

# UNIVERSITÀ DEGLI STUDI DI PADOVA

Dipartimento di Fisica e Astronomia “Galileo Galilei”

Corso di Laurea Triennale in Fisica

## Assessment of real quantum hardware with dynamical localization

Thesis supervisor

Dott.ssa Ilaria Siloi

Thesis co-supervisor

Prof. Simone Montangero

Candidate

Alberto Salvador

Academic Year 2022/2023



# Abstract

Quantum computing deeply relies on the generation of coherent dynamics and quantum entanglement. Current quantum hardware is strongly affected by various noise sources, which ultimately result in a loss of performance. As a consequence, having practical tools for evaluating hardware performance has become a crucial challenge to benchmarking the progress of quantum processors. In this thesis, we use the dynamical localization of the quantum sawtooth map, a highly sensitive coherent phenomenon, as a tool to compare real quantum machines. We first present the quantum sawtooth map, one of the simplest models where a periodically driven system exhibits dynamical localization. Then, we describe the quantum algorithm for an efficient simulation of the quantum sawtooth map on a quantum computer. Finally, we run the quantum simulation on hardware accessible through cloud quantum programming and compare performance from different quantum devices.

La computazione quantistica dipende fortemente dalla possibilità di generare dinamica coerente ed entanglement fra i qubit. Allo stato odierno i computer quantistici sono affetti da numerose fonti di rumore, le quali in definitiva risultano in una perdita prestazionale. Avere a disposizione uno strumento pratico con cui testarne l'efficienza è dunque di fondamentale importanza e il come ottenerlo si è rivelata essere una delle principali sfide per definire il progresso nello sviluppo degli hardware quantistici. In questa tesi abbiamo sfruttato il fenomeno della dynamical localization nella quantum sawtooth map, ovvero un fenomeno altamente sensibile alla presenza di stati coerenti, come strumento per comparare diversi processori quantistici. Innanzitutto presentiamo la quantum sawtooth map, uno dei modelli più semplici dove un sistema periodicamente azionato mostra la presenza di dynamical localization. Dopodichè, descriviamo l'algoritmo che permette di simulare efficacemente tale mappa su un computer quantistico. Presentiamo poi i risultati dell'implementazione di tale algoritmo su computer quantistici reali, accessibili attraverso cloud, ed infine riportiamo un confronto fra le performance ottenute fra diversi processori quantistici.



# Contents

<b>Abstract</b>	<b>iii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Quantum Computing</b>	<b>3</b>
1.1 Qubit . . . . .	3
1.1.1 Multiple qubit systems . . . . .	4
1.1.2 Mixed states and the density operator . . . . .	5
1.2 State evolution . . . . .	6
1.2.1 Non-unitary time-evolution . . . . .	6
1.3 Gates . . . . .	7
1.4 Hardware implementation . . . . .	8
1.5 Quantum circuits on real hardware . . . . .	9
<b>2 Dynamical Localization</b>	<b>13</b>
2.1 The sawtooth map . . . . .	13
2.1.1 Classical sawtooth map . . . . .	13
2.1.2 Quantum sawtooth map . . . . .	14
2.2 A quantum algorithm for the sawtooth map . . . . .	15
2.3 Dynamical localization . . . . .	17
<b>3 Results</b>	<b>21</b>
3.1 IBM quantum processors and simulators . . . . .	21
3.2 Dynamical localization on quantum hardware . . . . .	22
<b>Conclusions</b>	<b>25</b>
<b>Bibliography</b>	<b>28</b>



# Introduction

Quantum computation is a rapidly evolving field of research that promises to revolutionize our understanding of computation. It is rooted in the principles of quantum mechanics, which enable the manipulation of quantum bits, or qubits, to perform computationally hard tasks with improved efficiency [19]. The concept of quantum computation was first proposed by Richard Feynman in the 1980s as a means of simulating the behavior of quantum systems [8]. Since then, significant progress has been made in the development of quantum algorithms, quantum correcting schemes, and quantum hardware, culminating in the first demonstration of quantum supremacy by Google in 2019 [2]. However, significant challenges remain both in the development of new quantum algorithms and in the realization of quantum hardware for implementing a universal scalable quantum computer. While the Shor algorithm for factoring large numbers has been shown to be exponentially faster than classical algorithms [25], only a few of other quantum algorithms offer substantial computational gains over classical computing, and many of these algorithms need an error corrected quantum computer to properly perform. The issue of decoherence and various sources of noise hinder the amount of entanglement that can be generated in actual prototypes of quantum computers. We are in the so-called Noisy Intermediate-Scale Quantum (NISQ) era, where devices are prone to errors and the number of qubits ranges from tens to hundreds. This imposes a considerable challenge in achieving a quantum advantage in practically relevant problems such as biological processes, new material design, and chemical reactions. To gauge the progress of present devices, it is necessary to simulate less demanding, yet significant physical tasks. For these reasons, at this stage, it is unlikely that quantum computers will be able to outperform classical computers for general-purpose computations yet, they will be used to accelerate the solution of specific tasks, similar to GPU in classical computers.

At the same time, the search for the best qubit platform for quantum computing is ongoing. Various experimental platforms have shown promise as potential candidates for future quantum computers, and many are implemented in commercial devices. Notably, IBM Quantum, Google, DWave and Rigetti employ superconducting qubits technology [2] [9] [11] [21], while QuEra [14] and Pasqal [24] utilize neutral atom technology. Xanadu Quantum Technologies employs photonic quantum computers [15]. IonQ's hardware uses trapped ions in its platforms [7]. For this reason, it is essential to have comparison tools to test different quantum hardware and evaluate their performance.

In this Thesis, we investigate the dynamical localization of the quantum sawtooth map [22]. Dynamical localization is a phenomenon that characterizes the quantum behavior of classically chaotic systems, where quantum interference suppresses the diffusion in the underlying classical model, resulting in exponentially localized wave functions. More generally, localization is a common occurrence in wave physics, as it arises from the interference between scattering paths. Dynamical localization was first discovered in the quantum kicked-rotor model [4] and has since been experimentally observed in microwave ionization of Rydberg atoms [13], atom-optic systems [16] [18] [6], and nuclear magnetic resonance [10]. Recently, it has been established that in a quantum computer simulating dynamical localization, the degree of entanglement is related to the localization length of the simulated system [17]. Here, we implement the quantum sawtooth map as an efficient quantum algorithm and reproduce its dynamics both on simulators and real devices. In order to do that, we used an algorithm that allows reproducing its dynamics on simulators and real devices available through "IBM Quantum Experience", the cloud-based service for quantum computation of IBM. By comparing the results obtained from different devices, we compare their ability to detect dynamical localization. This can serve as a benchmark for evaluating the ability of the device to maintain coherence and to perform under different

types of perturbations, ultimately providing a powerful tool for comparing different quantum machines.

In the first chapter, we introduce the basic concepts in quantum information and quantum computation. We also present a description of the IBM architectures based on superconducting qubits, and how this relates to the problem of transpilation of an ideal circuit to a circuit that matches the hardware connectivity. In the second chapter, we focus on the quantum sawtooth map, a dynamical system that is a prototype for the study of quantum chaos. Firstly we introduce the classical version of it and its dynamic behavior. Then we discuss its extension to the quantum case. After that, we explain how to simulate its dynamics on quantum devices, providing an algorithm easily implementable on the computers we tested. The third chapter is focused on the quantum devices used for simulations and the results obtained from them. We simulate dynamical localization with  $n \leq 10$  qubits on real quantum hardware and provide a comparison with the results obtained in the noiseless and noise-simulated cases.

# Chapter 1

## Quantum Computing

In this Chapter, we review the main concepts of Quantum Information and Computation. Firstly, we introduce the qubit, the fundamental unit of information in the quantum realm, and how to manipulate qubits by using a sequence of quantum gates. Then, we consider a real hardware implementation with superconducting circuits with its typical source of noise. Finally, we show how the connectivity of a real quantum processing unit affects an ideal quantum circuit.

### 1.1 Qubit

The fundamental unit for carrying information in quantum computation is the qubit. Analogously to the classical bit, it can assume only two values, which can be referred to as 0 and 1. These are associated with two distinguished states of a quantum system. In general, to a quantum system it is associated an Hilbert space  $\mathcal{H}$  and all the possible physical states that it can assume are represented by ray vectors defined as:

$$\hat{\psi} = \{\lambda\psi \quad \text{with} \quad \lambda \in \mathbb{C}, \lambda \neq 0 \wedge \psi \in \mathcal{H}\}. \quad (1.1)$$

For instance, we consider a qubit associated with a system with spin 1/2 and we want to characterize its quantum states through the z-component of the spin observable. In quantum mechanics, each observable is associated with a self-adjoint operator in  $\mathcal{H}$ . The possible measurements of this observable are its operator eigenvalues, and that the Hilbert space can be decomposed in the eigenspaces associated to this operator, namely the spaces spanned by its eigenvectors. In this particular case we then have the  $\hat{S}_z$  operator, whose eigenkets  $|0\rangle, |1\rangle \in \mathcal{H}$  span a 2-dimensional space. These vectors compose the so-called *computational basis*. The eigenvalues associated to these eigenkets are  $\{+\frac{\hbar}{2}, -\frac{\hbar}{2}\}$ :

$$\begin{aligned} \hat{S}_z|0\rangle &= \frac{\hbar}{2}|0\rangle \\ \hat{S}_z|1\rangle &= -\frac{\hbar}{2}|1\rangle \end{aligned} \quad (1.2)$$

The operator  $\hat{S}_z$  can be written as  $\frac{\hbar}{2}\hat{\sigma}_z$ , where  $\hat{\sigma}_z$  denotes the z-component of the *Pauli operator*.  $|0\rangle$  and  $|1\rangle$  are eigenvector for  $\hat{\sigma}_z$  as well, with eigenvalues 1 and -1 respectively. The general state of the qubit  $|\psi\rangle$  would be a superposition of these two states:

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle, \quad (1.3)$$

where  $|c_0|^2 + |c_1|^2 = 1$ . The probability of measuring the value  $+\frac{\hbar}{2}$  is equal to:

$$w(+\hbar/2) = \frac{|\langle 0|\psi\rangle|^2}{\|\psi\|^2} = |c_0|^2. \quad (1.4)$$

When a measurement is performed, the physical state after the measurement “collapses” to its projection onto the eigenspace associated with the eigenvalue measured. If  $|\psi(0)\rangle$  is the state of the qubit

