Pós-Graduação em Ciência da Computação

QUANTUM WEIGHTLESS NEURON DYNAMICS

By

Fernando Maciano de Paula Neto

M.Sc. Dissertation

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE/2016

Federal University of Pernambuco
Center for Informatics
Graduate in Computer Science

Fernando Maciano de Paula Neto

QUANTUM WEIGHTLESS NEURON DYNAMICS

A M.Sc. Dissertation presented to the Center for Informatics of Federal University of Pernambuco in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

Advisor: *Teresa Bernarda Ludermir*
Co-Advisor: *Wilson Rosa de Oliveira*

RECIFE
2016

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

P324q Paula Neto, Fernando Maciano de
Quantum weightless neuron dynamics / Fernando Maciano de Paula Neto. – 2016.
79 f.: il., fig.

Orientadora: Teresa Bernarda Ludemir.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da computação, Recife, 2016.
Inclui referências.

1. Redes neurais artificiais. 2. Computação quântica. 3. Redes neurais quânticas. 4. Sistemas dinâmicos I. Ludemir, Teresa Bernarda (orientadora). II. Título.

006.3 CDD (23. ed.) UFPE- MEI 2016-025

Fernando Maciano de Paula Neto

Quantum Weightless Neuron Dynamics

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação

Aprovado em: 01/03/2016.

_____

Orientadora: Profa. Dra. Teresa Bernarda Ludermir

BANCA EXAMINADORA

_____

Prof. Dr. Borko Stosic
Departamento de Estatística e Informática / UFRPE

_____

Prof. Dr. Adenilton José da Silva
Departamento de Estatística e Informática / UFRPE

_____

Prof. Dr. Wilson Rosa de Oliveira Junior
Departamento de Estatística e Informática / UFRPE
(**Co-orientador**)

In some way, that it can bring benefit to all living beings

Acknowledgements

I am very grateful to all people who helped directly the execution of this work: my advisor professor Teresa and my co-advisor professor Wilson, for all the advice and motivation to work in this subject; my family who allowed me the comfort, warmth and respect for this work could be written.

Friends, classmates, Centro de Informática-UFPE professors, technical people and all others who contributed indirectly or inspired this work, I also thank them.

This work was supported by CNPq scholarship. I thank the Brazilian Federal Government for this incentive.

Quantum physics has bringing to light the importance to perceive that the scientist observation and the object are created together in the experiment.

—ALFREDO AVELINE, PHYSICS PROFESSOR, UFRS

Resumo

Os mais variados sistemas sociais, biológicos, físicos, químicos e computacionais tem sido investigados pela área de Sistemas Dinâmicos para formalizar o comportamento no tempo e quantificar e qualificar variações paramétricas desses sistemas. Na biologia em particular, estudos tem mostrado que a maximização de aprendizado de um neurônio pode acontecer dentro de certas condições da sua dinâmica onde o processamento de informação é otimizado. Espera-se então que essas condições possam ser reconhecidas e utilizadas em modelos artificiais.

Este trabalho descreve o comportamento do neurônio artificial quântico sem peso **qRAM**, desde a concepção de modelos de iteração - visto as condições físico-matemáticas da computação quântica que restringe a extração da informação isolada do valor de saída do neurônio a cada etapa de tempo - até sua análise paramétrica de onde comportamentos convergentes, amortecidos ou oscilatórios são detalhados. Ferramentas dos sistemas dinâmicos como diagrama de órbitas e séries temporais ilustram qualitativamente sua variabilidade temporal.

A principal contribuição desse trabalho é detalhar o comportamento do neurônio **qRAM** a fim de que os resultados possam ser usados dentro da área de aprendizagem de máquina, acoplado com sistemas maiores e complexos, com maximização de tarefas de aprendizado. Como resultado, há proposição de mais um modelo de dinâmica neuronal, o **QEM**, o estudo paramétrico dos modelos de dinâmicas existentes, que se identifica comportamentos subamortecidos, sobreamortecidos e não-amortecidos na dinâmica, assim como a apresentação de uma configuração neuronal dentro da arquitetura quântica que apresenta comportamento caótico. Um modelo de medição quantitivo para comparar dinâmicos foi também proposto.

Palavras-chave: Sistemas Dinâmicos. Computação Quântica. Neurônios Sem-Peso. Operadores Não-unitários. Emaranhamento. Problemas NP-Completo. Caos

Abstract

A wide spectrum of social, biological, physical, chemical and computational systems have been investigated by the tools and techniques from the field of Dynamical Systems Theory to formalize the behaviour in time and quantify and qualify the parametric variations of those systems. In Biology in particular, studies have shown that learning neuron maximization can occur in specific dynamics conditions where information processing is optimized. This it may be expected that some of those conditions can be recognized and used in artificial models.

This work studies the quantum artificial neuron weightless **qRAM** behavior, from the design iteration models - taking into account the physical and mathematical conditions of quantum computing that restricts the extraction of information at every time step - to its parametric analysis where converging behaviors, damped or oscillatory, are detailed. Tools of dynamical systems like orbits diagram and time series qualitatively illustrate its temporal variability.

