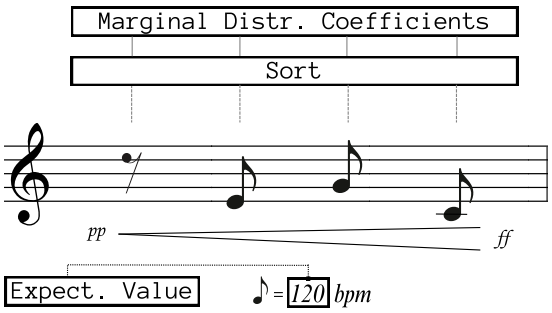


Fig. 1.23: Arpeggio Mapping

1.5.3.2 *Spatialization / Diffusion*

Another perceptual attribute of sound often used in music composition is its spatial location. Stereophony, for example, is popularly used in the recording industry. However, there are also many compositions that carefully position sound objects in space, using surround systems. Among the algorithms used to distribute sounds in space, we highlight Vector-Based Amplitude Panning (VBAP)[Pulkki (1997)], Distance-Based Amplitude Panning (DBAP)[Lossius and Pascal Baltazar (2009)], and higher-order Ambisonics [Poletti (2000)].

In contemporary music, sound systems are typically arranged in a circle around the audience, enabling the diffusion of quadraphonic, hexaphonic, octaphonic or hexadecaphonic sounds. The latter is usually configured in an upper and lower ring with 8 speakers each, achieving a three-dimensional spatialization. For illustration purposes, consider a sound object that can be placed at a fixed virtual distance away from a listener in any direction (Fig. 1.24). In this way, its position can be characterized by a parameter $\phi(t)$, which describes a dynamics around the circle. If there are N sound objects, their positions could be controlled by the MDC.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

$$\phi_n(t) = 2\pi \frac{(c_n(t) - c_{min})}{(c_{max} - c_{min})} \tag{1.27}$$

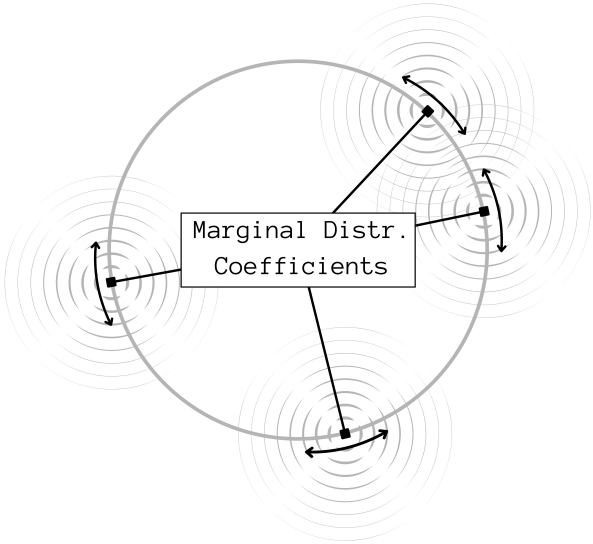


Fig. 1.24: Spatialization Mapping

Spectral Diffusion An interesting example of spatialization consists of dividing a sound into frequency bands using filter banks and spreading each band around the space ϕ_n, resulting in a spectral diffusion.

1.5.3.3 *Granular Synthesis*

Granular synthesis explores the phenomenon of sound in terms of particles rather than waves. Sparse sonic occurrences, or grains, are perceived by the ear as discrete, rhythmic events. As the grain density increases, a perceptual continuity emerges that can be experienced as pitch [Roads (2004)]. Grains can be short segments from a variety of sources - deterministic waveforms, noise or samples - and can be shaped with a variety of envelopes.

The sonic parameters of each grain can be considered within two perceptual domains simultaneously: temporally, including grain duration and

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

amplitude envelope, and spectrally, encompassing the frequency content of the individual grain and its perceived pitch. Moreover, the holistic organisation of grains can be approached in a number of ways, such as the granulation of prerecorded sounds or the implementation of physical models[Roads (2004)].

The wide range of time-domain and frequency-domain synthesis parameters used to define grains, along with the numerous algorithms employed to organise them, offers a wealth of opportunities for exploring different sonification methodologies within granular synthesis. We shall explore some of these techniques while presenting the compositional outcomes of the VQH in a following section (1.7).

