Quantum Synth: a Quantum-Computing-based synthesizer

Omar Costa Hamido
ocostaha@uci.edu
University of California, Irvine
Irvine, California

Giovanni Amedeo Cirillo
giovanni_cirillo@polito.it
Politecnico di Torino
Turin, Italy

Edoardo Giusto
edoardo.giusto@polito.it
Politecnico di Torino
Turin, Italy

ABSTRACT

In this paper we present the *Quantum Synth* project, an interface between Qiskit and Max for controlling sound synthesis parameters encoded on the basis states of a quantum computer. With this representation, sound synthesis is obtained from the potential measured outcomes of a quantum circuit. The effects of using fundamental quantum operations as found in the Bell circuit, for the generation of entangled states, and the Grover's search algorithm have been demonstrated. The interface has been designed to be used by music performers and composers in their creative process, and as a resource to both learn Quantum Computing and analyze the intrinsic noise of real quantum hardware.

CCS CONCEPTS

• **Applied computing → Sound and music computing**; *Performing arts*; • **Hardware** → Quantum computation.

KEYWORDS

Sound Synthesis, Quantum Computing, Qiskit, Max

ACM Reference Format:
Omar Costa Hamido, Giovanni Amedeo Cirillo, and Edoardo Giusto. 2020. *Quantum Synth*: a Quantum-Computing-based synthesizer. In *Proceedings of Audio Mostly (AM'20), September 15–17, 2020, Graz, Austria.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Quantum Mechanics and Acoustics are two correlated areas of Physics, since both their phenomena can be described in terms of waves. The possibility to adopt the mathematical formalism of Quantum Mechanics to describe acoustic phenomena has been proposed since the half of the previous century [3] and, more recently, the description of spin systems has been employed for analyzing the relations between turbulence, phonation and slow myoelastic vibrations [7]. In order to further investigate the relationships between these areas, a system for translating Quantum Mechanics phenomena into sound could be useful. The idea for Quantum Synth [5] was developed during the Qiskit camp Europe Hackathon, in Schilthorn, Switzerland. It is an audio synthesizer based on the imperfections of nowadays programmable Quantum Computers, inherently affected by noise. The tool is on the one hand didactic, providing a different way to learn and teach about certain aspects of quantum computing, and, on the other hand, it can also be used for music performing and composition. The *Quantum Synth* project received the Qiskit Community Choice Award at the venue.
After a brief introduction to Quantum Computing in Section 2, we describe in Section 3 the mechanism for controlling sound synthesis from a quantum computer, and the interface between a digital synthesizer and real quantum computers of International Business Machines (IBM). Section 4 reports some demonstrations of quantum-based synthesis; finally, future perspectives for the presented interface are introduced in Section 5.

2 QUANTUM COMPUTING

In classical microprocessors, binary information, the so-called **bit**, is encoded onto electrical voltages, and operations consist in changing the state of switches, implemented by transistors. On the other hand, a quantum processor encodes information onto a quantum physical quantity, as the spin states of an electron or the energy levels of an atom. The quantum unit of information, named **qubit**, is characterized by the so-called *superposition*, a typical property of quantum systems for which it shows non-null probabilities of being measured in two different states at the same time. A qubit described by superposition states can be represented by a vector delimited by the origin of the Cartesian axes and any point of the surface of the Bloch sphere (see Figure 1), whose poles $|0\rangle$ and $|1\rangle$ correspond to the basis states, which are also the admitted states of a classical bit. Superposition is the fundamental property allowing to qubits to be more powerful than their classical counterparts, since a higher degree of computing parallelism can be exploited by employing them; in fact, the same operation can be simultaneously

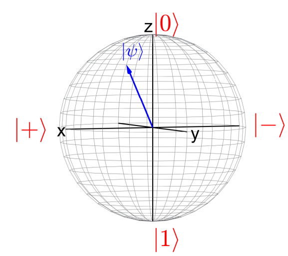


Figure 1: Bloch sphere, where the current state of the qubit is described by the vector

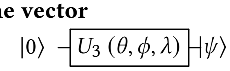


Figure 2: One-qubit quantum circuit for generating a generic superposition state $|\psi\rangle$ from the basis state $|0\rangle$