The main contribution of this work is to detail the neuron **qRAM** behavior so that the results can be used within the machine learning area, coupled with larger systems to achieve maximum learning tasks. As result, we propose a novel dynamical neuron model, named Quadratic Extraction Model (QEM), we perfom parametric studies of the existing models where underdamped, overdamped and undamped behaviour are encountered, and we present a presentation of a neuron configuration inside a quantum architecture with chaos behaviour. A quantitative measure model to compare dynamics orbits was also proposed.

Keywords: Dynamical Systems. Quantum Computing. Weightless Neurons. Non-Unitary Operators. Entanglement. NP-Complete Problems. Chaos

List of Acronyms

qRAM	Quantum RAM Based Node	18
ANN	Artificial Neural Networks	17
PLN	Probabilistic Logic Node	18
MPLN	Multi-valued Probabilistic Logic Node	18
REM	Real Extraction Model	20
QEM	Quadratic Extraction Model	51

Contents

1

Introduction

No man really becomes a fool until he stops asking questions.

—CHARLES P. STEINMETZ

The field of Dynamical Systems brings quantitative and qualitative working explanation for any existing system which evolves in time (STROGATZ, 1994). To find the analytical expression for the behaviour of a particular system does not necessarily lead to an understanding of its parametric dependency, its possible temporal convergence, whether there is chaos, fractality and bifurcation. Progress in this field is achieved by building techniques, tools and methods to examine how the various elements of the systems, from the simplest to the most complex, correlate themselves in time.

A dynamical model can be used to produce an expected behavior, serving as input signal generator or even as the main module functioning as a complex system, e.g. to encrypt texts may involve a chaotic module (BAPTISTA, 1998) or in fractals it can help either to generate or to segment textures (CHAUDHURI; SARKAR, 1995; KELLER; CHEN; CROWNOVER, 1989). Biological neurons should work better in chaotic regime, bifurcation or phase change (KINOUCHI; COPELLI, 2006; TORRES; MARRO, 2015; GROSS, Online em 2015).

The fact that biological neurons can learn under certain dynamical behavior inspired the research reported in this dissertation. A quantum artificial weightless neuron model named **qRAM** was proposed by OLIVEIRA (2009). To understand **qRAM** quantitative and qualitative dynamical became an important task because the **qRAM** is a neuron with parameters that will influence their internal characteristics, such as its computing power and learning speed.

The analysis of the dynamical of a quantum weightless artificial neuron is an interdisciplinary field of study involving dynamical systems, quantum computing and artificial intelligence. One could then offer several introductory views for this research. We choose to start from the history of dynamical systems and to move toward the study of the dynamical of classical and

quantum neurons.

1.1 The birth of the field of Dynamical Systems

The early history of Dynamical Systems goes back to the principles of science and astronomy. It is basically the study of celestial bodies positions and velocities changing the first contributions to the construction of the Dynamic Systems area. The philosopher Aristotle (384-322 BC), pupil of Plato, was responsible for Western knowledge structuring and started the study of celestial bodies dynamics. Among his treatises on Biology, Economics, Rhetoric, Aesthetics, Ethics, Politics and Psychology, it is also his legacy in Physics. In other words, one of his main contribution was the methodological rules organization about the investigation of the dynamical of the Moon orbits, the Sun, the Earth and the Universe. However, underpinned by his expertise in his few research resources, Aristotle had a qualitative and incomplete vision of celestial mechanics. There was no quantitative predictions on the location of planets (MONTEIRO, 2006).

C. Ptolemy (85-165) continued the Aristotelian studies. After him, came T. Brahe (1546-1601), N. Copernicus (1473-1543), J. Kepler (1571-1630) and some others. It is not our intention, in this introduction, mention all the great things and astronomical physics disappointments, but we can highlight that studies of celestial bodies culminated in the Treaty of Dynamic concepts, proposed by the physicist, mathematician, astronomer and philosopher Galileo Galilei (1564-1642). Galileo was pioneer using telescope and introduced the concept of inertia, acceleration of bodies and gravity.

In the year of Galileo's death, it was born E. Newton (1642-1727) who derived the law of universal gravitation, proposed the Newton Three Laws for moving bodies: principle of inertia, fundamental principle of dynamics and the principle of action and reaction as well he wrote the famous *Mathematical Principles of Natural Philosophy*. To validate his idea, Netwon tested his theories in the Earth-Moon system model. Newton proposed that it would be possible to describe the motion of bodies and show that they are affected by each other. That analytical solution is only trivial for two bodies. The following physical and mathematical scientists generations attempted to apply the method to three bodies and after one finally proved insoluble in the sense that you can not solve, analytically and accurately, the equations governing their movements. The solution of that equation came only appear in 1917, given by A. Einstein (1879-1955), through the General Theory of Relativity (MONTEIRO, 2006).

We are closer to the modern studies of the Dynamic Systems area, but we could speak before about quantum mechanics and its uncertainty principle, proposed by WK Heisenberg (1901-1976).

The Universe was the landscape for depth studies in Physics when scientists decided to look their lenses toward the microscopic world. To meet at the same time, the position and velocity of an electron was impossible within the theory of quantum mechanics. That occurs because to measure one of the observable parameters (position, for example) of an electron

system would need to change the system working (and also changing the object speed). To predict the positions in macromolecular scales are theoretically trivial and possible experimentally, as did Galileo with pendulums, inclined planes and projectiles. But that ease of calculation did not exist at the microscopic universe. One wonders if some existing phenomena had correlation with only quantum factors or intrinsic sensibility of the system working. For example, scientists often cite the difficulty in temperature variation predicting and the occurrence of rain as theoretically hard to do. The desire to understand **how** the systems behave and what the **cases** of system sensibilities began to gain power.