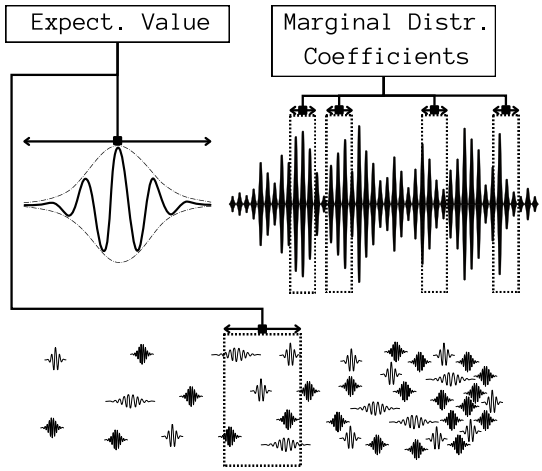


Fig. 1.25: Granular Synthesis Mapping Example

1.5.4 *Advanced Mappings: Live Coding*

Several of the mapping methods discussed previously operate within a linear input-output framework. Here, the results obtained from the Variational Quantum Eigensolver serve as control data for the chosen sonification strat-

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

egy, which then yields a single output. This approach aligns with more traditional methods of algorithmic composition; computational constraints require the composer to prepare a script in advance and then wait for the computer to synthesize the results.

With the advancement of computational power and the implementation of just-in-time programming within musical languages (as seen in Supercollider's JITLib), a seamless real-time interaction between composer and machine became achievable. The emergence of concepts such as "conversational programming," "on-the-fly programming," or "interactive programming" has led to the development of numerous creative languages and practices. Commonly termed 'live coding,' these activities have reshaped the landscape of algorithmic music over the past decades, moving it beyond its traditional academic confines to reach a wider audience [Magnusson and McLean (2018)]. Present-day live coding performances are hosted in a variety of both academic and non-academic contexts, where programmers craft code on stage while their laptop screens are projected for the audience to observe.

By utilising an appropriate live coding system, our aim is to engage with the data yielded by the VQE in an exploratory manner. This will make possible the concurrent employment of multiple synthesis techniques, as well as the ability to dynamically adjust the compositional algorithm in real-time as the data stimulates the creative ideas of the performer. In other words, in Live Coding, we depart from a more direct mapping of data onto sonic parameters, and move towards the use of data as a catalyst for creativity.

1.6 Synthesis

The implementation of a mapping strategy depends greatly on the synthesis platform used and how it can be connected to Python. The authors chose two platforms to carry the current work. First, Supercollider, which was used to implement the simple and intermediary mapping strategies. Then, the advanced mappings were implemented using Zen.

1.6.1 *Supercollider*

To establish a communication between the two systems, a Python client for SuperCollider was used [Jones (2020)]. This package implements API messages to carry messages to a supercollider server though OSC messaging,

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

eabling us to instantiate synthesizers, flexibly change synthesis parameters, manage nodes and audio buses, load buffers and trigger patterns from python. Altough it does not implement a fully featured `sclang`, it suffices for the purposes of this work.

The methodology is as follows. First, the user creates and compiles a synthesizer in SuperCollider, using the `SynthDef` Unit Generator. All parameters that should be controlled by python should be exposed as induvidual function arguments. Then, by using the "`.store`", method, the synthdef is compiled into machine code and saved automatically in a system folder.

Thereby, the key used to name the syntheziser can be used to create a synth node from python. Further, all exposed attributes can also be controlled by a '`.set()`' message.

1.6.2 *Zen*

Zen is a live coding language designed for expressing multidimensional musical patterns through succinct code, with a particular focus on pattern interference. Beyond the language itself, the Zen ecosystem encompasses an integrated development environment, featuring a code editor, pattern visualiser, and synthesis engine. All components are developed for the web, creating a comprehensive performance tool that necessitates no installation beyond a modern browser. Zen is built upon JavaScript, offering a familiar foundation for the majority of artist-programmers. However, it also incorporates some domain-specific syntax to augment JavaScript's capabilities.

Through the textual interface, Zen allows the user to express causal relationships between musical layers. A composer is able to define discrete patterns, and then identify musical or sonic parameters between patterns that should interfere with each other; for example, pitch, amplitude, timbral, or spatial parameters. Any parameter from another musical layer can be utilised and manipulated as the input for any other parameter. Composing in this way leads to surprising, often unintended results, challenging the common portrayal of inspiration as a form of guiding agent; instead, seeking to stimulate ideas through processes that go beyond the mind, as a means of creating original work. See Fell [Fell (2022), p. 21] and Magnusson and McLean [Magnusson and McLean (2018), p. 262] for further exposition upon the value of harnessing processes that go beyond the imagination of the composer. We hold the belief that such a compositional attitude resonates particularly well with the aims of the VQH project, which similarly

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

aims to unveil new musical territories by employing processes divergent from conventional musical practice.

Zen allows the user to map musical and sonic parameters across a three-dimensional canvas. Separate streams can be moved independently around this virtual space, with the parameters of sonic events being determined by their current position in time and space.