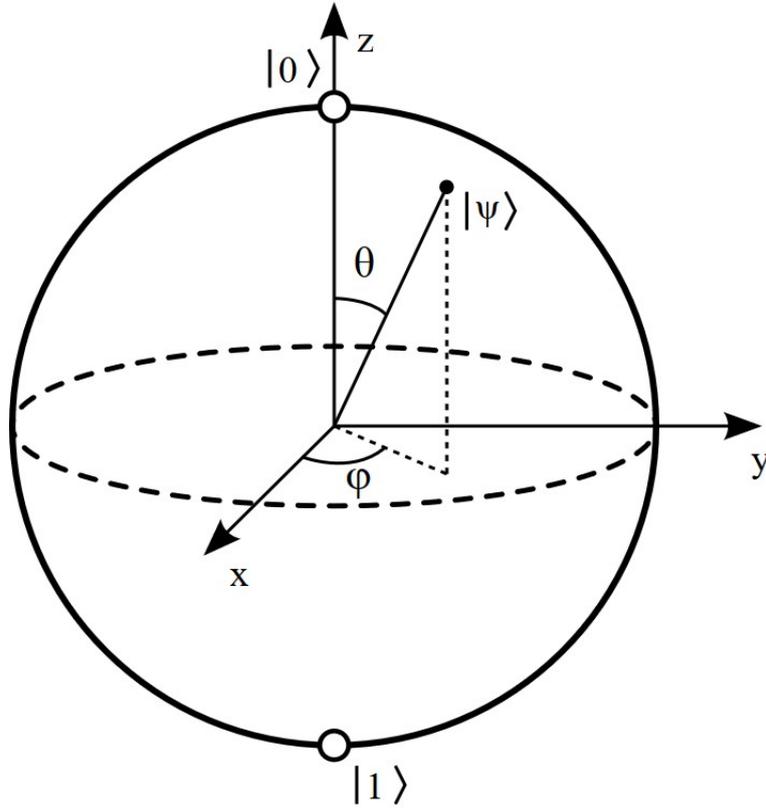


Figure 1.1: The Bloch Sphere. As shown in Equation (1.5) the  $\hat{z}$  unit vector represent the  $|0\rangle$  state and  $-\hat{z}$  the  $|1\rangle$  state. The angles  $\theta$  and  $\phi$  identify the qubit state. Image from: [https://upload.wikimedia.org/wikipedia/commons/6/6b/Bloch\\_sphere.svg](https://upload.wikimedia.org/wikipedia/commons/6/6b/Bloch_sphere.svg)

immediately before a measurement that results in  $+\frac{\hbar}{2}$ , then the state immediately after it would be  $|\psi(0^+)\rangle = |0\rangle$ . This guarantees that a measurement performed immediately after will certainly result in  $+\frac{\hbar}{2}$  again.

A useful way of representing the state of a qubit is the so-called *Bloch Sphere*: a unitary sphere in a 3-dimensional space. As represented in Fig. 1.1, the qubit state is a vector pointing to the surface of the sphere and is uniquely identified by the angles  $\theta$  and  $\phi$ , where  $0 \leq \theta \leq \pi$  and  $0 \leq \phi < 2\pi$ . We have:

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + \sin \frac{\theta}{2} e^{i\phi} |1\rangle. \quad (1.5)$$

From Eq. 1.5 is evident that using a coherent superposition opens the possibility to encode an infinite amount of information in a qubit state. However, each measurement will produce only one outcome,  $|0\rangle$  or  $|1\rangle$ , as for a classical bit.

### 1.1.1 Multiple qubit systems

A system composed of two or more distinct physical systems, is said to be a composite quantum system. The state space of such a system is the tensor product of the state spaces of its subsystems.

$$\mathcal{H} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_N. \quad (1.6)$$

If we consider  $N$  multiple systems prepared in states labeled  $|\psi_i\rangle$ , then the N-body state will be:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_N\rangle. \quad (1.7)$$

An operator that acts only on a subsystem without affecting the others is said *local operator*. Formally, let  $\mathcal{H}$  be the Hilbert space associated with a composite quantum system that can be divided into two subsystems, with corresponding Hilbert spaces  $\mathcal{H}_1$  and  $\mathcal{H}_2$ . A local operator acting in the second subspace can be written as:

$$A = \mathbb{I}_1 \otimes A_2, \quad (1.8)$$

where  $\mathbb{I}_1$  is the identity operator in  $\mathcal{H}_1$ , so that this subsystem remains unaffected, and  $A_2$  is an operator on  $\mathcal{H}_2$ .

One of the most interesting and puzzling phenomena associated with composite quantum systems is *quantum entanglement*. As an example, we consider the state:

$$|\phi\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}}, \quad (1.9)$$

called ‘‘Bell couple’’. This state has the remarkable propriety that there are no single qubit states  $|\phi_1\rangle$  and  $|\phi_2\rangle$  in  $\mathcal{H}$  such that  $|\phi\rangle = |\phi_1\rangle \otimes |\phi_2\rangle$ . When a state of a composite system cannot be written as a product of states of its component, the state is said to be *entangled*, in contrast to *separable* ones. Entangled states contain purely quantum correlations between the subsystems that cannot be explained in terms of classical physics. For instance, by performing a measurement on one subsystem we can instantly obtain information on the other subsystem without actually measuring it. This correlation does not violate special relativity since the information obtained must reach the second subsystem via classical communication. It is worth noting that entanglement is considered to be a resource for establishing the power of quantum computation compared to classical computation. This is because entangled quantum states can exhibit exponentially greater complexity than classical ones, and then can be used to perform certain computational tasks much faster than classical computers.

### 1.1.2 Mixed states and the density operator

When we have access to complete information about a quantum state, it is defined as *pure state*. In the majority of cases, the quantum state of a qubit is coupled to the external environment so that we can only have access to a portion of the information that it carries. It is then impossible to write the state in the form of Eq. 1.5, and we need to introduce a new formalism, that extends the concept of pure states to statistical mixtures of pure states, also known as *mixed states*. In order to describe them, the idea of *density operator* was introduced. The density operator  $\rho$  is the density operator associated with some ensemble  $\{p_i, |\psi_i\rangle\}$  if and only if it has trace equal to 1 and it is a positive hermitian operator [19]. It is defined as:

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|. \quad (1.10)$$

All the known results valid for pure states could then be rewritten in the density operator picture. The expectation value of  $\hat{A}$  is:

$$\langle\hat{A}\rangle_\rho = \text{tr}(\rho\hat{A}). \quad (1.11)$$

If the system is described by the state  $\rho$  (normalized), the probability that a measurement of a generic observable  $\hat{A}$  gives, as a result, its eigenvalue  $a$  is given by:

$$W(a) = \text{tr}(\hat{A}_a^\dagger \hat{A}_a \rho). \quad (1.12)$$

When we consider a composite system, whose subsystems are numbered from 1 to  $N$  and with  $\rho_i$  the state of the generic system numbered  $i$ , the joint state of the total system is given by the tensor product of the subsystem’s states, explicitly  $\rho_1 \otimes \rho_2 \otimes \dots \otimes \rho_N$ .

## 1.2 State evolution

The evolution of a pure state is given by the Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = H|\psi\rangle, \quad (1.13)$$

where  $H$  is the Hamiltonian of the system.

Solving this equation allows us to determine the explicit form of the time-evolution unitary operator  $U(t)$ , which acts on the states of the system as  $|\psi(t)\rangle = U(t)|\psi\rangle$ . For the case of a time-independent Hamiltonian:

$$U(t) = e^{-\frac{i}{\hbar}Ht}. \quad (1.14)$$

Thanks to the spectral theorem, the set  $\{|n\rangle\}$  of eigenfunction of  $H$  with eigenvalues  $\{E_n\}$ , forms an orthonormal base for  $\mathcal{H}$ . This allows us to decompose a generic state as  $|\psi\rangle = \sum_n c_n |n\rangle$ , and ultimately to write the action of the time-evolution operator onto the state  $|\psi\rangle$  as:

$$|\psi(t)\rangle = \sum_n c_n e^{-\frac{i}{\hbar}E_n t} |n\rangle. \quad (1.15)$$

When considering the time evolution of a system described by the density operator  $\rho(t=0)$ , in turn, we have:

$$\rho(t) = U\rho(0)U^\dagger = \sum_i p_i U|\psi_i\rangle\langle\psi_i|U^\dagger. \quad (1.16)$$

### 1.2.1 Non-unitary time-evolution

The type of evolution described before is characteristic of unitary dynamics. When we deal with real quantum systems we must consider the interaction with the environment. As a consequence non-unitary effects are involved in the dynamics of the system. Since we cannot express such processes as a unitary evolution, we must adopt the formalism of *Kraus operators*. As an example, we now discuss the evolution of a subsystem in a 2-body system, where  $\rho = \rho_1 \otimes \rho_2$  is the generic state given by the tensor product of  $\rho_1$ , the state of the subsystem 1, and  $\rho_2$ , the state of the subsystem 2. The evolution of the first subsystem is characterized by the Kraus superoperator  $E_k$  so that:

$$\rho_1 \rightarrow \rho'_1 = \text{tr}_2(\rho(t)) = \sum_{k=1}^{\dim \mathcal{H}_2} E_k \rho_1 E_k^\dagger \quad \text{with} \quad \sum_k E_k^\dagger E_k = \mathbb{I}. \quad (1.17)$$

In general, we define a map  $S : \rho_1 \rightarrow \rho'_1$  with the property of preserving the Hermiticity, the trace, and the positivity of  $\rho$ . In addition to that, its inverse exists if and only if  $S$  is unitary and if  $\rho' = S_1(\rho)$  and  $\rho'' = S_2(\rho')$  then  $\rho'' = S_2(S_1(\rho))$

The interaction between quantum systems and their environments is a fundamental source of non-unitary effects, which can lead to the degradation of quantum information and the loss of coherence between states. To model this interaction, *quantum channels* are commonly used. In the following, we provide some of the main channels encountered when dealing with practical applications.

The *amplitude damping* channel describes the loss of energy from a quantum system to its environment. It is characterized by a time constant  $T_1$ , called *relaxation time*, which measures the rate of the relaxation. The *phase damping* channel, in turn, describes the loss of coherence between different states of the system, transforming the state  $\frac{1}{2}(|0\rangle + |1\rangle)(\langle 0| + \langle 1|)$  into the state  $\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)$ . This arises due to fluctuations in the phase of the system's wave function caused by interactions with the environment. It is characterized by a time scale denoted  $T_2$ , called *coherence time*. The *depolarizing channel* is another common type of quantum channel, and it describes the randomization of the system's state. This acts on the system by randomly selecting one of several possible unitary operations, effectively "erasing" the original state. In particular, it decreases the projections of the state over the axis x,y, and z over time, progressively causing the decay into the state  $|0\rangle\langle 0| + |1\rangle\langle 1|$ .

In addition, there are two types of channels that describe the flipping of individual qubits: the *bit flip* channel, which flips the state of a qubit from 0 to 1 or vice versa, and the *phase flip* channel, which introduces a phase shift of  $\pi$  around the Z-axis of the Bloch sphere.

A thorough understanding of the effects of these channels is essential for developing strategies to protect quantum systems from decoherence and other non-unitary effects. In particular, the characterization of the time constants  $T_1$  and  $T_2$  is critical for the design and optimization of quantum error correction codes and other techniques for mitigating the effects of noise and decoherence in quantum systems.