JH Poincaré (1854-1912) then introduced the qualitative analysis of the systems, using geometrical and topological techniques. His work is then pioneer on the qualitative theory since he realized that the system could be investigated without being needed to find analytical solutions, solving nonlinear differential equations (still non-trivial nowadays). Analyzing the background of the problem initialized by Newton three-body problem, Poincare found that small variations in the early system conditions led him to a final behavior entirely different. That is a narrative of a concept that would only become formalized nearly 62 years later through the Chaos definition.

In 1962, EN Lorenz (1917-2008) became professor of atmospheric science at MIT. At that time, to predict the temperature meant using the linear prediction method: to find the linear coefficient weights of the various system variables (humidity, pressure, temperature the day before, temperature in the previous hour, etc).

Lorenz wonders if any problem related to weather forecast could be solved by linear forecasting method, as believed N. Wiener (1894-1964) (MONTEIRO, 2006). For that, Lorenz decided to generate a time series of a nonlinear differential equation numerical solution. Such non-linear differential equation system, the Navier-Stokes equations, modelled in a simplified way the atmospheric circulation.

What Lorenz identified is that linear forecast model is enough to predict stationary and periodic systems. When he found parameters in the equation system which generated aperiodic data, Lorenz showed that the linear prediction really could not predict those series.

For to do that, Lorenz used a computing to integrate numerically during on average six hours at every step (MONTEIRO, 2006). As he checked the program running once every day, he had printed the value of 12 variables every 4 integration steps, with only first 3 significant digits of each variable, appearing on the screen in a single line. Deciding view in more details the behavior of the 12 variables, Lorenz modified his code to print more variables informations in the middle of the list that was already being printed. However, the result was that the calculated values were not equal now to those obtained in the previous calculation. Lorenz changed his computers valve, believing be experiencing some electrical problem. He re-examined the data and found that the first of the new experiment values were equal to the previous experiment, but after some iterations, the values differed in the last decimal place and “accumulate the error". The difference between sets then increased until the numbers turn completely different. Lorenz then understood that by modifying the code, he had inserted a tiny change in the calculation

of number rounding, which made the series generated exponentially distanced series of the first experiment. That analysis made him interpret that some equations had sensitivity to initial conditions and the numerical accuracy for predicting atmospheric circulation could lead to the calculation completely different values than expected. That sensitivity became known as *butterfly effect*, “one of a butterfly wings beating in Brazil can cause the appearance of a tornado in Texas" said Lorenz.

Because the discovery of that sensibility to initial conditions of certain systems, Lorenz is known as "father of chaos" and the search area grew. Soon after, B. Mandelbrot (1924-2010) understood and popularized fractals, showing that the study of dynamical systems could be in the sciences and the arts, synthetically generating textures, understanding the biological oscillations such as heartbeat, etc (STROGATZ, 1994).

With the advancement of computers and computer graphics, it was possible to visualize the behavior of systems in massively repetitive calculations, avoiding manual hard working. In 1976, J. Hubbard (1945), analyzing the dynamical of Newton’s method to find the root of complex equations, took an important step towards the analysis of interesting patterns in dynamical systems. Newton’s method is an example of dynamical system, because the input value of its function is its output in the previous time (STROGATZ, 2012).

The starting point is the initial value. If it is close to the root, it converges to a root system. In equations with two roots, the method converges to the root closest to the initial condition. In equations with more than two roots, the algorithm is very sensitive. To display this behavior, Hubbard ran the algorithm for thousands of different starting points on a graph and colored with a hue in accordance with the dynamic convergence speed. It is intuitive to think that the points near roots attract more quickly to the roots, but how behave the convergence time at equidistant points from two roots? The result was the appearance of complex regions, with patterns that mingled and formed repetitive but staggered figures: so it was born an area of dynamical system called *dynamical of complex number*. In the figure created by Hubbard, at the border, there was clearly a chaotic behavior: to move toward a root depends on the valuation of decimal places of the initial condition value, an unpredictable event to be taken simply by looking at the value of the number.

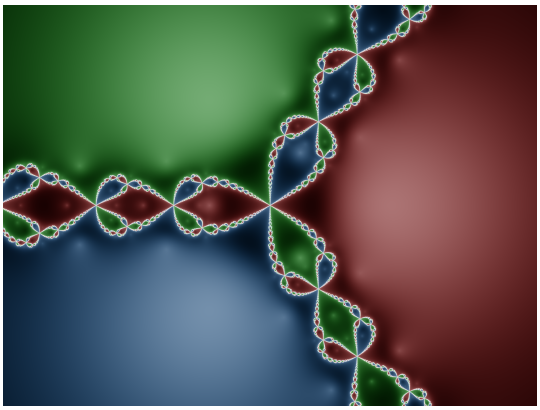
Then the area of dynamical systems focuses on 3 major concerns:

1. To simulate objects do not yet exist: through a simulation, check the object behavior in various iterations within their environment.

2. To explain the operation of existing systems: the main task of physics, which elucidates the laws governing the functioning of the Universe, in astronomical or atomic scales. Biology, Engineering and Chemicals also make his role in the categorization of various objects according to their iteration of the environment.