1.6.2.1 *Exporting data to Zen*

The transmission of control data to Zen is done via HTTP. One can serve a data file or folder onto a simple HTTP server, so that Zen's web interface can poll data from, so long as they are connected to the same network. This is how the piece *Dependent Origination* was technically realized. In summary, Zen can access data that is made available hosted locally or remotely through HTTP requests. The data is saved in a JSON file format, allowing Zen to parse the data by key. This process was improved for the piece *Hexagonal Chambers* by hosting a purpose-built API in the cloud, allowing each performer to connect via WIFI, rather than a local network. This is discussed in more detail in the following section (1.7.1).

1.7 Composing with VQH

In our capacity as composer-researchers, we have begun to explore the sonification of data generated using quantum algorithms through compositional processes.

Within the spectrum of quantum-computer assisted composition (QAC) we can highlight two distinct methods: either deferred, where musical forms and textures are carefully constructed outside of the act of performance, or in real-time, where improvisation is the prevalent process.

Acousmatic composition can be understood as a deferred compositional activity. Within this paradigm, the VQH can be used to manufacture, process, inter-connect, and inter-textualize a variety of sound objects in order to define the sonic and aesthetic landscape of a work. Conversely, the manipulation of synthesis parameters as a means of making the listener conscious of the interaction between artists during improvisation is a real-time activity, and one that positions the VQH as a musical instrument. Here, the interpreter learns to manipulate the QUBO coefficients to intentionally achieve musical variabilities.

Quantum technologies have not yet developed a mature artistic lan-

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

guage characteristic of their medium. However, the influence of quantum mechanics can be observed in the early history of electronic music; Dennis Gabor, Werner Meyer-Eppler, and others provided the theoretical basis for granular synthesis based on Gaussian particles [Gabor (1947)] [Roads (2004)]. The use of granularity provides us with both the historical and aesthetic motivation with which to explore the application of the VQH within new musical terrains.

Rasgar, Saber An initial example of an artistic output that used the VQH focused on exploring intermediary mappings (Sec. 1.5.3) to create sound objects used in an acousmatic composition. *Rasgar, Saber* (Itaboraí, 2023) is an acousmatic study that explores the concept of *Quantum Itineraries of Sound*, as defined and thoroughly explored by Itaboraí (2023). In this work, though the main focus was the quantum representation of audio and the ways that audio signals can be translated between different media, there was an additional aesthetic concern being to establish a connection between early electronic music vocabulary and QAC. In this context, 4-qubit QUBOs were mapped into a granular synthesis generator driven by paper sounds, which were recorded and later composed as transitions between sections of the piece. Additionally, these objects were spatialized by VQH in four channels by using the strategy discussed in section 1.5.3.2. Both mappings were implemented in SuperCollider.

1.7.1 *Hexagonal Chambers*

Our most fruitful musical outcomes have been achieved by employing live coding software to shape and transform the data in an exploratory manner (Sec. 1.5.4). We start with a direct mapping between data and sound, gradually loosening our approach as the composition unfolds and creative ideas emerge.

In technical research, a frequent tension arises between the simultaneous needs to prove a hypothesis and allow for the emergence of more ambiguous aesthetic considerations. To remedy this, we often seek to locate the poetic amongst the empirical, emphasizing the need for academic rigour to be complemented by the ambiguity of metaphor, myth, and storytelling. By exploring literary narratives that evoke the technical processes we are researching, an encouraging feedback loop emerges: our research inspires our music, and in turn, our music inspires further research.

Hexagonal Chambers (Thomas & Itaborai, 2023) links the optimization

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

process of variational quantum algorithms with a short story by Jorge Luis Borges; exploring the transition from chaos to order in all facets of musical language. The piece was performed using Zen, a live coding and data sonification system devised and built by Peter Thomas[Thomas (2024)] — discussed in the previous section — using data generated in real-time by Paulo Itaborai using the Variational Quantum Harmonizer.

In Jorge Luis Borges's short story "The Library Of Babel" [Borges *et al.* (1962)], the author explores the universe through the metaphor of a vast library. Comprised of a warren of hexagonal rooms, its catalog spans every possible permutation of language in a virtually infinite collection of incomprehensible texts. The library thereby contains all the knowledge ever to be written (and never to be written). Within this sea of chaos are the Vindications: books to absolve humanity of its sins, granting unity with God. And yet, due to the scale of the library, the search for these works — tomes of clarity amidst compendiums of noise, the light of meaning within a fog of meaninglessness — is considered to be futile.