### 1.3 Gates

The circuit model of quantum computation is a direct generalization of classical circuits to the quantum realm. The basic mechanism of information processing in these models is unitary evolution, where quantum information is processed without measurement until the end. The results obtained from measuring then allow us to convert quantum information into classical one, in order to read out classical answers [5].

A classical computer is a physical object capable of calculating functions of the form  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , where  $n$  represents the number of bits involved. Any function  $f$  can be expressed as a finite series of operations, called *gates*. The set of logical gates necessary to codify the computation of any function  $f$  is called *universal set*. A commonly used example of a universal gate set is  $\{AND, OR, NOT, COPY\}$ .

Similarly, to manipulate the state of a qubit, *quantum gates* are defined. These are represented as unitary operators acting on qubits to preserve the normalization of probability amplitudes and to guarantee the reversibility of quantum operations.

Similarly to the classical case, any universal set of quantum gates counts single a two-qubit gates. We discuss the action of some important single-qubit and two-qubit gates in the computational basis. Each gate is represented as a 2x2 matrix that operates on the state of a qubit. The first set of gates we discuss is single-qubit gates.

- Hadamard gate (H):

It is a fundamental gate in quantum computing that generates superposition states. It can be visualized as a  $\pi/2$  rotation of the state vector in the (x,z) plane around the y-axis. The Hadamard gate acts on the computational basis as follows:  $[|0\rangle, |1\rangle] \rightarrow \left[ \frac{|0\rangle+|1\rangle}{\sqrt{2}}, \frac{|0\rangle-|1\rangle}{\sqrt{2}} \right]$

The matrix representation of the Hadamard gate is:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \quad (1.18)$$

- Phase shift ( $P(\phi)$ ):

It applies a relative phase shift to a qubit's state. It can be visualized as a rotation of the state around the z-axis. The phase shift gate adds a relative phase to the qubit's state, taking the state  $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$  to  $\frac{|0\rangle+e^{i\phi}|1\rangle}{\sqrt{2}}$ .

The matrix representation of the phase shift gate is:

$$R_z(\Phi) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\Phi} \end{pmatrix}. \quad (1.19)$$

- Not (X):

It is a fundamental gate that flips the state of a qubit. It can be visualized as a  $\pi$  rotation of the state on the x-axis. The Not gate acts on the computational basis as follows:  $[|0\rangle, |1\rangle] \rightarrow [|1\rangle, |0\rangle]$ .

The matrix representation of the Not gate is:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \quad (1.20)$$

The second set of gates we discuss is two-qubit gates.

- Control Not (CNOT):

It applies a Not gate to the target qubit if and only if the control qubit is in the  $|1\rangle$  state.

The matrix representation of the CNOT gate is:

$$CX = \mathbb{I}_{1,2 \times 2} \otimes X_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (1.21)$$

Here,  $\mathbb{I}_{2 \times 2}$  is the 2x2 identity matrix, and  $\mathbb{O}_{2 \times 2}$  is the 2x2 zero matrix.

- Control Phase (CPHASE):

It applies a phase shift to the target qubit if and only if the control qubit is in the  $|1\rangle$  state.

The matrix representation of the CPHASE gate is:

$$CPHASE(\theta) = \mathbb{I}_{1,2 \times 2} \otimes R_z(\theta) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & e^{i\theta} \end{pmatrix}. \quad (1.22)$$

- SWAP:

It exchanges the state of two qubits, explicitly:

$$SWAP(|j\rangle \otimes |k\rangle) = |k\rangle \otimes |j\rangle. \quad (1.23)$$

It has been shown that a universal set of quantum gates is  $\{H, R_z(\pi/4), CNOT\}$ .

A *quantum circuit* is composed of a combination of quantum gates applied to a multi-qubit system, usually initialized in the state  $|0\rangle$ . Any quantum circuit ends with a measurement. In context, a *quantum algorithm* is a set instruction, provided as a specific sequence of gates, that specifies how to manipulate and measure quantum states in order to solve a particular problem. The efficiency of a quantum algorithm is measured in terms of the number of gates and qubits required, as well as the number of measurements needed to obtain the desired output. We define the *depth* of a circuit as the number of quantum gates it contains.

## 1.4 Hardware implementation

In this Section, we examine specific physical hardware developed by IBM, which we will use to produce the results for this Thesis, as detailed in Chapter 3. IBM hardware uses superconducting circuits and qubits, therefore the name *superconducting qubits*.

A fundamental element in any superconducting qubit architecture is the *Josephson junction*, which consists of two superconducting electrodes separated by a thin barrier made up of an insulating material. One can compute the density of the superconducting charge carrier component and see that, due to the presence of a phase that gives rise to interference phenomena, electron pairs are formed, called *Cooper pairs*, of charge  $2e$  and integer spin (0), thus behaving like bosons [1].

In a Josephson junction, the separation barrier is sufficiently thin to allow a weak coupling between the two electrodes, leading to the tunneling of Cooper pairs across the insulator. When cooled down to sufficiently low temperatures, Josephson junction circuits exhibit quantum properties, which is the

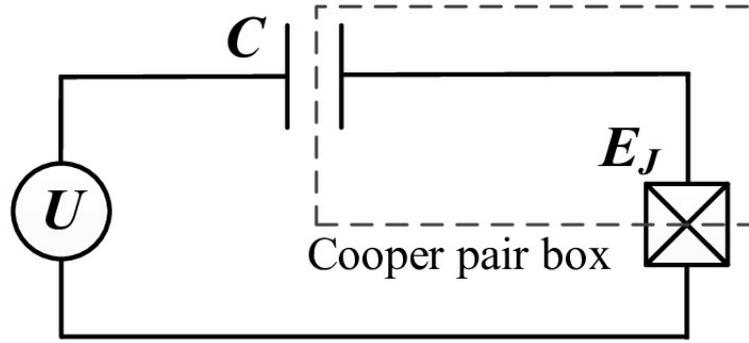


Figure 1.2: Circuit diagram of the Cooper pair box. Charges tunnel onto the island through the Josephson junction, represented by the crossed square. Image adapted from [26]

basis for turning superconducting circuits into qubits. A particular circuit of this type is the single Cooper pair box (CPB), which couples a small superconducting island through a Josephson junction to a gate electrode. In Figure 1.2 the related circuit diagram. The peculiarity of this system is that it can be treated effectively as a two level system, whose associated states can be assumed as the two characteristic states  $|0\rangle$  and  $|1\rangle$  of a qubit as seen in Section 1.1. When a shunt capacitance is added between the gate and the island, we talk about *transmon*, the actual types of qubits implemented in the IBM’s quantum computer we investigated and present in Section 3.1. Increasing the capacitance of the island lowers the charging energy, and one can see that this ultimately leads to a mitigation of the problem of charge noise dephasing, improving coherence lifetime.

Superconducting qubits are subject to various sources of noise that can cause errors in their operations, we now describe some of them as a point of reference. Charge noise is caused by random variations in the charge on the surface of the qubit. Another example is the resonator noise, which results from noise in the coupling resonators used to control and read out the qubits. In addition to them, we can have noise that arises from defects in the materials used to fabricate the qubits and crosstalk. This phenomenon occurs when qubits are placed close to each other, leading to unwanted interactions between them.

One way to implement gates acting on single superconducting qubits is by applying microwave Rabi pulses by means of magnetic flux to a superconducting ring. Each quantum gate applied to a qubit causes a rotation of the qubit state around a specific axis on the Bloch sphere. However, due to imperfections in the physical system, there is always some degree of error or deviation in the angle of rotation. As a result, the gate may not produce the intended final state of the system, and we refer to this error as the gate failure probability. In a quantum circuit, the total probability of a failure is the product of the errors associated with each gate.

We stress that, for the decoherence phenomena seen in Section 1.2, when dealing with real implementations of quantum hardware, in addition to the gate-application error described above, we must consider that each qubit has a specific gate failure probability. This is the probability that a gate applied to the qubit may give as output an unexpected result because the state of the qubit is not the one predicted due to noise effects in the interaction with the environment.

## 1.5 Quantum circuits on real hardware

When implementing quantum circuits on real hardware, specifically in the case of superconducting qubits, it is essential to consider that the qubits are arranged in specific spatial positions that define the computer’s *connectivity*. Since these positions are fixed, the physical architecture allows for only a limited set of qubit pairs to interact with each other. In other words, applying a two-qubit gate to a generic pair of qubits is not straightforward. Accommodating this constraint can be achieved in a few ways. A common choice consists in the utilization of SWAP operations: by inserting chains of SWAP gates, logical qubit states can be routed throughout the physical system, allowing the locality constraint to be satisfied for all the two-qubit gates of the algorithm. This clearly increases the circuit

depth, which increases the error rate of the circuit when run on real quantum hardware.

Another important aspect that must be considered is that different quantum computers can possess different sets of fundamental gates. The choice of the fundamental gates associated with a computer depends on the particular qubits architecture. The availability and performance of these gates impact the efficiency and effectiveness of quantum algorithms and simulations, thus knowing that is fundamental for optimizing the performance.

`Qiskit` is the software developed by IBM for simulating quantum algorithms and running them on real devices through cloud. In order to ensure that the ideal circuit can be implemented on the real device, `Qiskit` performs all the transformations needed through a process called *transpilation*. This includes the before-mentioned fix for limited connectivity as well as the decomposition of common gates into sequences of  $U_2$  and CNOT gates (the actual gates typically available for IBM's computers), where  $U_2$  is a single-qubit rotation about the X+Z axis, whose matrix representation in the computational basis is:

$$U_2(\Phi, \lambda) = \frac{1}{2} \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\Phi} & e^{i(\Phi+\lambda)} \end{pmatrix} \quad (1.24)$$

For instance let us consider an architecture such as the one depicted in Figure 1.3, which represents the connectivity graph of IBM's computer named `ibmq_guadalupe`, and the circuit depicted in figure 1.4(a), composed by a single qubit gate H acting on  $q_0$  and a two qubits gate CNOT acting on two physically unconnected qubits, namely  $q_0$  and  $q_2$ . After applying the transpilation process we obtain the circuit in figure 1.4(b), where we can see that `Qiskit` added three intermediate CNOT gates between qubits  $q_1$  and  $q_2$ . These three gates swap the states of the two qubits they act on. Theoretically, the resulting entangled state of qubits  $q_0$  and  $q_2$  in the simulator is the same as the entangled state of qubits  $q_0$  and  $q_1$  in the physical implementation. In practice, however, it is expected to be slightly different due to errors.

In summary, `Qiskit's` transpiler changes the circuit in a way that allows for the physical limitations of the quantum computer to be overcome while leaving the algorithm theoretically unaffected, but it also introduces an unruly amount of CNOT gates into the circuit, which is subject to higher error rates than single-qubit gates.