3. To understood risk situations: some systems need to be carefully studied before being executed, such vaccination systems, or piloting an aircraft system under turbulent

Figure 1.1: Hubbard experiment: Coloring points according to the speed of convergence to the root of the equation show the system fractality $f : z \to z^3 - 1$. The three colors are the three roots of the system and it is possible to see the fractality (ocurrence of a same figure in differents scales) in regions of borders.



Source: `https://en.wikipedia.org/wiki/Julia_set`.

conditions.

1.2 Quantum Dynamics

Iterate a quantum systems is theoretically to apply several times the same unit operator to a quantum state. However it is considered physically hard to keep quantum states isolated from all external perturbation (SCHLOSSHAUER, 2005). On this point of view, open quantum systems or measurement may have nonlinear behavior (TERNO, 1999; KISS et al., 2011; BECHMANN-PASQUINUCCI; HUTTNER; GISIN, 1998; KISS et al., 2006). This nonlinearity had profound consequences because it is a fundamental condition for the emergence of chaotic standards. Then, the quantum systems can be analyzed from the point of view of classical chaos (STÖCKMANN, 2007). In studies related to the dynamics and quantum computation, there is used non-linear operators in addition to the normal measuring operators (KISS et al., 2006), founding chaos in the dynamical of an arbitrary quantum operator using the nonlinear operator of BECHMANN-PASQUINUCCI; HUTTNER; GISIN (1998), considering that iterated model an purification protocol to increasing entanglement within a system (KISS et al., 2006).

T. Kiss *et al.* model (KISS et al., 2006) analyses the dynamics over one qubit coupled with the nonlinear operator proposed by BECHMANN-PASQUINUCCI; HUTTNER; GISIN (1998). The BECHMANN-PASQUINUCCI; HUTTNER; GISIN (1998) operator is employed to distinguish optimally between two non-orthogonal spin-1/2 states. The KISS et al. (2006) dynamics is mapped by Equation (1.1), where N is the function that normalises the qubit with a factor. In this qubit, the factor is $1/\sum \rho_{ij}^2$.

$$\rho = S\rho, \quad \rho_{ij} \xrightarrow{S} N\rho_{ij}^2 \tag{1.1}$$

Thus, if we have one qubit, its transformation S proposed by Bechmann *et al.* is:

$$|\psi\rangle_{input} = \alpha|0\rangle + \beta|1\rangle \xrightarrow{S} |\psi\rangle_{output} = N(\alpha^2|0\rangle + \beta^2|1\rangle) \tag{1.2}$$

KISS et al. (2006) propose to include this transformation S during the dynamics over one qubit. The rotation operator U is a generic rotation operator in the Hilbert space, with x and ϕ variables, as shown in the Equation (1.3).

$$U(x,\phi) = \begin{pmatrix} cos(x) & sin(x)e^{i\phi} \\ -sin(x)e^{-i\phi} & cos(x) \end{pmatrix} \tag{1.3}$$

Given a $|\psi\rangle$ qubit, an initial pure state:

$$|\psi\rangle = N(z|0\rangle + |1\rangle) \tag{1.4}$$

where N is the renormalization factor $\frac{1}{\sqrt{1+|z|^2}}$. The quadratic operator S (BECHMANN-PASQUINUCCI; HUTTNER; GISIN, 1998) and a generic rotation operator are applied. The application of these operations are as described below:

$$\begin{aligned} &|\psi\rangle = N(z|0\rangle + |1\rangle) \xrightarrow{S} |\psi_2\rangle = N(z^2|0\rangle + |1\rangle) \\ &U|\psi_2\rangle = |\psi_3\rangle = N((z^2 cos(x) + sin(x)e^{i\phi})|0\rangle + \\ &+ (-sin(x)e^{-i\phi}z^2 + cos(x))|1\rangle) \end{aligned} \tag{1.5}$$

.

The state $|\psi_3\rangle$ is the quantum state after the first iteration. To recovery the z value after this dynamics, it is necessary to transform the state to the original format of pure state.

$$\left|\psi_{output}\right\rangle = N(z'|0\rangle + |1\rangle) \tag{1.6}$$

In $|\psi_3\rangle$ qubit, the $|0\rangle$ amplitude value is divided by the $|1\rangle$ amplitude value, the value of z after one iteration is encountered:

$$z' = \frac{z^2 cos(x) + sin(x)e^{i\phi}}{-sin(x)e^{-i\phi}z^2 + cos(x)} \tag{1.7}$$

It is easy to see that the normalization rate does not need to be considered because it will be cancelled after the division of the amplitudes.

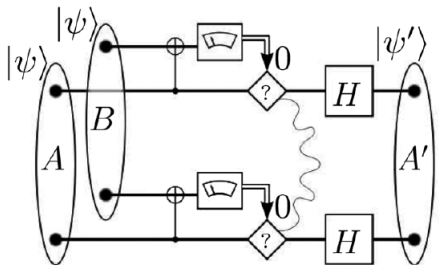
Considering $p = tan(x)e^{i\phi}$, the analytic formula of this dynamical model that is studied in (KISS et al., 2006) is:

$$F_p(z) = \frac{z^2 + p}{-p^* z^2 + 1} \tag{1.8}$$

The dynamics proposed in (KISS et al., 2006) has one free variable z and one constant

p whose the initial condition determines the existence or not of chaos during its dynamics. After that, KISS et al. (2011) extend the purification protocol dynamics for more than one qubit depicted in Figure 1.2. It was demonstrated that the nonlinear map of that purification protocol

Figure 1.2: Schematic representation of one iteration step of the entanglement purification protocol.