In *Hexagonal Chambers*, the performers act as librarians, with rooms represented by self-contained Zen algorithms, and books being separate datasets returned from the VQH. Leveraging the possibilities of live coding, the performers are able to execute and edit different algorithms on-the-fly, swapping out the underlying datasets and using them as the basis for further improvisation. This symbolizes the librarian's search through different rooms and texts. The composition's structure mirrors the optimization procedure found in the VQH algorithm, progressing from a state of meaningless noise to a stable truth.

1.7.1.1 *Generating Data*

The VQH is operated during the performance to generate new *books*. Starting from simple linear QUBO matrices, and evolving in complexity to highly interconnected Ising systems with the inclusion of external Transverse Fields and non-linear coefficients (see sec. 1.4.2.1), a set of one to three new experiments was run for each section of the piece. Based on the current status of the piece, a new QUBO is designed and a classical optimizer selected. The choices are made with the intention to provoke musical gestures and variability. The parameters were controlled using a text-based approach, as the one illustrated in Fig. 1.26

As the VQH generates new data, they are uploaded to a purpose-built, remotely-hosted API. The VQH mapping function makes a HTTP `POST`

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

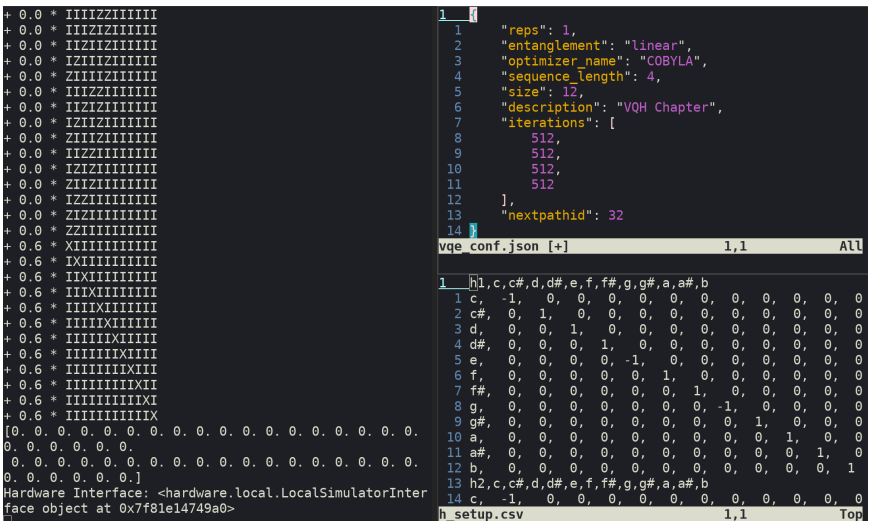


Fig. 1.26: VQH Interface for *Hexagonal Chambers*

call to an API endpoint. The API accepts new datasets, or books, and stores them as single entries in the database. Zen was modified to be able to request new books via HTTP, as they are written.

Each new experiment generated a specifically designed dictionary containing all data mentioned in section 1.5.1, in JSON format.

As the mapping is triggered in VQH and the dictionary is sent to the remote server, Zen receives a notification, informing it of the arrival of a new *book*.

1.7.1.2 *Sonifying Data*

In our six-qubit system, we mapped the marginal coefficients to the xy-positions of the separate streams on the Zen canvas (Fig. 1.27. Each stream was assigned to a separate instrument; which included samplers, and FM and granular synthesizers. Sonic events were triggered using the 6-bit binary string returned at each iteration, assigning one bit to each stream and triggering an event when the value was equal to 1. The resulting rhythms and spatial positioning were used as a basis for improvisation in real-time; experimenting with assigning different sonic parameters to each axis as the composition progressed.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