We conclude by saying that since there exist different quantum computers with different connectivity and fundamental gates, physicists came up with the idea of a general parameter used to evaluate the performance of quantum computers, which ultimately allows the comparison between all sorts of quantum computers. This metric is named *quantum volume* and is defined as:

$$VQ = 2n \cdot f, \quad (1.25)$$

where  $n$  is the number of qubits and  $f$  is the depth of the circuit. By taking into account both the number of qubits in a quantum computer and the quality of its quantum gates, we can say that it measures the maximum size of a random quantum circuit that a quantum computer can successfully execute. A high Quantum Volume indicates that it can handle a large number of qubits and execute complex quantum circuits with a high level of accuracy and fidelity.

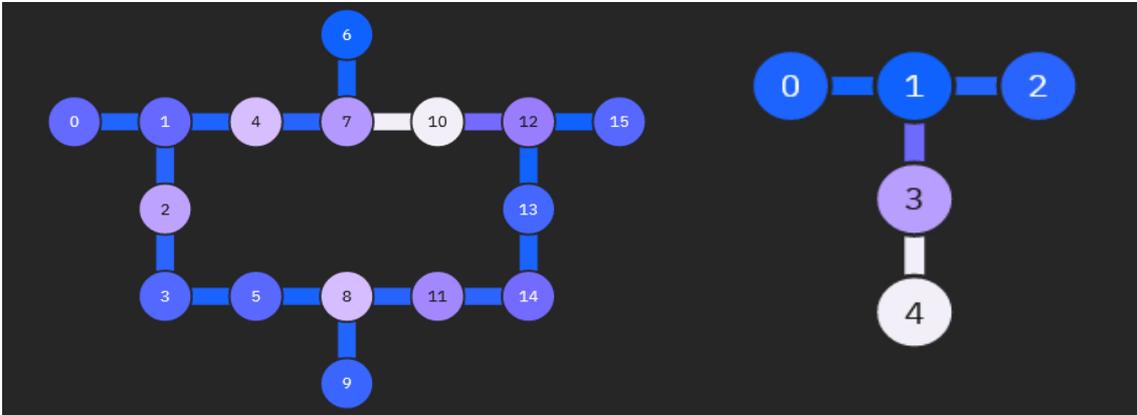


Figure 1.3: Connectivity graphs of `ibmq_guadalupe` (on the left) and `ibmq_lima` (on the right). Physical qubits are represented by the nodes and the edges correspond to the possible locations of two-qubit gates. A darker color represents a smaller error, respectively the readout assignment error for the qubit and the CNOT error between two qubits. The right one is called a linear arrangement while the one on the left is called a square grid arrangement. <https://quantum-computing.ibm.com/services/resources?tab=systems>

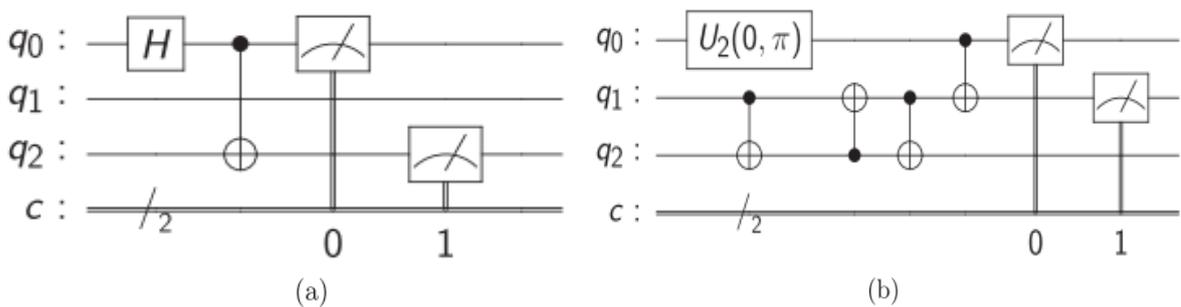


Figure 1.4: Effect of transpilation on several representative circuits. (a) Base and (b) transpiled circuit that entangles two qubits that are not physically connected directly. After each time the circuit is run, the qubits are measured. The symbols  $q_0$  and  $c_0$  are the names of a qubit and a classical bit, respectively. In this and all figures in this paper, qubits are initialized to the  $|0\rangle$  state, and classical bits are initialized to 0. Credit: [12]



## Chapter 2

# Dynamical Localization

In this chapter, we discuss a possible implementation of a quantum algorithm that allows us to simulate the dynamics of the so-called quantum sawtooth map. In particular, we describe the phenomenon of dynamical localization, which we will exploit in the next chapter in order to compare the performances of real quantum devices.

### 2.1 The sawtooth map

The sawtooth map is a prototype model in the studies of the dynamics of quantum and classical systems as it exhibits a wide variety of physical phenomena, from chaos to integrability, from normal to anomalous diffusion of momentum. Here we are specifically interested in the phenomenon named *dynamical localization*, which we will define in the next sections. We start the description from the classical perspective and then move to the quantum one.

#### 2.1.1 Classical sawtooth map

The sawtooth map describes a periodically driven dynamical system, whose dynamics is related to the Hamiltonian:

$$H(\theta, I; \tau) = \frac{I^2}{2} + V(\theta) \sum_{j=-\infty}^{+\infty} \delta(\tau - jT), \quad (2.1)$$

where  $(I, \theta)$  are conjugate momentum-angle variable,  $\theta$  ranges in  $[0, 2\pi)$ . The first term represents the kinetic energy of a free particle moving on a circle. For our purposes, the potential energy is defined as  $V(\theta) = -\frac{k}{2}(\theta - \pi)^2$ . It is associated with the force  $F(\theta) = -\frac{dV(\theta)}{d\theta}$ , which acts on the particle instantaneously every period  $T$ , therefore describing a dynamics *periodically kicked*, with kick strength  $k$ . From Figure 2.1 it is obvious the origin of the name of this map, since the shape of the  $F(\theta)$  graph recalls a sawtooth.

The corresponding Hamiltonian equations are:

$$\begin{cases} \dot{I} = -\frac{\partial H}{\partial \theta} = -F(\theta) \sum_{j=-\infty}^{+\infty} \delta(\tau - jT) \\ \dot{\theta} = \frac{\partial H}{\partial I} = I \end{cases}. \quad (2.2)$$

Equations (2.2) are easily integrable over time. Considering the evolution from time  $lT^-$  (prior to the  $l$ -th kick) to time  $(l+1)T^-$  (prior to the  $(l+1)$ -th kick) we have:

$$\begin{cases} I_{l+1} = I_l + F(\theta) = I_l \\ \theta_{l+1} = \theta_l + TI_{l+1} \end{cases}. \quad (2.3)$$

We rewrite equations (2.4) rescaling  $I \rightarrow J = TI$ , so that:

$$\begin{cases} J_{l+1} = J_l + K(\theta - \pi) \\ \theta_{l+1} = \theta_l + J_{l+1} \end{cases}, \quad (2.4)$$

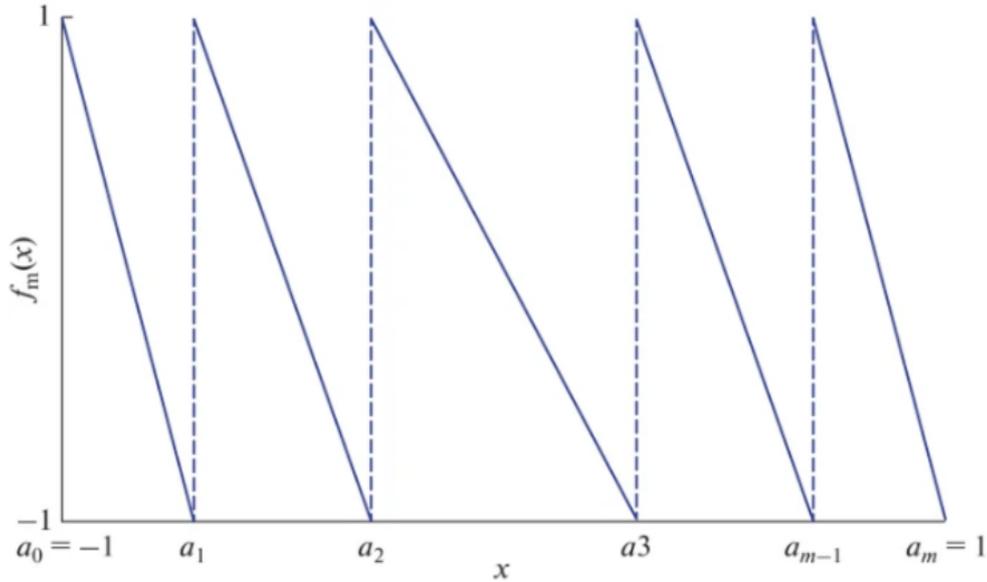


Figure 2.1:  $m$ -Branch negative for the sawtooth map. Credit: A. Sahnoune, D. Berkani/ Allerton Press [23]

where  $K = kT$  is a parameter that characterizes the classical dynamics of the quantum sawtooth map. From the periodicity of the position variable, we have that the cylindrical phase space can be represented on a torus by truncating the momentum space to length  $\frac{2\pi L}{T}$  and applying a periodic boundary condition.

Studying the dependence of the solutions of (2.4) on initial conditions, in particular for a fixed initial momentum  $J = 0$  and a randomized initial position, it is possible to demonstrate that the system is:

- *Stable* if  $-4 \leq K \leq 0$

The phase space has a complex structure of elliptic islands, and one can observe anomalous diffusion characterized by equation  $\langle (\Delta J(t))^2 \rangle \propto t^\alpha$  with  $\alpha \neq 1$

- *Chaotic* if  $K < -4 \cup K > 0$

This is due to the term  $K(\theta - \pi)$  in equation (2.4), which gives a kick to the momentum at each iteration of the map with a strength that can be approximated as a quasi-random sequence, making the motion along the momentum direction in practice indistinguishable from a random walk. This leads to diffusive broadening along the momentum dimension.

We define the *diffusion coefficient* as:

$$D = \lim_{t \rightarrow \infty} \frac{\langle (\Delta J(t))^2 \rangle}{t}, \quad (2.5)$$

where  $\Delta J(t) = J - \langle J \rangle$  and  $\langle \dots \rangle$  denotes the average over an ensemble of trajectories.

It can be shown that, from a heuristic point of view,  $\langle (\Delta J(t))^2 \rangle \approx Dn$ , where  $D \approx (\pi^2/3)k^2$  for the sawtooth map [3]. In other words, as the map is iterated, the diffusion continues indefinitely, tending to a uniform probability distribution.

### 2.1.2 Quantum sawtooth map

The quantum version of the sawtooth map can be obtained by means of the usual quantization rules, namely  $\theta \rightarrow \hat{\theta}$  and  $I \rightarrow \hat{J} = -i\frac{\partial}{\partial \theta}$ , with commutation rule:

$$[\hat{\theta}, \hat{J}] = T[\hat{\theta}, \hat{I}] = iT = i\hbar_{eff}, \quad (2.6)$$

where we used  $\hbar = 1$ . Since we have a time-dependent Hamiltonian, in general, at different times it does not commute. For this reason, a spectral representation of the evolution operator cannot be derived. In the case of periodically driven systems, Floquet theory can be applied [20]. This has the

great advantage that allows studying them with techniques similar to those used in time-independent ones, by replacing the Hamiltonian operator with the so-called *Floquet operator*. This is simply the propagator  $\hat{U}(t, t + T)$  over one period, defined as:

$$\hat{U}(t, t + T) = \mathcal{T} e^{-i \int_{lT^-}^{(l+1)T^-} H(\hat{\theta}, \hat{I}; \tau) d\tau}, \quad (2.7)$$

where  $\mathcal{T}$  indicates that the ordered product is taken in the exponential. It can be seen that, if it is computed at different times, it does not change thanks to the periodicity of the Hamiltonian.