Source: (KISS et al., 2011)

$$g(x) = \frac{2x^2}{1+x^4} \tag{1.9}$$

represented in Equation 1.9 using a couple of two qubits presents chaos and sensibility of initial conditions.

In KISS et al. (2006, 2011), the dynamics studies have objective to increase entanglement over the purification protocol. For that, the initial condition quantum state has the form either $|\psi(x)\rangle = N(x|0\rangle + |1\rangle)$ or $|\psi(x)\rangle = N(|00\rangle + (x)|11\rangle)$, where N is the normalization factor of the system. This form is because the degree of entanglement of that state is completely determined by x via its binary entropy function.

How our aims is to parametrize the **qRAM** neuron dynamics, in this dissertation we introduce two other ways to iterate the quantum system state. The main differences of the proposed iterated models in this dissertation from the other ones are (i) auxiliary qubits are in the dynamics and the output needs to be extracted to feedback the dynamics; (ii) the parameters of the dynamical system are the parameters of the neurons; (iii) the input system is the neuron input; (iv) input qubit state is free of predetermined format and it can be any qubit, in opposition of that is done in purification protocol dynamics.

1.3 Neurons Dynamics

Biological neurons receive the attention of dynamical systems researchers since the brain is a self-feedback system (BRAGA; CARVALHO; LUDERMIR, 2000). To understand the parameters, bifurcations and phase change of neurons help us to understand how information processing can be maximized in the brain activities (HERZ et al., 2006; PASEMANN, 1993; GARLIAUSKAS, 1998; KAK, 1995). In the artificial world, there is further evidence pointing agreements between chaotic environments and Artificial Neural Networks (ANN) learnability

(GROSS, Online em 2015). FREEMAN (1992); FREEMAN; JAKUBITH (1993) have argued that neuron populations are predisposed to instability and bifurcation that depend on external input and internal parameters.

Existing quantum neurons models are inspired by quantum computing or are quantum models ZHOU; DING (2007); PANELLA; MARTINELLI (2011); SILVA; OLIVEIRA; LUDERMIR (2012); PAULA NETO et al. (2013). They have been proposed to solve pattern recognition problems and in general machine learning tasks LI; ZHAO; ZHENG (2002); SILVA; OLIVEIRA; LUDERMIR (2012); MANJU; NIGAM (2014); LIN; CHUNG; CHEN (2007).

About neuron dynamics, BEHRMAN et al. (2000) proposed to use quantum dot molecules simulating a quantum neural computer - it is not a feed-forward network: all virtual neurons are connected to all other virtual neurons, both forward and backward in time, by effects of the environment. In that study, a training algorithm is associated with the energy minimization of the system inspired in the Hopfield network HOPFIELD (1984) and it is capable to training logic gates.

In (BEHERA; KAR; ELITZUR, 2005), a model of recurrent quantum neural network is proposed to described eye tracking of moving targets. In that model, the fitting of parameters is done by a feed-forward of the estimated amplitude wave output and by the desired amplitude wave output. The estimated output is encountered considering the quantum state as a probability density function and it is estimated by its expected value formula.

The study of quantum weightless neural networks was introduced by OLIVEIRA (2009), in the case of quantization of logical and probabilistic neurons, Probabilistic Logic Node (PLN), and the multi-valued probabilistic neuron, the Multi-valued Probabilistic Logic Node (MPLN) (LUDERMIR et al., 1989). Despite its simplicity, memory-based classical RAM neural networks have good generalization capacity and computing power (SOUTO; LUDERMIR; OLIVEIRA, 2005; DE SOUZA et al., 2009). The proposal to find the respective quantum model of these classical weightless networks is to link simplicity, computational capacity and quantum features - such as quantum parallelism. In recent work, there are training proposals for quantum weightless neurons using quantum superposition and demonstrating its comprehensive computational power (SILVA; LUDERMIR; DE OLIVEIRA, 2012).

1.4 Objectives

This dissertation aims to characterize the dynamical of the Quantum RAM Based Node (qRAM) (OLIVEIRA, 2009) on three aspects: (i) to investigate its dynamics: it was proposed one method of dynamical qubit extraction methods (Chapter 2) and we propose more one method that allows a complex values manipulations (Chapter 5); (ii) to perform an algebraic analysis of the qRAM dynamics: we analyse what are the equations of the dynamics and its behaviours: a study of extracted analytical equations of dynamics (Chapter 3); and finally (iii) demonstrating how to solve complex problems in computation (Chapter 4). Each objective is presented and

discussed in a different chapter in the form of technical reports published and accepted for journal publication.

Finally, Chapter 6 is the conclusion, where it is possible to enumerate the results of this study.

2

Chaos in Quantum Weightless Neuron Node Dynamics

One flap of a sea gull's wings would be enough to alter the course of the weather forever.