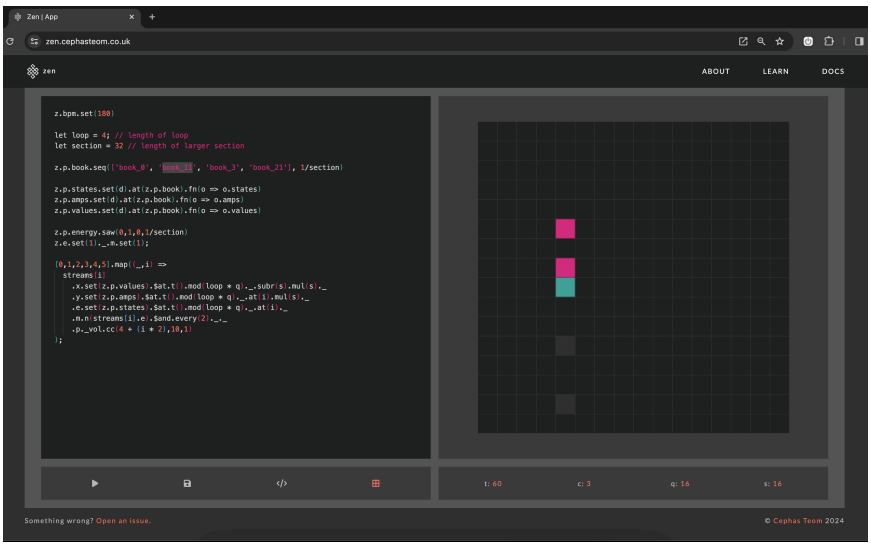


Fig. 1.27: Zen Interface for *Hexagonal Chambers*

1.7.1.3 *Syncing Zen and VQH*

For the execution of the piece, it was decided that the audience should also visualize the data being sonified. To that end, a real-time visualization module, implemented in Processing [Fry and Reas (2014)], was incorporated to the VQH system used in the performance. When a new *book* was sent to the API, it also updated a local file used by processing to create a coloured plot of the marginal distribution data (Fig. 1.28).

Furthermore, a MIDI connection was created between the machines running Zen and VQH. As a consequence, it was possible to send clock ticks from Zen's procedural step-generation as MIDI Control Change messages. This information was used to move a cursor in the visualization's plot, indicating which data points were driving the sound at each step.

To send the time information as MIDI CC, which can span only 128 different values, the clock values had to be encoded into a stream of three consecutive MIDI messages, which could represent enough time steps in binary form to span the expected duration of the musical performance.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

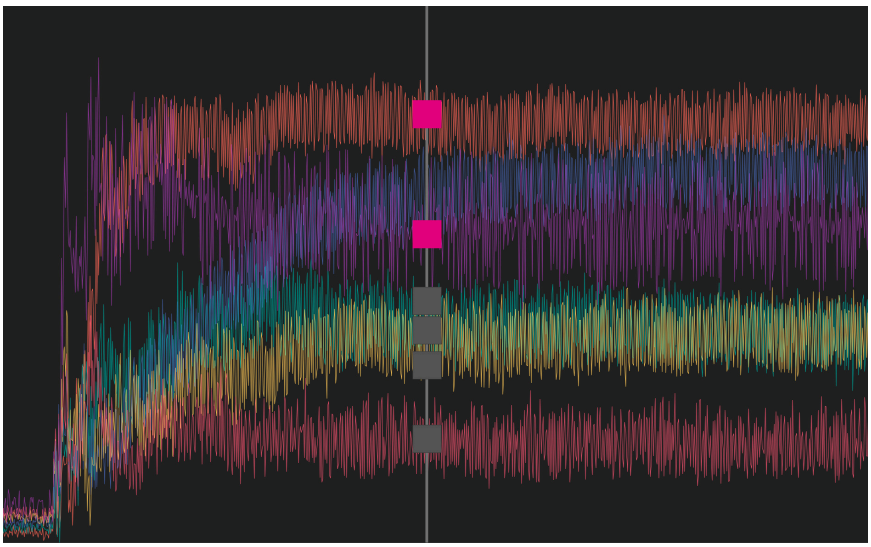


Fig. 1.28: Live visualization of the sonification data in Processing

1.8 Discussion

A novel framework for the sonification of Variational Quantum Algorithms was presented, and specific applications using a QUBO optimization problem and VQE minimization were shown. The implemented software interface, Variational Quantum Harmonizer, enabled artists to integrate quantum technologies in their compositional process, both in technical and aesthetic aspects, by using quantum principles as a musical form. These principles are used through quantum algorithms and methods that extend the traditional possibilities of procedural sound generation to the navigation of larger (Hilbert) spaces. The way a quantum computer processes information is profoundly different from the classical approach and requires its own musical language so that its manipulation can create meaningful sounds.

The VQH demonstrates preliminary pathways for the creation of audible visualization for Quantum Algorithms, by focusing on mapping sampled distributions of quantum states into sound. This structure could also be employed by researchers to sonify complex physical systems in the natural sciences to construct auditory displays of a problem of interest, potentially facilitating experiment workflows and the detection of new properties. Further, there are physical systems that are hard to compute with classical systems, such as the study of real-time phenomena in complex systems.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

These physical problems can benefit both from quantum computational approaches for simulation and from sonification. The elaboration of complex sonification mappings for large systems often involves artistic practices and practitioners who engage in interdisciplinary research. A noticeable example is the work on data sonification / musification of Wigner functions and Rabi Oscillations, realised within a prestigious institution of photonic research [Yamada *et al.* (2023)].