In summary, for one iteration of the sawtooth map, we have:

$$\psi_{l+1} = \hat{U} \psi_l = \exp \left[ -i \int_{lT^-}^{(l+1)T^-} H(\hat{\theta}, \hat{I}; \tau) d\tau \right]. \quad (2.8)$$

Since the potential is switched on only at discrete times  $lT$ , the contribution from the angular coordinate and the momentum can be factorized as:

$$\psi_{t+1} = U_t U_k \psi_t \quad \text{with} \quad U_T = e^{-\frac{iTj^2}{2}} \quad U_k = e^{i \frac{k(\hat{\theta} - \pi \hat{I})^2}{2}}. \quad (2.9)$$

The quantum evolution depends not on a single parameter as the classical one, but on  $k$  and  $T$  separately, as it can be seen from (2.9). The classical limit  $\hbar_{eff} \rightarrow 0$  is obtained by taking  $k \rightarrow \infty$  and  $T \rightarrow 0$ , while keeping  $K = kT$  constant [3].

We set  $j$  and  $\theta_m$  the eigenvalues related to the momentum and position basis so that:

$$\begin{cases} \hat{J}|j\rangle = j|j\rangle \\ \hat{\theta}|\theta_m\rangle = \theta_m|\theta_m\rangle \end{cases}, \quad (2.10)$$

where  $-N/2 \leq j < N/2$  and  $0 \leq m < N$ . If  $n$  is the number of qubits, we have  $N = 2^n$  possible states in the momentum basis, therefore  $N = \frac{2\pi L}{T}$  for what we said about the toroidal phase space in Section 2.1.1.

Since we will deal with quantum systems composed of  $n$  interacting qubits, it is useful to write each eigenket as a function of the quantum states of these types of systems. For this reason, we define a binary notation where the 0s and 1s are associated with the single states of the qubits that the system.

We have:

$$\begin{aligned} \theta &= 2\pi \sum_{j=1}^n \alpha_j 2^{-j} \rightarrow |\theta\rangle = |\alpha_n \alpha_{n-1} \dots \alpha_1\rangle \\ J &= N \sum_{j=1}^n \alpha_j 2^{-j} - \frac{N}{2} \rightarrow |J\rangle = |\beta_n \beta_{n-1} \dots \beta_1\rangle, \end{aligned} \quad (2.11)$$

where  $\alpha_j, \beta_j \in \{0, 1\} \quad \forall j$ .

## 2.2 A quantum algorithm for the sawtooth map

Using quantum computers we can efficiently simulate the time-evolution of a system that follows equations (2.9). This is possible thanks to the algorithm described in [22] [3] [10] and that we will briefly discuss in the following. This algorithm is based on the forward/backward quantum Fourier transform (QFT), a well-known quantum algorithm that has the action of toggling between the position and momentum basis representation. It ultimately requires only phase-shift and control phase-shift gates, so that it is easily implementable on superconducting-based quantum computers as the IBM ones that we used.

One map iteration requires the following steps:

1. After the system is initialized in the momentum basis state, we apply QFT, changing from the  $J$  to the  $\theta$  representation. It ultimately requires  $n$  Hadamard gates and  $\frac{1}{2}n(n-1)$  C-PHASE gates.
2. We apply  $\hat{U}_k$  to the state of the system. Using the basis in which  $\theta$  is expressed in binary notation (see Eq. 2.11), the representation of  $\hat{U}_k$  is diagonal and it can be decomposed into one- and two-qubit gates, as we will demonstrate in the following. By substituting the binary decomposition of  $\theta$  we firstly obtain:

$$(\theta - \pi)^2 = \left( 2\pi \sum_{j=1}^n \alpha_j 2^{-j} - \pi \right)^2 = 4\pi^2 \left[ \sum_{j=1}^n \left( \alpha_j 2^{-j} - \frac{1}{2n} \right) \right]^2 = 4\pi^2 \sum_{j_1, j_2=1}^n \left( \alpha_{j_1} 2^{-j_1} - \frac{1}{2n} \right) \left( \alpha_{j_2} 2^{-j_2} - \frac{1}{2n} \right). \quad (2.12)$$

We then extend this result to the operator  $\hat{\theta}$ :

$$\begin{aligned} (\hat{\theta} - \pi \hat{\mathbb{I}})^2 = 4\pi^2 \sum_{j_1, j_2=1}^n & \hat{\mathbb{I}}_1 \otimes \cdots \otimes \hat{\mathbb{I}}_{j_1-1} \otimes \hat{A}_{j_1} \otimes \hat{\mathbb{I}}_{j_1+1} \\ & \otimes \cdots \otimes \hat{\mathbb{I}}_{j_2-1} \otimes \hat{A}_{j_2} \otimes \hat{\mathbb{I}}_{j_2+1} \cdots \otimes \hat{\mathbb{I}}_{j_n} \end{aligned} \quad (2.13)$$

where  $\hat{\mathbb{I}}_j$  is the identity operator acting on the  $j$ -th qubit while  $\hat{A}_j$  is an operator whose definition comes from the result previously obtained, and that can be written as:

$$\hat{A}_j = \frac{1}{2^j} \frac{\hat{\mathbb{I}}_j - (\hat{\sigma}_z)_j}{2} - \frac{1}{2n} \hat{\mathbb{I}}_j \quad (2.14)$$

From Eq.2.9 we then have:

$$\begin{aligned} U_k = e^{i \frac{k(\hat{\theta} - \pi \hat{\mathbb{I}})^2}{2}} = \prod_{j_1, j_2=1}^n & \exp \left[ i 2\pi^2 k \left( \hat{\mathbb{I}}_1 \otimes \cdots \otimes \hat{\mathbb{I}}_{j_1-1} \otimes \hat{A}_{j_1} \otimes \hat{\mathbb{I}}_{j_1+1} \right. \right. \\ & \left. \left. \otimes \cdots \otimes \hat{\mathbb{I}}_{j_2-1} \otimes \hat{A}_{j_2} \otimes \hat{\mathbb{I}}_{j_2+1} \cdots \otimes \hat{\mathbb{I}}_{j_n} \right) \right], \end{aligned} \quad (2.15)$$

This is the product of  $n^2$  two-qubit gates, in particular C-PHASE shift gates, acting non-trivially on every possible pair of qubits  $j_1$  and  $j_2$ . We then consider a computational basis defined as  $\{|\alpha_{j_1} \alpha_{j_2}\rangle\} = \{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ , which collects all of the possible states the subsystem composed by  $j_1$ -th and  $j_2$ -th qubits can assume. In this basis the  $\hat{U}_k$  operator obtained in Eq.2.15 can be rewritten in a simple form as a function of the  $D_{j_1, j_2}$  diagonal matrices:

$$\hat{U}_k = \prod_{j_1, j_2=1}^n e^{i 2\pi^2 k D_{j_1, j_2}} \quad \text{with} \quad D_{j_1, j_2} = \begin{pmatrix} \frac{1}{4n^2} & 0 & 0 & 0 \\ 0 & -\frac{1}{2n} \left( \frac{1}{2^{j_2}} - \frac{1}{2n} \right) & 0 & 0 \\ 0 & 0 & -\frac{1}{2n} \left( \frac{1}{2^{j_1}} - \frac{1}{2n} \right) & 0 \\ 0 & 0 & 0 & \left( \frac{1}{2^{j_1}} - \frac{1}{2n} \right) \left( \frac{1}{2^{j_2}} - \frac{1}{2n} \right) \end{pmatrix} \quad (2.16)$$

Neglecting a global phase factor, it can be shown that each operator  $e^{i 2\pi^2 k D_{j_1, j_2}}$  can be further decomposed into a product of two phase-gates, applied to qubit  $j_1$  and  $j_2$  respectively, and a C-PHASE gate applied on both qubits if  $j_1 \neq j_2$  [22]. Namely we define this operators as  $R_z(\theta_{k,j})$  and  $CR_z(\Phi_{k,j_1,j_2})$  with phases:

$$\theta_{k,j} = -\frac{2k\pi^2}{2^j} + \frac{k\pi^2}{2^{2j-1}} \quad \Phi_{k,j_1,j_2} = \frac{2k\pi^2}{2^{j_1+j_2-1}}, \quad (2.17)$$

with  $j, j_1, j_2 \in \{0, 1, \dots, N\}$ .

3. The change from the  $\theta$  to the  $J$  representation is obtained by application of the inverse quantum Fourier transform ( $QFT^\dagger$ ), which requires  $n$  Hadamard gates and  $\frac{1}{2}n(n-1)$  C-PHASE gates.

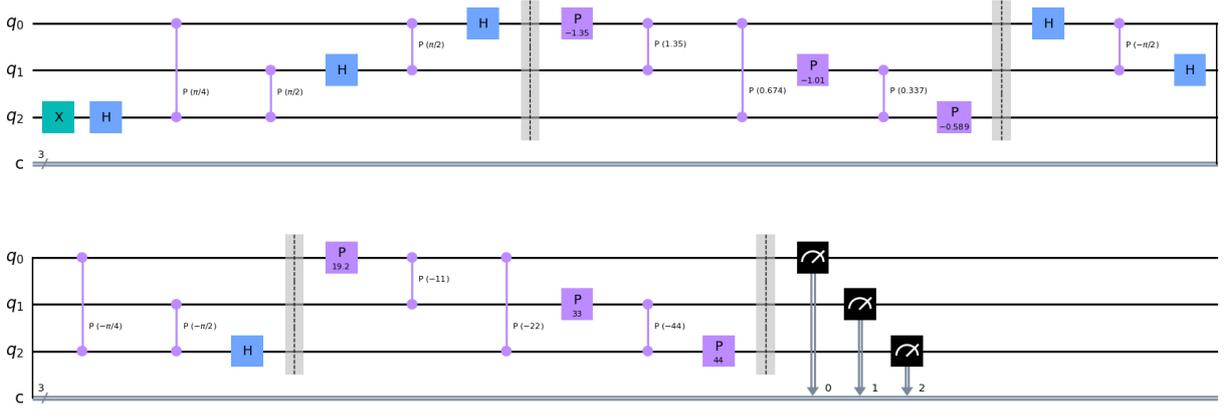


Figure 2.2: Quantum circuit of sawtooth map with  $n=3$  (from Qiskit)

The purple vertical lines correspond to CNOT gates while the P stands for phase gate. At the end of the circuit, we measure the momentum state of the system, storing the information in three classical bits

4. We apply the  $\hat{U}_T$  operator. This has essentially the same form as the operator  $\hat{U}_k$  in the  $\theta$  representation and, for this reason, an analogous decomposition. In particular, after substituting the binary decomposition 2.11 into 2.9, we obtain a product of  $n^2$  C-PHASE shift gates. This can further be decomposed into a product of phase shift gates acting on each qubit of the circuit and a series of C-PHASE gates acting on each possible pair of qubits if  $j_1 \neq j_2$ , whose phases are:

$$\theta_{T,j} = \frac{2N^2T}{2j+2} - \frac{N^2T}{2^{2j+1}} \quad \Phi_{T,j_1,j_2} = -\frac{2N^2T}{2^{j_1+j_2+1}}, \quad (2.18)$$

with  $j, j_1, j_2 \in \{0, 1, \dots, N\}$

In Figure 2.2 we present the resulting circuit in the case of  $n = 3$ .