—EDWARD LORENZ

In this chapter, we present a qRAM dynamics study. The paper published at the *Neurocomputing Journal* (PAULA NETO et al., 2016) discusses the qubit extraction limitation after quantum operators application. We show that in quantum controlled operations the system information can entangle itself in a quantum state, being impossible decompose it, isolating the operated qubits (Theorem 6.1). We discuss the extraction dynamics method of information of the qubits amplitudes, the Extraction Dynamics model, which it will have phase loss, but it will be possible to iterate the quantum neuron (Section 6.2). In this dissertation, the Extraction Dynamics model is also named Real Extraction Model (REM). The proposed extraction method creates a nonlinear condition in the dynamics which it assumes 3 possible behaviour: underdamped, overdamped, undamped / oscillatory (Section 6.2.2). An evaluation measure to compare different dynamics is discussed (Section 6.3) and some models of dynamics is compared quantitatively.

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Chaos in Quantum Weightless Neuron Node Dynamics

Fernando M. de Paula Neto [a,*], Wilson R. de Oliveira [b], Adenilton J. da Silva [b,a], Teresa B. Ludermir [a]

[a] Centro de Informática, Universidade Federal de Pernambuco, Brazil
[b] Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco, Brazil

ARTICLE INFO

Article history:
Received 31 August 2014
Received in revised form
29 January 2015
Accepted 28 February 2015
Available online 30 December 2015

Keywords:
Quantum computing
Weightless neural networks
Quantum neural networks
Chaos

ABSTRACT

In order to investigate the dynamics of a quantum weightless neuron node we feed its output back as input. Due to the fact that controlled operators used in the neuron circuit usually generate entanglement, we propose a mathematical method to extract the output at time t and build from that output the input at time $t+1$. As a result the time evolution is a real-valued nonlinear map with one real parameter. The dynamics orbits are plotted showing acute sensitivity to initial conditions clearly exhibiting nonlinearity by just looking at amplitude graphs. The fractal geometry and regions of convergence are discussed by their Julia Set images and a new measure for model comparison is put forward.

1. Introduction

We are surrounded by complexity and non-linearity. They emerge from the interactions of systems of either the same or different kinds. Biological systems, weather phenomena, fluid turbulence, radar backscatter from the sea surface, multipath in mobile communication systems and control systems are examples of complex systems. Research in dynamical systems has increased in the past few years in order to understand these systems from initial conditions and their asymptotic behaviour [1,2] with the increase in power of the computer systems.

Poincaré studied the three-body problem when he discovered that small perturbations can significantly affect the solution [3]. Concepts of phase portrait, Poincaré section, periodic orbit, return map, bifurcation and fixed point were first introduced by Poincaré as key descriptive aspects of dynamical systems. The first representation of a chaotic attractor was provided by Edward Norton Lorenz [4] in his attempt to understand weather forecasting through numerical solutions in systems of differential equations.

Since then, important advances in computer graphics, fractals and physics stimulated developments in the field of dynamical systems. Many systems are understood in detail and have been classified into categories according to their number of variables and non-linearity [5].

Chaos in classical neural node [6,7] and networks [8] have been reported in the literature. Evidences for the importance of chaos in natural and artificial brain have been collected in a short survey by Dave Gross in an electronically available article and in the references there [9].

Closed quantum systems are linear (unitary) and the majority of quantum computing literature deals with unitary evolution despite the apparent difficulty of physically isolating quantum systems [10]. In its turn open and measurement based systems can be nonlinear [11–13]. Notwithstanding the traditional unitary approach in quantum computing many studies have been carried out employing nonlinear operators as gates [14]. We should mention that the assumption that a fully quantised system evolution is not sensitive to initial conditions is nevertheless not unanimously accepted and sensitivity to initial conditions of physically realisable fully quantised system has been controversially reported in [15–19]. Measurement of quantum systems affects their dynamics and a non-linear behaviour can emerge from the systems [20,12]. This nonlinear behaviour has serious consequences in the dynamics of the systems bringing chaotic patterns into consideration. Another line of work in this field but not pursued here is the study of quantum systems that are classically chaotic [21].

Some quantum algorithms are naturally iterative but commonly implemented in acyclic circuits subordinated by a classical control. For example the Grover algorithm is the $\Theta(\sqrt{n})$ repetition of the Grover operator G [22]. Grover algorithm can be understood as a set of quantum operators that, through a closed loop, reapplies the output in the input, and the qubits are measured after $\Theta(\sqrt{n})$ times of iteration. Another physical system intrinsically iterative is the control system of a quantum robot [23,24] interacting with the environment for navigation or identification,

* Corresponding author.
E-mail address: fmpn2@cin.ufpe.br (F.M. de Paula Neto).

http://dx.doi.org/10.1016/j.neucom.2015.02.103
0925-2312/

where a quantum computer controls its operations. Despite not being the concern here, when studying cyclic networks of quantum gates is important to comprehend their relation to the halting problem for Turing machines. In acyclic networks of gates it is possible to determine if a algorithm will stop in contrast to cyclic network of arbitrary complexity [25].

Feedback control in quantum computing usually can be performed in the following way. A measurement is performed in some quantum registers and the measurement result is used as feedback. For instance, this strategy is used in [26]. In this paper, we are interested in the alternative method proposed in [27], where it is shown that quantum information in cyclic networks can be beneficial when there is no measurement.