AM'20, September 15–17, 2020, Graz, Austria

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

evaluated on multiple data (*e.g.* all the numbers representable with N qubits, *i.e.* from 0 to $2^N - 1$). Some problems, like searching an element in an unsorted database [4], have been already proved to be more efficiently solved by a quantum computer than a classical one. Operations on a quantum computer are implemented, according to the so-called quantum circuit model [6], as sequences of quantum gates changing the probability distribution of the outcomes of the quantum computer, as the single-qubit $U_3(\theta, \phi, \lambda)$ gate in Figure 2. Quantum computers have been already built and they can also be programmed *via*-cloud. Currently, the most easily accessible hardware is the one provided by IBM-Q. This tool gives the possibility to design quantum circuits and execute them on:

- real devices, with different number of qubits (up to 16);
- simulators, i.e. classical computers simulating a quantum computer in presence or in absence of noise models for the emulation of real hardware.

Quantum programming is facilitated by Qiskit [2], "an open source quantum computing software development framework for leveraging today's quantum processors". Qiskit supports interfaces for both real devices and simulators. It is important to precise that current quantum computers are not very resilient to the effects of non-ideal phenomena caused by their interaction with the external environment. Among other things, this means that they can provide reliable results only when few tens of quantum gates are applied on them. After timescales of tens of microseconds, the quantum system naturally decays in its lowest energy state (which corresponds to the state $|0\rangle$ on the quantum computer).

3 THE IDEA: QUANTUM-BASED SYNTHESIS

The timbre of musical instruments can be considered - according to the Fourier theory - made of multiple harmonic or tones, nothing but sine waves of different frequency, amplitude, and phase. The combination of different harmonics generates sounds with different timbres, and this is what allows us to perceive the difference between musical instruments. A music synthesizer electronically generates and modifies sounds: waveforms generated in hardware or software are then altered in intensity, duration, frequency, and timbre, according to the preferences of the musician. This technique allows one to produce sounds with harmonics/notes on a frequency range much different than conventional musical instruments. In the current version of *Quantum Synth* the subtractive synthesis is available: a white noise audio signal is filtered by up to twelve bandpass filters in order to alter its timbre with the selected specific harmonics to be kept. The probability distribution of a quantum circuit after applying a sequence of quantum gates is exploited for encoding the frequencies to be selected with this synthesis. *Quantum Synth* currently uses four qubits to obtain the probability distribution exploited in sound synthesis, each resulting state is associated to a center frequency band in an octave. The block scheme of the system is reported in Figure 3. The main components are Qiskit and Max, a visual programming language for music and multimedia [1]. The composer can choose the configuration mode of the synthesizer and the quantum circuit to be exploited for sound generation in Max, then the selected quantum circuit is generated in Qiskit and executed on one of the different IBM-Q backends, real or simulated. Finally, the results provided by the quantum circuit are sent back to Max and exploited for music synthesis. A Max interface has been developed for subtractive synthesis, which allows the user to choose:

- the backend, which could be a simulator (affected or not by noise) or a real quantum computer;
- the quantum circuit employed as reference for synthesis operations;
- the Q-factor for harmonic selection, typical of filtering in subtractive synthesis;
- the noise fidelity.

Qiskit returns a string containing the probabilities to be exploited for sound synthesis. In order to interface Max and Qiskit, a User Datagram Protocol (UDP)-based server was developed. The middleware parses a configuration command from Max, and then it runs the circuit on the selected backend via Qiskit. After finishing the job, the middleware sends the final set of probabilities to the Max application, thus closing the loop of the block-scheme (Figure 3).

4 DEMONSTRATION

The key components of *Quantum Synth* are the prebuilt quantum circuits. Each circuit produces a specific state for the qubits, that are then encoded into the assigned synthesis parameters. The quantum circuits currently interfaced with *Quantum Synth* are:

- Bell state generator, which entangles two qubits;
- Equal superposition of N states which can be chosen by the user;
- Hadamard, for the equal superposition (same probability) of all states;
- Grover's search quantum circuit.

Again, the results can be retrieved from real hardware or simulator (with or without noise models). In fact, since each circuit outputs a different probability distribution, they will sound different from each other. Furthermore, due to the noise, the same circuit can output slightly different results if run on a real machine or simulated with a noise model! Four use cases are provided in the following subsections, in order to understand not only how *Quantum Synth* generally works, but also how noise can be significant in the probability distribution at the end of a quantum circuit.