In summary, the quantum algorithm for the quantum sawtooth map requires  $\mathcal{O}(n^2)$  gates per iteration, which is much faster than the  $\mathcal{O}(nN)$  operations needed by classical computers using the fast Fourier transform. Moreover, the resources required for quantum simulation are only logarithmic in system size  $N$ . With the further decomposition into a series of phase-shift gates and control-phase-shift gates we showed, one step of the quantum sawtooth map can be simulated using  $2(n^2 - n)$  CP,  $2n$  P, and  $2n$  H gates. For example, simulating  $n=3$  requires 24 quantum gates (12 single-qubit and 12 two-qubit gates) on IBM quantum hardware. We conclude that this circuit provides a simulation of the dynamics of the quantum sawtooth map which is exponentially faster than any known classical algorithm. Even though the gain in simulating the dynamics is exponential, it can be shown that the gain in extracting useful information from the simulation on a quantum computer is only quadratic [3].

## 2.3 Dynamical localization

The quantum sawtooth map demonstrates strikingly different behavior from the classical sawtooth map, particularly with respect to the diffusion of the momentum variable in the chaotic regime: after a certain time the diffusion of the momentum that we described in 2.1.1 is suppressed due to quantum interference, a phenomenon called *Dynamical Localization*.

We define  $t^*$  the *break time*, the time after which the quantum distribution reaches a steady state. Associated with it, we can define  $n^*$  the number of iterations of the map required to have an evolution of the system to time  $t^*$  (we remember that each iteration makes the system evolve to a time  $t \rightarrow t + T$ ). For  $t > t^*$  the state decays exponentially over the components of the momentum eigenkets with a distance major than a certain value, that we name  $l$ , from the initial momentum, which we call  $j_0$ .

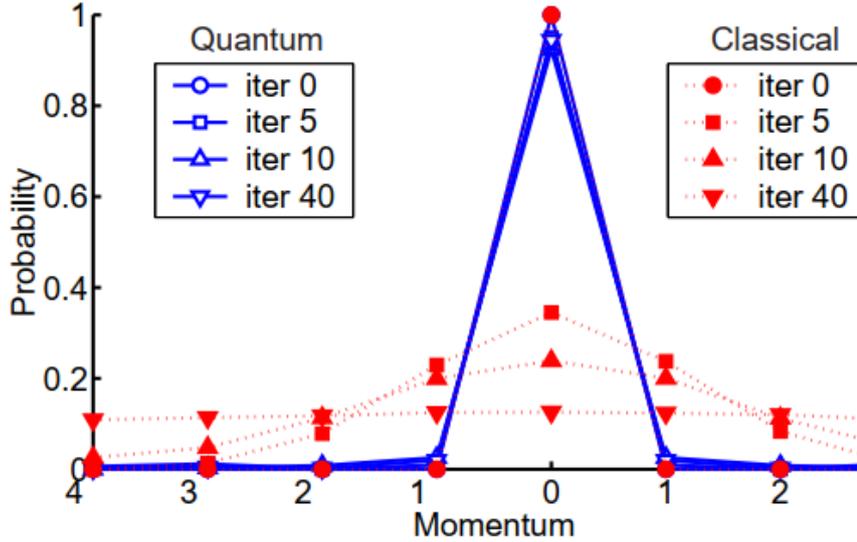


Figure 2.3: Momentum distribution after 0, 5, 10, and 40 iterations of the classical (red, filled markers) and quantum (blue, unfilled markers) sawtooth map ( $L = 7$ ,  $K = 1.5$ ,  $N = 8$ ), chaotic regime, localization length is less than 1 so that the state remains exponentially localized after a single iteration. We can see how the classical distribution is diffused as the number of iterations of the algorithm, corresponds to an evolution of time. In contrast, since we are considering a noiseless system, the quantum distribution is frozen and localized for every possible iteration of the map. The classical distribution represents 20,000 realizations of the map, with initial momenta  $j = 0$  and random initial positions uniformly distributed over the phase space. Credit: Henry K. M. et al./Physical Review [10]

Explicitly:

$$W_j = |\langle j|\psi\rangle|^2 \approx \frac{1}{l} \exp\left[-\frac{2|j-j_0|}{l}\right], \quad (2.19)$$

where the  $j$  singles out the eigenstates of  $\hat{J}$ . Therefore, for  $t > t^*$  only  $\sqrt{\langle(\Delta j)^2\rangle} \approx l$  levels are populated.

It can be shown from Heisenberg's principle that  $t^*$  is the time needed to resolve the particular energy spacing which characterizes dynamical localization. From this, it is straightforward to obtain that:

$$t^* \approx l. \quad (2.20)$$

At the same time, from an analogy with classical diffusion for the quantum sawtooth map (see 2.1), we have:

$$\sqrt{\langle(\Delta m)^2\rangle} \approx \sqrt{D_m T^*} \longrightarrow t^* \approx l \approx D_m \approx (\pi^2/3)k^2. \quad (2.21)$$

From this heuristic approximation, we see how a theoretical prediction of the localization length can be deduced from the kick strength  $k$ .

We stress the fact that in an ideal (noiseless) quantum system the time-evolution for  $t > t^*$  is “frozen” (see Fig. 2.3). However, the degree to which a physical system becomes localized may be reduced by noise effects, as we will deeply discuss in chapter 3.

After simulating the quantum sawtooth map dynamics on a quantum computer it is possible to extract the localization length  $l$  by running the algorithm repeatedly, measuring the state of the system in the momentum base, and storing the results which will ultimately be plotted on a histogram, as the one of Fig. 2.3.

We need about  $t^* = \mathcal{O}(l)$  map iterations to obtain the localized distribution. It is reasonable to use a basis size  $N = \mathcal{O}(l)$  to detect localization (so that the total number of states exceeds the localization length and we can visualize it on the histogram). In such a situation a classical computer

requires  $\mathcal{O}(l^2 \log l)$  operations to extract the localization length, while a quantum computer would require  $\mathcal{O}(l(\log l)^2)$  elementary gates. Therefore, the quantum computer provides a *quadratic speed up* in computing it.

We finally point out that this surprising interference effect requires the coherence of the quantum state. Quantum noise in the interaction between qubits and the external environment involves quantum decoherence effects as described in 1.5. For this reason, the study of this algorithm and its implementation on real quantum devices are very useful for determining how much quantum noise affects a certain computer or comparing the performances obtained from different quantum computers.



# Chapter 3

## Results

In this chapter we present the results obtained for the simulation of the quantum sawtooth map on IBM simulators and real quantum processing units. First, we briefly characterize the quantum simulators and real processors we used; then, we observe the emergence of dynamical localization with  $n \geq 3$  qubits on real quantum hardware and provide a comparison with the results obtained in the noiseless and noise-simulated cases. In the end, we discuss how the depth of the circuit and the number of iterations of the algorithm affect the possibility of visualizing dynamical localization after the measurement.

All simulations reported in this Thesis are obtained according to the following protocol:

- The initial state is peaked in momentum space,  $\psi_0(j) = \delta_{j,j_0}$ , and centered on  $j_0 = 0$ ;
- Parameters in the Hamiltonian equations 2.4 are set to:  $k = 0.273$  and  $K = kT = 1.5$ . In this regime, the break time  $t^* \approx k^2 < 1$ , so that the distribution is already localized, namely it decays exponentially in the momentum eigenbasis, after a single step of the dynamical map;
- To obtain significant statistics on the output distribution, the quantum algorithm is run 8192 times (shots);
- Results for the real quantum hardware are averaged over 10 repetitions, in order to average out fluctuations due to different noise realizations, and different timing in calibration

Data were collected in the timeframe between 2023/02/18 and 2023/03/01.

### 3.1 IBM quantum processors and simulators

IBM provides access to various quantum hardware and simulators through “IBM Quantum Experience”, a cloud-based quantum computing service. Among the quantum devices available, we used `ibmq_lima` and `ibmq_guadalupe` processors. Both of them use superconducting transmon qubits operating at a temperature close to absolute zero (15 mK). The qubits of `ibmq_lima` are arranged in a linear array, while the ones of `ibmq_guadalupe` in a square grid (see Figure 1.3). These devices have different numbers of qubits and exhibit different levels of noise, making them suitable for studying the dynamics of the quantum sawtooth map under different conditions. We report the technical description of these two processors in Table 3.1.

As a reference, we classically emulate the algorithm using the *Aer simulator*, an open-source high-performance simulator that can simulate quantum circuits with up to 30 qubits in noiseless conditions. Furthermore, “IBM Quantum Experience” also provides noise models, that can be used to account for the noise present on each device. We use this feature to classically emulate the real dynamics on the QPU, hereafter referred to as *fake lima* and *fake guadalupe*.

	Qubits	QV	Median CNOT error	Median readout error	Median $T_1$	Median $T_2$
ibmq_lima	5	8	1.072e-2	1.980e-2	86.16 $\mu s$	108.96 $\mu s$
ibmq_guadalupe	16	32	8.974e-3	1.565e-2	102.25 $\mu s$	121.53 $\mu s$

Table 3.1: Technical specifications of `ibmq_lima` and `ibmq_guadalupe`. The median CNOT error refers to the probability of a failure in the application of a CNOT gate, which is a reference value for the error for a two qubits gate. The median readout error is the median value of the error rate associated with the measurement process. *QV* refers to *quantum volume*, a metric used to evaluate the performance of quantum computers, defined as  $QV = 2^n * f$  where  $n$  is the number of qubits and  $f$  is the depth of the circuit. A high QV indicates that a quantum computer can handle a large number of qubits and execute complex quantum circuits with a high level of accuracy and fidelity. Credits: <https://quantum-computing.ibm.com/services/resources?tab=systems>

### 3.2 Dynamical localization on quantum hardware

In the following section, we present the results from the simulation of the dynamics of the quantum sawtooth map on real quantum hardware and classical simulators using the algorithm described in 2.2.