One can understand the dynamics of quantum cyclic networks under the point of view of their operators, its phase analysis, extracting eigenvalues and eigenstates [25]. Studies about weak measurements back into the dynamics of ensemble of quantum systems were presented by Lloyd and Slotine [28]. Conditional dynamics of qubits iterated by a unitary operator coupled with a measurement-induced nonlinearity is investigated by Kiss et al. [20] and shown to be exponentially sensitive to initial condition with positive Lyapunov demonstrating chaotic behaviour. The nonlinear operator employed arises in quantum state purification protocols where the nonlinear effects can guarantee the unconditional security of quantum cryptographic key distribution protocols. Quantum systems that interact with an environment through measurement can be chaotic and nonlinear [29–33].

In this work we show a set of experiments and analysis of the dynamics of the qRAM and $|\psi\rangle$-RAM quantum weightless neuron nodes which demonstrate high degree of sensitivity to the initial conditions and chaotic behaviour. For that, we have used the quantum operator of the respective node and a measurement induced nonlinear step. After a short review of the basic notions of Quantum Computing (Section 2), Dynamical Systems (Section 3), Classical Weightless Neural Networks (Section 4) and Quantum Weightless Neural Networks (Section 5) the proposed Models of Dynamics (Section 6) is presented where is given a proof that the target qubits after a generic controlled unitary operator cannot always be decomposable as a product of two isolated quantum states, *i.e.* they are entangled, Theorem 6.1. In Section 6.2.1 the method for mathematically extract the amplitudes of a (possibly) entangled states is presented while in Section 6.2.2 the experiments are explained and analysed. A mathematical procedure to recover the amplitudes of the output qubit is discussed. It is observed high sensitivity to initial conditions and chaotic behaviour. After that, the results are analysed under the perspective of Amplitude Graph and a quantitative study is introduced by a measure of variation.

2. Quantum computing

One *quantum bit* (qubit) is a two-dimensional vector in the complex vector space $\mathbb{C}^2$. Any qubit $|\psi\rangle$ can be written as a linear combination of vectors (or states) of $\mathbb{C}^2$ canonical (or computational) basis $|0\rangle=[1,0]^T$ and $|1\rangle=[0,1]^T$ as viewed in the following equation:

$$|\psi\rangle=\alpha|0\rangle+\beta|1\rangle \tag{1}$$

where α and β are complex numbers and $|\alpha|^2+|\beta|^2=1$.

Tensor product $\otimes$ is used to represent quantum systems with two or more qubits $|ij\rangle=|i\rangle\otimes|j\rangle$. Let A and B be two vector spaces the tensor product of A and B, denoted by $A\otimes B$, is the vector space generated by the tensor product of all vectors $|a\rangle\otimes|b\rangle$, with $|a\rangle\in A$ and $|b\rangle\in B$. Some states $|\psi\rangle\in A\otimes B$ cannot be written as a product of states of its component systems A and B. States with this property are called *entangled* states, for instance two entangled qubits are the Bell states described in the following equation:

$$\begin{aligned}|\Phi^+\rangle&=\frac{|00\rangle+|11\rangle}{\sqrt{2}}\\ |\Phi^-\rangle&=\frac{|00\rangle-|11\rangle}{\sqrt{2}}\\ |\Psi^+\rangle&=\frac{|01\rangle+|10\rangle}{\sqrt{2}}\\ |\Psi^-\rangle&=\frac{|01\rangle-|10\rangle}{\sqrt{2}}\end{aligned} \tag{2}$$

Quantum operator **U** over n qubits is a unitary complex matrix of order $2^n\times 2^n$. For example, some operators over 1 qubit are Identity **I**, NOT **X** and Hadamard **H**, described below in Eqs. (3) and (4) in matrix form and operator form. The combination of these unitary operators forms a quantum circuit.

$$\mathbf{I}=\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{matrix}\mathbf{I}|0\rangle=|0\rangle\\ \mathbf{I}|1\rangle=|1\rangle\end{matrix},\quad \mathbf{X}=\begin{bmatrix}0&1\\1&0\end{bmatrix}\begin{matrix}\mathbf{X}|0\rangle=|1\rangle\\ \mathbf{X}|1\rangle=|0\rangle\end{matrix} \tag{3}$$

$$\mathbf{H}=\frac{1}{\sqrt{2}}\begin{bmatrix}1&1\\1&-1\end{bmatrix},\quad \begin{matrix}\mathbf{H}|0\rangle=1/\sqrt{2}(|0\rangle+|1\rangle)\\ \mathbf{H}|1\rangle=1/\sqrt{2}(|0\rangle-|1\rangle)\end{matrix} \tag{4}$$

The Identity operator **I** generates the output exactly as the input; **X** operator works as the classic NOT in the computational basis; Hadamard **H** generates a superposition of states when applied in a computational basis. The CNOT operator has 2 inputs and 2 outputs and flips the second one if the first is 1, as shown in Fig. 1.

The operators of quantum computation can be seen as special kinds of linear transformations, as matrices that operates in a vector basis. These special matrices are unitary and invertible [34].

3. Dynamical systems

Systems that have variation in time can be usually dealt with mathematical structures having time as parameter. This time iterative process is the subject of the field Dynamical Systems where there are many tools and concepts that help designers and engineers to investigate the temporal behaviour of systems. Some of these concepts are presented in this section to help understanding and evaluating the models investigated in this work.