Moreover, VQH can be used as a powerful tool for training and education, specially when introducing Variational Quantum Algorithms. Sound and music can provide intuition to a broader range of individuals, even when they lack a formal musical background. For example, an audience in a concert can easily sing simple intervals or scales just by intuition. Thus, a deeper insight on how Variational Quantum Algorithms optimize can be achieved by *listening* to the process of optimization. This can be interpreted as a perceptual approach to gain understanding on abstract processes an ideas.

As a blueprint of a sonification toolbox, the VQH has a modular implementation, which enables for flexible scalability for the sonification of real complex problems, for circuit optimization and integration in different quantum computing cloud services and hardware, as well as for careful sound design tailored to a specific musical or scientific application.

The implementation of the VQH is Open Source, and the source code, together with audio examples and code snippets, can be found and downloaded online [Itaboraí *et al.* (2023b)].

1.8.1 *Future work*

At the time of writing, the authors are extending the implementation of the VQH for further research on the sonification. The following research directions are being currently addressed:

- **Generalized OSC Communication**: The current implementation of VQH strongly reliant on SuperCollider and Zen. To become more flexible, a generic mapping is being developed, which dumps the sonification data with formatted Open Sound Control messages, which could be grabbed by any OSC-enabled software.
- **Musical Intrument Design**: Improvements on the implementation will permit the embedding of this sonification toolset in more encolsed systems, such as a Digital and Augmented Musical Instru-

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

ments.

- **New Sonification Protocols**: There is an ongoing investigation on alternative qubit-efficient encoding methods and possibilities for mapping notes and musical features into quantum states. Two examples are Amplitude Encoding and Pauli-correlation Encoding [Sciorilli *et al.* (2024)]. Another range of encodings can be found in Quantum Representations of Audio [Itaboraí (2023)].
- **Sonification of High Energy Physics**: The group has been investigating the problem of sonifying Lattice Gauge Theories using VQH, such as the simulation of lattice QED in 2+1-dimensions [Clemente *et al.* (2022a)].
- **Generalized Signals**: Sonification methods create one-dimensional signal streams, which are transformed into audio. However, in principle, this signal could be interpreted or transformed into other types of signals, from symbolic musical data, to images, ultrasound, microwaves, brain wave signals, and more.

1.9 Acknowledgements

P.I's and K.J.'s works are funded by the European Union's Horizon Europe Framework Programme (HORIZON) under the ERA Chair scheme with grant agreement no. 101087126 (QUEST ERA Chair).

K.J. and A.C. are supported with funds from the Ministry of Science, Research and Culture of the State of Brandenburg within the Centre for Quantum Technologies and Applications (CQTA).

A. C. is supported in part by the Helmholtz Association —"Innopool Project Variational Quantum Computer Simulations (VQCS)."

T.S. is supported by the Einstein Research Unit -"Perspectives of a quantum digital transformation: Near-term quantum computational devices and quantum processors"

We thank Goethe-Institut for programming the piece *Hexagonal Chambers* at the *Wave Dysfunction* performance by Eduardo R. Miranda and QuTune Collective (including P.I, P.T. and M.Y.) during the 25th edition of the CTM Festival in Berlin, through their Studio Quantum Project. The

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

event took place at Radialsystem, Berlin, 02 February 2024.

The VQH project started as a collaboration between the Interdiciplinary Centre for Computer Music Research at the University of Plymouth and the Center for Quantum Technologies and Applications at DESY, by initiative of professors Eduardo Reck Miranda and Karl Jansen.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

Bibliography

Aleksandrowicz, G., Alexander, T., Barkoutsos, P., Bello, L., Ben-Haim, Y., Bucher, D., Cabrera-Hernández, F., Carballo-Franquis, J., Chen, A., Chen, C., *et al.* (2019). Qiskit: An open-source framework for quantum computing.(jan. 2019), *URL https://qiskit.org/documentation/stubs/qiskit* .

Borges, J., Kerrigan, A., Reid, A., Bonner, A., Temple, H., and Todd, R. (1962). *Ficciones. C Edited and with an Introduction by Anthony Kerrigan*, An Evergreen book, E-368 (Grove Press), ISBN 9780802130303, `https://books.google.co.uk/books?id=1FrJqcRILaoC`.

Cage, J. (1973). *Experimental Music* (WESLEYAN UNIVERSITY PRESS), ISBN 9780819560285, pp. 7–12, `https://books.google.com.cy/books?id=zKQkLS5zKWAC`.

Clemente, G., Crippa, A., and Jansen, K. (2022a). Strategies for the determination of the running coupling of (2+1)-dimensional qed with quantum computing, *Physical Review D* **106**, 11, doi:10.1103/physrevd.106.114511, `http://dx.doi.org/10.1103/PhysRevD.106.114511`.