In Figure 3.1 we compare the state probability distribution in the momentum space as obtained from the simulation of dynamical localization with the *Aer simulator*, the *fake lima* simulator, and the real quantum hardware `ibmq_lima`. We observe a peak in the momentum distribution centered on  $j = 0$ . The probability distribution is exponentially suppressed on the neighboring momenta as expected in the localized phase. The height of this peak ( $W_1(0)$ ) decreases as we consider the effect of noise:  $W_1(0) \approx 0.82$  in the noiseless case,  $W_1(0) \approx 0.62$  for the fakeLima, and  $W_1(0) \approx 0.5$  on Lima. These results reproduce those already present in the literature [22]. We notice that the noise model underestimates some of the relevant noise channels: for this reason, fakeLima distribution is more peaked than Lima. Interestingly, the distribution obtained in the presence of noise displays a small peak on  $j = -4$ . This feature is due to the encoding of the momentum:  $j = -4$  is encoded in the  $|000\rangle$  state which is the target state of the noise channel (amplitude damping).

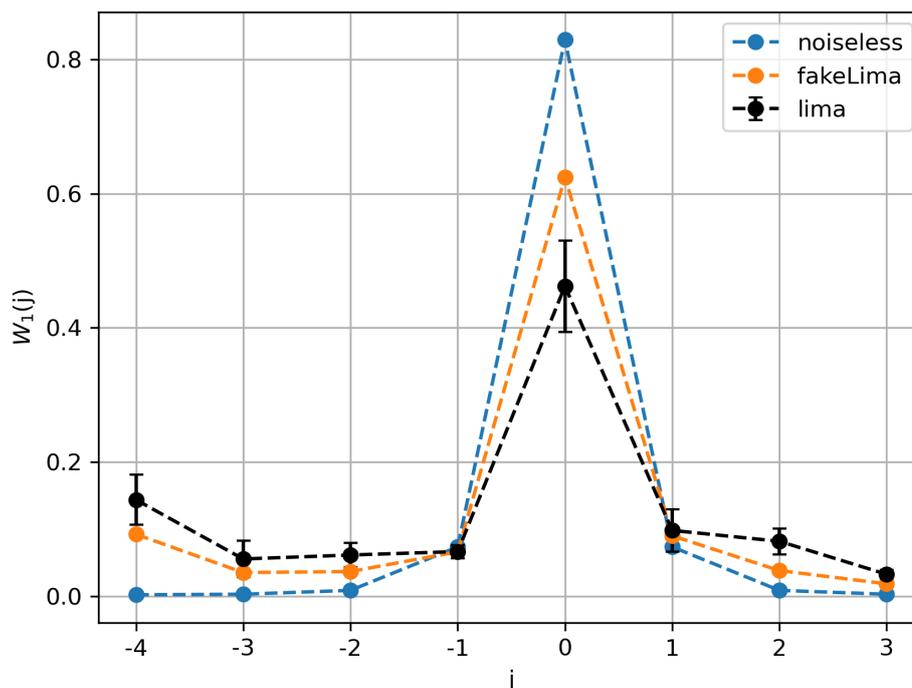


Figure 3.1: Height of the localization peak ( $W_1(0)$ ) in the quantum sawtooth map with  $n = 3$  qubits,  $K = 1.5$ ,  $k \approx 0.273$  ( $T = \frac{2\pi L}{N}$ , that gives  $L = 7$ ). Data from the quantum processor lima are obtained after averaging over 10 repetitions of 8192 experimental runs. Data were collected from 2023/02/18 to 2023/03/01.

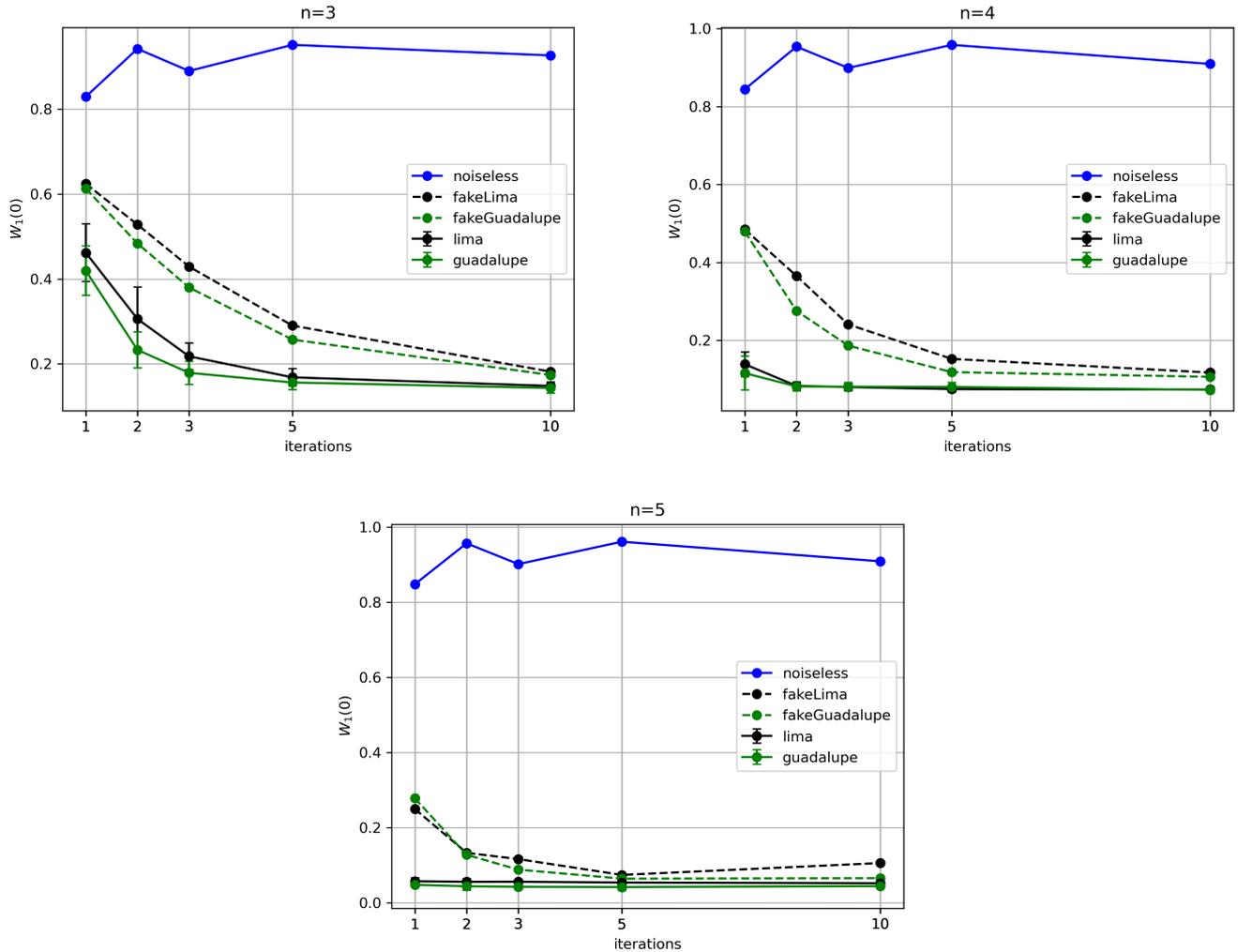


Figure 3.2: Height of the localization peak, namely  $W_1(0)$  as a function of the number of map iterations. The top-left figure refers to the case with  $n = 3$  qubits, the top-right to  $n = 4$  while the bottom one the case of  $n = 5$ . The parameters used are  $K = 1.5$  and  $k \approx 0.273$ . From relation  $T = \frac{2\pi L}{N}$  we have  $L \approx 7$  for  $n = 3$ ,  $L \approx 14$  for  $n = 4$ ,  $L \approx 28$  for  $n = 5$ . Data are for *ibmq\_lima* (black) and *ibmq\_guadalupe* (green) and are obtained after averaging over 10 repetitions of 8192 experimental runs. Data were collected from 2023/02/18 to 2023/03/01.

As discussed in Section 2.3, the quantum state in the localized regime ( $t > t^*$ ) preserves the localization for successive applications of the dynamical map. We show this behavior for the three qubit case in Fig. 3.2(top-left) (noiseless), by considering the height of the localization peak as a function of the number of iterations. By contrast, the real quantum hardware behaves very differently: at every iteration, there is a progressive loss in the localization peak, which ultimately becomes indistinguishable from noise, for both devices. Despite the better specifics (see Table 3.1), *ibmq\_guadalupe* is more prone to noise effects than *ibmq\_lima*. Nevertheless, the coherent dynamics is no longer observable on both quantum devices after five applications of the map.

In Figure 3.2(top-right) and 3.2(bottom) we consider the case where the coherent localization phenomenon involves a larger number of qubits ( $n = 4$  and  $n = 5$ ). The parameters in the Hamiltonian are modified to preserve the same localization conditions. In these cases, the height of the peak is hardly detectable on the real hardware even after the first map application. This feature affects similarly both the devices regardless of the number of qubits available on the QPU. As for  $n = 3$ , the noise simulators considerably underestimate the level of noise.

As discussed in Chapter 2.1, the quantum algorithm for the quantum sawtooth map requires all-to-all connectivity which is very hard to implement on hardware with fixed connectivity. For this reason,

we have to transpile the original circuit into a new one that matches the connectivity on the QPU. Therefore, the same circuit has different depths depending on the QPUs. However, in this case, even if the two QPUs have different connectivity graphs, the circuit depths are close when considering the same number of qubits. In order to compare the performance of the two QPUs, we consider the ratio between the localization peak ( $W_1(0)$ ) on the real device and the one in the noiseless case for quantum circuits with different depths, each corresponding to a different number of qubits. Results are shown in Figure 3.3. Using a logarithmic scale on both axes, we can highlight the polynomial scaling of the localization peak with the depth, for both devices. From this comparison, `ibmq_lima` still performs better than `ibmq_guadalupe`. However, this result is in disagreement with the specifics for the two devices reported in Table 3.1, where `ibmq_guadalupe` seems to outperform `ibmq_lima`. This discrepancy is probably due to the different nature of the metrics to assess the capability of a quantum device, the quantum volume, as opposed to the localization effect, which is a truly many-body phenomenon.