4.1 Bell state generation

Entanglement is a property of quantum mechanics for which two or more states cannot be described separately, but by a unique state vector/wave function, because of an intrinsic correlation between them. The simplest proof of this correlation is the immediate recognition of the values of the whole entangled qubits system when a qubit of theirs is measured. Given the Bell state $\left|\Phi^+\right\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$, if $|0\rangle$ is measured on the left qubit, the right one will be surely $|0\rangle$ without any measurement. From a more general computational point of view, this property can be exploited for reducing the number of operations required for reaching a problem's solution. *Quantum Synth* presents a pre-defined routine for generating sounds from the quantum circuit generating $\left|\Phi^+\right\rangle$ in Figure 4. It is possible to clearly observe two peaks in the spectrum in Figure 5(a), one associated to the eigenstate $|00\rangle$, the other to $|11\rangle$.

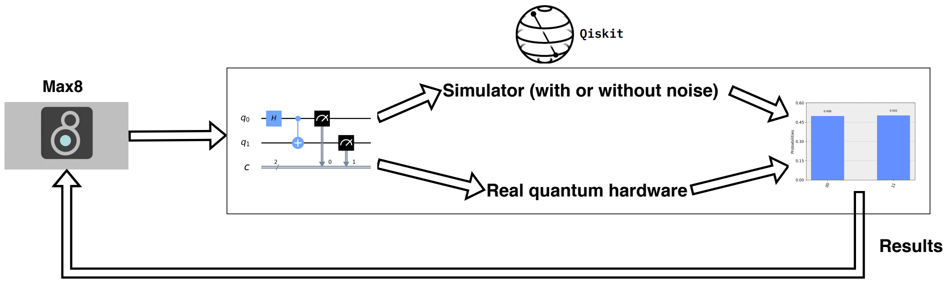


Figure 3: Block scheme of Quantum Synth.

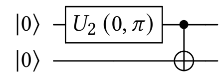
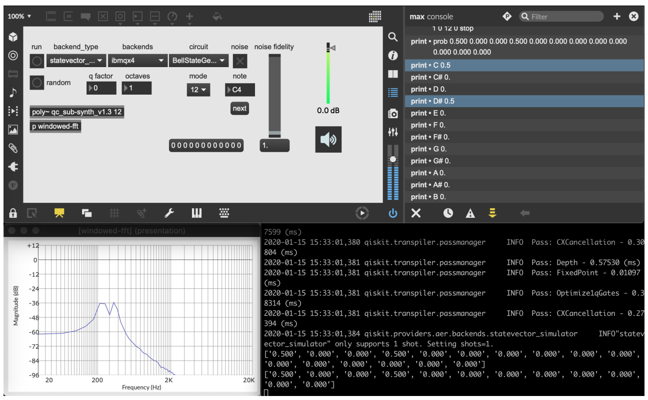
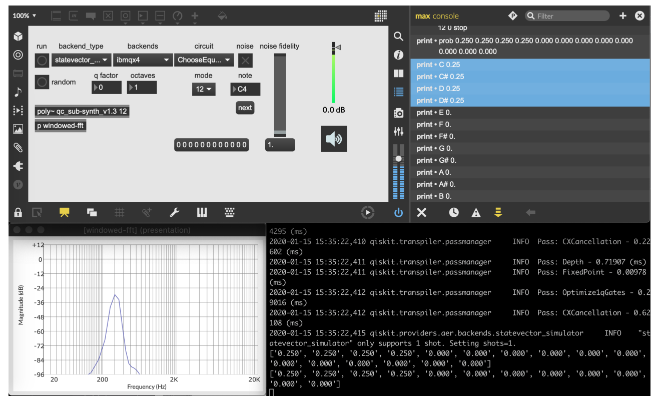