Clemente, G., Crippa, A., Jansen, K., and Tüysüz, C. (2022b). *New Directions in Quantum Music: Concepts for a Quantum Keyboard and the Sound of the Ising Model* (Springer), ISBN 978-3-031-13909-3, pp. 433–445, doi:10.1007/978-3-031-13909-3_17, `https://doi.org/10.1007/978-3-031-13909-3_17`.

Corredera, J. M. and Tanaka, A. (2023). EMG sonification as a tool for functional rehabilitation of spinal-cord injury. in *Proceedings of the International Conference on New Interfaces for Musical Expression* (NIME), `https://nime.org/proceedings/2023/nime2023_93.pdf`.

Curtiss, L. (1950). *The Geiger-Müller Counter*, Circular of the National Bureau of Standards (U.S. Government Printing Office).

Date, P., Arthur, D., and Pusey-Nazzaro, L. (2021). QUBO formulations for training machine learning models, *Scientific reports* **11**, 1, p. 10029.

Fell, M. (2022). *Structure and Synthesis: The Anatomy of Practice* (MIT Press), ISBN 9781913029951.

Fletcher, N. H. and Rossing, T. D. (1998). *The Physics of Musical Instruments* (Springer Science & Business Media), ISBN 978-0-387-98374-5.

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

Fry, B. and Reas, C. (2014). Processing library for visual arts and design, *URL http://www. processing. org* .

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

Glinsky, A. (2000). *Theremin: ether music and espionage* (University of Illinois Press), ISBN 9780252025822.

Hamido, O. C. (2022). *QAC: Quantum-Computing Aided Composition* (Springer), ISBN 978-3-031-13909-3, pp. 159–195, doi:10.1007/978-3-031-13909-3_8, https://doi.org/10.1007/978-3-031-13909-3_8.

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

Hamido, O. C. and Itaboraí, P. V. (eds.) (2023b). *Proceedings of the 2nd International Symposium on Quantum Computing and Musical Creativity* (Zenodo), https://isqcmc.github.io/Proceedings, [Conference proceedings].

Itaboraí, P. V. (2023). *Towards Quantum Computing for Audio and Music Expression*, Master thesis, University of Plymouth, doi:10.24382/5119.

Itaboraí, P. V. and Miranda, E. R. (2022). *Quantum Representations of Sound: From Mechanical Waves to Quantum Circuits* (Springer), ISBN 978-3-031-13909-3, pp. 223–274, doi:10.1007/978-3-031-13909-3_10, https://doi.org/10.1007/978-3-031-13909-3_10.

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

Itaboraí, P. V., Schwägerl, T., Yáñez, M. A., Crippa, A., Jansen, K., Miranda, E. R., and Thomas, P. (2023b). Variational Quantum Harmonizer (Source Code), https://github.com/iccmr-quantum/VQH, Last Accessed 26 Mar 2024 [Software].

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

Landy, L. (ed.) (2024). *Organised Sound* (Cambridge University Press), ISSN 1355-7718 [Journal].

Lossius, T. and Pascal Baltazar, T. d. l. H. (2009). Dbap - distance-based amplitude panning, in *Proceedings of the International Computer Music Conference (ICMC)*, https://jamoma.org/publications/attachments/icmc2009-dbap-rev1.pdf.

Magnusson, T. and McLean, A. (2018). *The Oxford Handbook Of Algorithmic Music*, chap. Composing With Patterns Of Time (Oxford UP).

Matsuura, S., Yamazaki, T., Senicourt, V., Huntington, L., and Zaribafiyan, A. (2020). Vanqver: the variational and adiabatically navigated quantum eigensolver, *New Journal of Physics* **22**, 5, p. 053023, doi:10.1088/

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

1367-2630/ab8080, https://dx.doi.org/10.1088/1367-2630/ab8080.

McCartney, J. (2002). Rethinking the computer music language: SuperCollider, *Computer Music Journal* **26**, 4, pp. 61–68.

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

Miranda, E. R. (ed.) (2022). *Quantum Computing in the Arts and Humanities: An Introduction to Core Concepts, Theory and Applications* (Springer International Publishing AG, Cham), ISBN 978-3-030-95537-3.

Miranda, E. R. (ed.) (2022b). *Quantum Computer Music: Foundations, Methods and Advanced Concepts* (Springer International Publishing AG, Cham), ISBN 978-3-031-13909-3.

Miranda, E. R. and Shaji, H. (2023). Generative music with partitioned quantum cellular automata, *Applied Sciences* **13**, 4, doi:10.3390/app13042401, https://www.mdpi.com/2076-3417/13/4/2401.

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

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

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

Nakanishi, K. M., Fujii, K., and Todo, S. (2020). Sequential minimal optimization for quantum-classical hybrid algorithms, *Physical Review Research* **2**, 4, p. 043158.