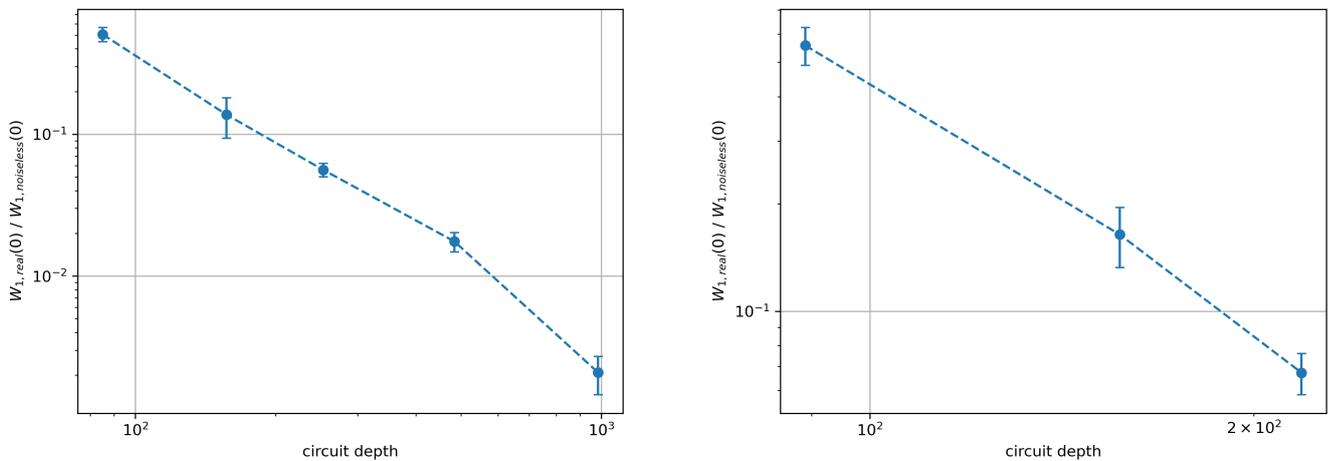


Figure 3.3: Decay of the localization peak with the circuit depth. In the right picture, the circuit depths point to the cases for 3, 4, and 5 qubits, respectively. Data are obtained from the `ibmq_lima` QPU after averaging over 10 repetitions of 8192 experimental runs each. In the left picture, the circuit depths point to the cases for 3, 4, 5, 7, and 10 qubits respectively. Data are obtained from the `ibmq_lima` QPU after averaging over 10 repetitions of 8192 experimental runs each. Both vertical and horizontal scales are logarithmic. Data were collected from 2023/02/18 to 2023/03/01.

# Conclusions

In this thesis, we have reproduced the dynamical localization phenomenon of the quantum sawtooth map on real quantum hardware and simulators available through IBM Quantum Experience. Since dynamical localization is extremely sensitive to noise, the simulation on actual quantum hardware offers a suitable test case to prove the reliability of the present platforms. Furthermore, it allows comparison in terms of performance between quantum processing units (QPU) with different connectivity. To this aim, we have studied and implemented an efficient quantum algorithm [3] based on the forward/backward Fourier transform, which requires  $\mathcal{O}(n^2)$  gates where  $n$  is the number of qubits. Parameters in the model have been chosen in a way that a single application of the quantum map allows the emergence of the localized phase. In agreement with [22], we have observed the localization peak for the case of  $n = 3$  qubits both on `ibmq_lima` and `ibmq_guadalupe`, although significantly reduced with respect to the noiseless dynamics. Interestingly, the noise models available for the two QPUs underestimate some of the relevant noise channels, thus poorly reproducing data from real hardware.

The persistence of the localization phenomenon has been considered by subsequent applications of the quantum map. By contrast with the noiseless case, the real quantum hardware shows a progressive loss in the localization peak, which becomes indistinguishable from noise after five map applications, for both devices. The same analysis has been extended to the case where the coherent localization phenomenon involves  $n = 4, 5$  qubits: the localization peak is reduced but still detectable. Finally, in order to compare the performance of the two QPUs, we have considered the dependence of the localization peak on the circuit depth, which highlights a polynomial scaling.

From our analysis, we conclude that `ibmq_lima` performs better than `ibmq_guadalupe`. However, this result is in disagreement with the specifics reported for the two devices, where `ibmq_guadalupe` outperforms `ibmq_lima` in terms of quantum volume. This discrepancy is probably due to the different metrics used, the quantum volume as opposed to the localization test, which is a truly many-body phenomenon. The results motivate further analysis also on different platforms where different connectivities can be realized, as it happens with trapped ions or neutral atoms.



# Bibliography

- [1] G. ANDERSSON, *Circuit quantum electrodynamics with a transmon qubit in a 3d cavity*, Master's thesis, Technische Universitat Munchen, 2015.
- [2] F. ARUTE, K. ARYA, R. BABBUSH, D. BACON, J. C. BARDIN, R. BARENDS, R. BISWAS, S. BOIXO, F. G. S. L. BRANDAO, D. A. BUELL, AND ET AL., *Quantum supremacy using a programmable superconducting processor*, *Nature*, 574 (2019), pp. 505–510.
- [3] G. BENENTI, G. CASATI, AND S. MONTANGERO, *Quantum computing and information extraction for dynamical quantum systems*, in *Experimental Aspects of Quantum Computing*, Springer US, pp. 273–293.
- [4] G. BENENTI, G. CASATI, AND G. STRINI, *Principles of Quantum Computation and Information*, World Scientific, 2007.
- [5] H. J. BRIEGEL, D. E. BROWNE, W. DÜR, R. RAUSENDORF, AND M. V. DEN NEST, *Measurement-based quantum computation*, *Nature Physics*, 5 (2009), pp. 19–26.
- [6] J. CHABÉ, G. LEMARIÉ, B. GRÉMAUD, D. DELANDE, P. SZRIFTGISER, AND J. C. GARREAU, *Experimental observation of the anderson metal-insulator transition with atomic matter waves*, *Physical Review Letters*, 101 (2008).
- [7] S. DEBNATH, N. M. LINKE, C. FIGGATT, K. A. LANDSMAN, K. WRIGHT, AND C. MONROE, *Demonstration of a small programmable quantum computer with atomic qubits*, *Nature*, 536 (2016), pp. 63–66.
- [8] R. P. FEYNMAN, *Simulating physics with computers*, *International Journal of Theoretical Physics*, 21 (1982), pp. 467–488.
- [9] J. M. GAMBETTA, J. M. CHOW, AND M. STEFFEN, *Building logical qubits in a superconducting quantum computing system*, *npj Quantum Information*, 3 (2017).
- [10] M. K. HENRY, J. EMERSON, R. MARTINEZ, AND D. G. CORY, *Localization in the quantum sawtooth map emulated on a quantum-information processor*, *Phys. Rev. A*, 74 (2006), p. 062317.
- [11] M. W. JOHNSON, M. H. S. AMIN, S. GILDERT, T. LANTING, F. HAMZE, N. DICKSON, R. HARRIS, A. J. BERKLEY, J. JOHANSSON, P. BUNYK, E. M. CHAPPLE, C. ENDERUD, J. P. HILTON, K. KARIMI, E. LADIZINSKY, N. LADIZINSKY, T. OH, I. PERMINOV, C. RICH, M. C. THOM, E. TOLKACHEVA, C. J. S. TRUNCIK, S. UCHAIKIN, J. WANG, B. WILSON, AND G. ROSE, *Quantum annealing with manufactured spins*, *Nature*, 473 (2011), pp. 194–198.
- [12] S. JOHNSTUN AND J.-F. V. HUELE, *Understanding and compensating for noise on IBM quantum computers*, *American Journal of Physics*, 89 (2021), pp. 935–942.
- [13] P. KOCH, *The importance of resonances in microwave “ionization” of excited hydrogen atoms*, *Physics Reports*, 255 (1995), pp. 289–403.
- [14] H. LEVINE, A. KEESLING, G. SEMEGHINI, A. OMRAN, T. T. WANG, S. EBADI, H. BERNIEN, M. GREINER, V. VULETIĆ, H. PICHLER, AND M. D. LUKIN, *Parallel implementation of high-fidelity multiqubit gates with neutral atoms*, *Physical Review Letters*, 123 (2019).

- [15] L. S. MADSEN, F. LAUDENBACH, M. F. ASKARANI, F. RORTAIS, T. VINCENT, J. F. F. BULMER, F. M. MIATTO, L. NEUHAUS, L. G. HELT, M. J. COLLINS, A. E. LITA, T. GERITS, S. W. NAM, V. D. VAIDYA, M. MENOTTI, I. DHAND, Z. VERNON, N. QUESADA, AND J. LAVOIE, *Quantum computational advantage with a programmable photonic processor*, *Nature*, 606 (2022), pp. 75–81.
- [16] I. MANAI, J.-F. CLÉMENT, R. CHICIREANU, C. HAINAUT, J. C. GARREAU, P. SZRIFTGISER, AND D. DELANDE, *Experimental observation of two-dimensional anderson localization with the atomic kicked rotor*, *Physical Review Letters*, 115 (2015).
- [17] S. MONTANGERO, *Dynamically localized systems: Exponential sensitivity of entanglement and efficient quantum simulations*, *Phys. Rev. A*, 70 (2004), p. 032311.
- [18] F. L. MOORE, J. C. ROBINSON, C. F. BHARUCHA, B. SUNDARAM, AND M. G. RAIZEN, *Atom optics realization of the quantum kicked rotor*, *Physical Review Letters*, 75 (1995), pp. 4598–4601.
- [19] M. A. NIELSEN AND I. L. CHUANG, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.
- [20] S. NOTARNICOLA, *Coupled quantum kicked rotors: a study about dynamical localization, slow heating and thermalization*, Master’s thesis, Scuola Internazionale Superiore di Studi Avanzati, 2018.
- [21] J. S. OTTERBACH, R. MANENTI, N. ALIDOUST, A. BESTWICK, M. BLOCK, B. BLOOM, S. CALDWELL, N. DIDIER, E. S. FRIED, S. HONG, P. KARALEKAS, C. B. OSBORN, A. PAPA-GEORGE, E. C. PETERSON, G. PRAWIROATMODJO, N. RUBIN, C. A. RYAN, D. SCARABELLI, M. SCHEER, E. A. SETE, P. SIVARAJAH, R. S. SMITH, A. STALEY, N. TEZAK, W. J. ZENG, A. HUDSON, B. R. JOHNSON, M. REAGOR, M. P. DA SILVA, AND C. RIGETTI, *Unsupervised machine learning on a hybrid quantum computer*, 2017.
- [22] A. PIZZAMIGLIO, S. Y. CHANG, M. BONDANI, S. MONTANGERO, D. GERACE, AND G. BENENTI, *Dynamical localization simulated on actual quantum hardware*, *Entropy*, 23 (2021).
- [23] A. SAHNOUNE AND D. BERKANI, *On statistical properties of chaotic signals generated by negative sawtooth maps*, *Automatic Control and Computer Sciences*, 56 (2022), pp. 356–363.
- [24] P. SCHOLL, M. SCHULER, H. J. WILLIAMS, A. A. EBERHARTER, D. BARREDO, K.-N. SCHYMIK, V. LIENHARD, L.-P. HENRY, T. C. LANG, T. LAHAYE, A. M. LÄUCHLI, AND A. BROWAEYS, *Quantum simulation of 2d antiferromagnets with hundreds of rydberg atoms*, *Nature*, 595 (2021), pp. 233–238.
- [25] P. W. SHOR, *Algorithms for quantum computation: Discrete logarithms and factoring*, in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, IEEE, 1994, pp. 124–134.
- [26] R. WU, T.-F. LI, A. KOFMAN, J. ZHANG, Y.-X. LIU, Y. PASHKIN, J.-S. TSAI, AND F. NORI, *Spectral analysis and identification of noises in quantum systems*, *Physical Review A*, 87 (2012).