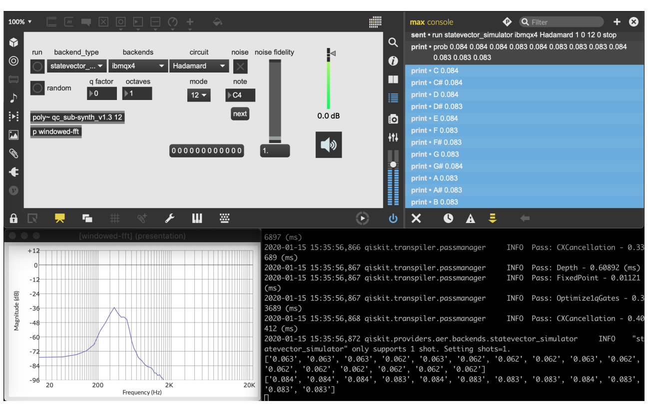
Figure 4: Quantum circuit for Bell state $\frac{|00\rangle+|11\rangle}{\sqrt{2}}$ generation.



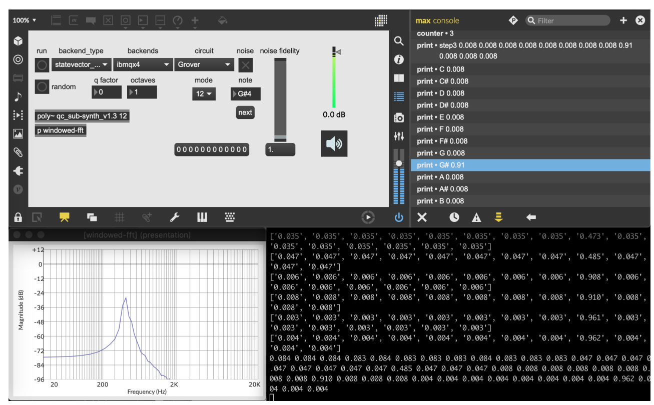
(a) *Bell state* generation mode



(b) *Equal superposition* mode



(c) *Hadamard* mode



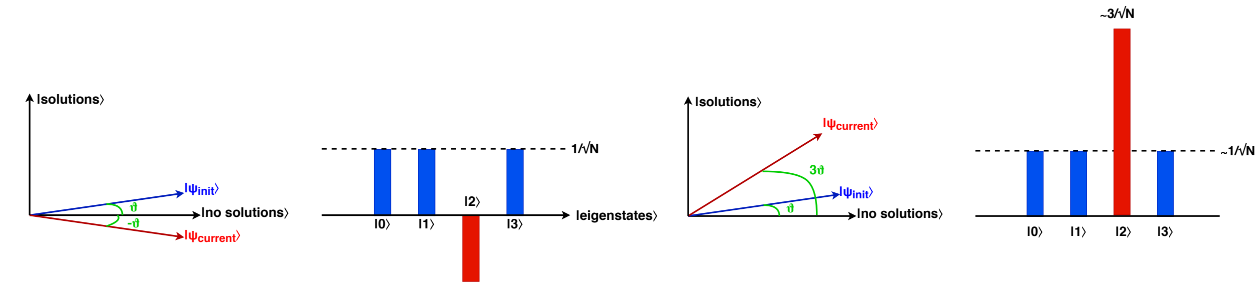
(d) Second iteration of amplitude amplification in Grover's search

Figure 5: *Quantum Synth*'s demonstrations for different quantum circuits.

4.2 Equal superposition and Hadamard gates

In both *equal superposition* and *Hadamard* modes it is possible to generate equal superposition of quantum states, i.e. they present the same probability of being measured. The main difference between them is that *equal superposition* creates an equal superposition to an arbitrary number of states (four by default) by exploiting the Qiskit's *initialize* method, while *Hadamard* exploits the quantum Hadamard gate [6]. In a single-qubit case, Hadamard gate creates an equal superposition of $|0\rangle$ and $|1\rangle$ when it is applied to a qubit in $|0\rangle$ state. In *Quantum Synth*, four Hadamard gates are employed, each one to a single qubit, in order to create an equal superposition of all the 2^4 states associated to a four-qubit system; finally twelve states among those are selected for our subtractive synthesis.

The two modes can be seen in Figures 5(b) and 5(c): it is possible to clearly observe a wider spectrum in Hadamard mode than equal superposition because of the higher number of states (and consequently frequency bands) in superposition.



(a) Phase kickback (b) Amplitude amplification with diffusion operator

Figure 6: First iteration of Grover's search algorithm.