NIME Proceedings (2021). *Proceedings of the International Conference on New Interfaces for Musical Expression*, ISSN: 2220-4806.

Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P. J., Aspuru-Guzik, A., and O'Brien, J. L. (2014). A variational eigenvalue solver on a photonic quantum processor, *Nature Communications* **5**, 1, p. 4213, doi:10.1038/ncomms5213, https://www.nature.com/articles/ncomms5213.

Poletti, M. A. (2000). A unified theory of horizontal holographic sound systems, *Journal of the Audio Engineering Society* **48**, 12, pp. 1155–1182.

Powell, M. J. (1994). *A direct search optimization method that models the objective and constraint functions by linear interpolation* (Springer).

Puckette, M. *et al.* (1996). Pure data: another integrated computer music environment, *Proceedings of the second intercollege computer music concerts* , pp. 37–41.

Pulkki, V. (1997). Virtual sound source positioning using vector base amplitude panning, *Journal of the Audio Engineering Society* **45**, 6, pp. 456–466.

Roads, C. (1996). *The computer music tutorial* (MIT press), ISBN 9780262680820.

Roads, C. (2004). *Microsound* (MIT press), ISBN 0-262-18215-7.

Rodrigues Miranda, C. (2019). A sounded journey from nano to macro into everyday materials through multiscale molecular simulations, in *19th Inter-*

This is a non-edited pre-publication version of a chapter to appear in the book
"*Advances in Quantum Computer Music*", World Scientific, editor E. R. Miranda, 2024.
DOI:10.1142/14025 ISBN:978-981-98-0017-9

national Workshop on Computational Physics and Material Science: Total Energy and Force Methods (ICTP), seminar.

Scarassatti, M. A. F. (2001). *Retorno ao futuro: Smetak e suas plasticas sonoras [Return to the future: Smetak and his sonic plastics]*, Master thesis, State University of Campinas, in Portuguese.

Sciorilli, M., Borges, L., Patti, T. L., García-Martín, D., Camilo, G., Anandkumar, A., and Aolita, L. (2024). Towards large-scale quantum optimization solvers with few qubits, `arXiv:2401.09421 [quant-ph]`.

Spall, J. (1992). Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Transactions on Automatic Control* **37**, 3, pp. 332–341, doi:10.1109/9.119632.

Straebel, V. (2010). The sonification metaphor in instrumental music and sonification's romantic implications, in *Proceedings of the 16th International Conference on Auditory Display (ICAD-2010)* (Georgia Institute of Technology), pp. 287–294, `http://hdl.handle.net/1853/50066`.

Thomas, P. (2024). Zen software, `https://zen.cephasteom.co.uk/`.

Weaver, J. L. (2022). *Quantum Music Playground Tutorial* (Springer), ISBN 978-3-031-13909-3, pp. 197–222, doi:10.1007/978-3-031-13909-3_9, `https://doi.org/10.1007/978-3-031-13909-3_9`.

Wright, M. and Freed, A. (1997). Open Sound Control: A new protocol for communicating with sound synthesizers, in *International Computer Music Conference (ICMC)*, Vol. 1997, pp. 101–104, `http://hdl.handle.net/2027/spo.bbp2372.1997.033`.

Xenakis, I. (1992). *Formalized Music: Thought and Mathematics in Composition*, Harmonologia series (Pendragon Press), ISBN 9781576470794.

Yamada, R., Piñol, E., Grandi, S., Zakrzewski, J., and Lewenstein, M. (2023). Towards the intuitive understanding of quantum world: Sonification of rabi oscillations, wigner functions, and quantum simulators, in *2nd International Symposium on Quantum Computing and Musical Creativity (ISQCMC Berlin)* (Zenodo), doi:10.5281/zenodo.10206508, `https://doi.org/10.5281/zenodo.10206508`.

Yáñez, M. A. (2023). Developing a quantum step-sequencer, in *2nd International Symposium on Quantum Computing and Musical Creativity (ISQCMC Berlin)* (Zenodo), doi:10.5281/zenodo.10206500, `https://doi.org/10.5281/zenodo.10206500`.

Zanella, A., Harrison, C. M., Lenzi, S., Cooke, J., Damsma, P., and Fleming, S. W. (2022). Sonification and sound design for astronomy research, education and public engagement, *Nature Astronomy* **6**, 11, pp. 1241–1248, doi:10.1038/s41550-022-01721-z, `https://doi.org/10.1038/s41550-022-01721-z`.

Zicarelli, D. (1998). An extensible real-time signal processing environment for Max, in *Proceedings of the International Computer Music Conference 98*, pp. 463–466.