4.3 Grover's Search

Grover's algorithm [4] helps finding an element in an unsorted database (like picking a certain card from a deck) in a computational time lower than the corresponding classical counterpart. It is one of the most famous quantum algorithms and it is used in many different applications, such as optimization procedures, pattern recognition, machine learning and testing of digital systems.

Each outcome of a quantum computer with N qubits (from 0 to $2^N - 1$) is associated to an element of the dataset, *e.g.* a card of the deck. The initial operation of this algorithm consists in initializing the quantum computer into a superposition state corresponding to the expected initial probability distribution of the dataset, so in the case of a deck with all different cards, each outcome can be measured with the same probability. The algorithm iteratively increases - according to the **amplitude amplification** procedure - the probability of measuring the desired outcome by labeling the target with a sign flip of its probability amplitude (*phase kickback*, Figure 6(a)) and then increasing the target's probability with the *diffusion operator* (Figure 6(b)).

In *Quantum Synth*, Grover's search is employed in the context of subtractive synthesis; the amplitude amplification of a quantum state is interpreted as a volume amplification of a corresponding frequency bands. Starting from an equal superposition of low-volume bands, behaving as noise, Grover's search gradually amplifies the volume of the desired band, until it is well-distinguishable among the others (see Figure 6(b) for an idea of how amplitude amplification works). It is possible to prove that three iterations are required for reaching the optimal amplitude amplification of one state among twelve, which is the number of states employed in the subtractive synthesis of *Quantum Synth*. Figure 5(d) shows the state of the quantum system at the end of the third iteration: the desired frequency band (G#4) is encoded onto the ninth state vector and its probability has been already significantly amplified (about 90%). The most distinguishable component in the sonic spectrum is at the frequency corresponding to G#4, thus proving the effectiveness of the Grover's quantum circuit.

5 CONCLUSIONS

An overview of the main components of *Quantum Synth* was presented with some brief demonstrations. New quantum circuits and synthesis techniques are already being explored and added to the infrastructure, in order to expand its usefulness for teaching quantum computing, and for music performance and composition. In addition, the characterization, in sonic terms, of the intrinsic noise of quantum computers (a critical aspect limiting the construction of more powerful and fault-tolerant hardware) has been identified as an important and challenging context of application for *Quantum Synth*. The development of ad-hoc routines for quantum hardware noise characterization could allow the use of *Quantum Synth* in the community of experimental physicists and engineers working on the design and manufacturing of quantum hardware. Lastly, besides all possible research areas that can be envisaged for *Quantum Synth*, we also believe this is a promising tool, with a potential for collaborative interdisciplinary work across technological and artistic fields.

ACKNOWLEDGMENTS

We would like to thank Mohammad Ghazi Vakili and Alberto Baiardi for helping us in the development of the *Quantum Synth*esizer. This article results from part of the work developed within the scope of a project by the first author, a project developed with the financial support of the Foundation for Science and Technology (FCT) and the European Social Fund-Operational Program Human Capital (FSE-POCH).

REFERENCES

[1] Cycling '74. 2020. *Max official website.* https://cycling74.com
[2] Héctor Abraham, AduOffei, Ismail Yunus Akhalwaya, Gadi Aleksandrowicz, Thomas Alexander, Gadi Alexandrowics, Eli Arbel, Abraham Asfaw, et al. 2019. Qiskit: An Open-source Framework for Quantum Computing. https://doi.org/10.5281/zenodo.2562110
[3] Dennis Gabor. 1947. Acoustical quanta and the theory of hearing.
[4] Lov K. Grover. 1996. A Fast Quantum Mechanical Algorithm for Database Search *(STOC '96).* ACM, New York, NY, USA, 212–219. https://doi.org/10.1145/237814.237866
[5] Omar Costa Hamido, Mohammad Ghazi Vakili, Edoardo Giusto, Alberto Baiardi, and Giovanni Amedeo Cirillo. 2019. Quantum Synth. https://github.com/menegolli/Quantum_synth
[6] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (10th ed.). Cambridge University Press, New York, NY, USA.
[7] Davide Rocchesso and Maria Mannone. 2018. Embryo of a Quantum Vocal Theory of Sound. In *Conference of Musical Informatics (CIM) Proceedings, Udine, http://cim.lim. di. unimi. it/2018_ CIM_XXII_Atti. pdf.*