3.1. Orbits

There are many problems in Science in general and in Mathematics in particular that involve iteration [5]. Iteration means to repeat a process many times. In dynamics the process that is repeated is the application of a function. The result of the application of a function in previous time is used as input in the same function in the current time.

Given $x_0\in\mathbf{R}$, we define the orbit of x_0 under F to be the sequence of points $x_0, x_1=F(x_0), x_2=F(x_1),\cdots x_n=F(x_{n-1})\cdots$. The point x_0 is called "seed" of the orbit.

Sometimes it is useful to deal with a family of functions parametrised by a constant and so it is normal to represent it as $F_c(z)$ where c is a constant. As example, we have $F_c(z)=z^2+c$, and $F_2(z)=z^2+2$, where $c=2$. This representation helps us to categorise these families of functions.

3.2. Julia Set

Julia Set is the place where every chaotic behaviour of a complex function occurs [35]. For example, the squaring map $Q_0(z)=z^2$ is chaotic in the unit circle, because if $|z|<1$ then $|Q_0^n(z)|\to 0$,

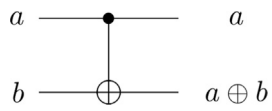


Fig. 1. CNOT operator.

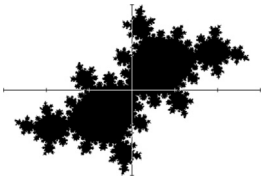


Fig. 2. Filled Julia Set of the $Q_c(z)=z^2+c, c=-0.159054-0.58679i$.

when $n\to+\infty$, and if $|z|>1$ then $|Q_0^n(z)|\to+\infty$ when $n\to+\infty$. $Q_0(z)$ is a notation of the orbital 0 to $Q(z)$.

Through the *Filled Julia Set* is possible to identify fractals whose the nature is very peculiar. The Filled Julia Set of Q_c is the set of points whose orbits are limited. The Julia Set of Q_c is the limit of the filled Julia Set. $K_0=z||z|\le 1$ and $J_0=z||z|=1$ are, respectively, Filled Julia Set and Julia Set examples. An example of Filled Julia Set can be observed in Fig. 2.

The simplified algorithm to find the Filled Julia Set is described in Algorithm 1.

Algorithm 1. Algorithm to find an approximation of Filled Julia Set.

1 Choose a maximum number N of iterations.
2 For each point of z in the grid, compute the first N points of the orbital z. If $|Q_c^i(z)|>\max\{|c|,2\}$, for some $i\le N$, then stop iterating and colour z white.
3 If $|Q_c^i(z)|\le\max\{|c|,2\}$, for all $i\le N$, then colour z black.
4 The white points represent the orbital that escaped, and the black points that ones did not, at least in the first N iterations.
5 So, the black points are the approximation of the Filled Julia Set.

4. Weightless neural network

The first model of Weightless Neural Networks (WNN) was proposed and investigated by Igor Aleksander [36]. WNN are the models of neural computation which have binary inputs and outputs and there are no weights between their nodes. The weightless neuron is based on the simple operations of a look-up table which is best implemented by random access memory (RAM) and where the knowledge is directly "stored" in the memory (via "look-up tables") of the nodes during learning. Learning on these systems generally consists of changing the contents of lookup table entries, which results in highly flexible and fast learning algorithms.

4.1. RAM node

In contrast to biologically motivated nodes, RAM nodes were initially designed as engineering tools to solve pattern recognition problems. An N input RAM node (Fig. 3) has 2^N memory locations, addressed by the N-bit vector $\mathbf{a}=a_1,a_2,\ldots,a_N$. A binary signal $\mathbf{x}=x_1,x_2,\ldots,x_N$ on the input lines will access only one of these locations, that is, the one for which $\mathbf{a}=\mathbf{x}$. The bit, $C[\mathbf{x}]$, stored at this activated memory represents the output of the node, that is, $y=C[\mathbf{x}]$. In other words, the Boolean function performed by the

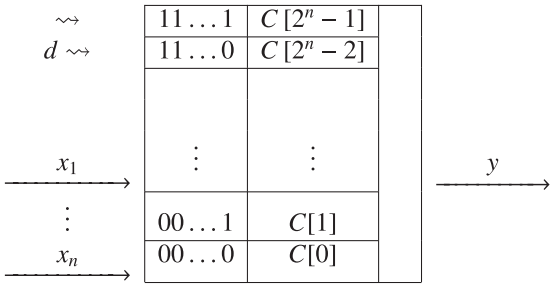


Fig. 3. RAM Node. Given an input **x**, the RAM Node outputs a bit y stored in the $C[x]$.

neuron is determined by the contents of the RAM. There are 2^{2^N} different functions which can be performed on N address lines and these correspond to the 2^N states that the RAM node can be in. Thus, a single RAM can compute any Boolean function of its inputs. Despite its simplicity, RAM nodes have good computational power [37].

4.2. Probabilistic logic node PLN

A PLN [38] node differs from a RAM node in that a 2-bit number (rather than a single bit) is now stored at the addressed memory location. The content of this location is turned into the probability of firing (i.e., generating 1) at the overall output of the node. In other words, a PLN consists of a RAM node augmented with a probabilistic output generator. Thus, like in a RAM node, the N binary inputs to a PLN node form an address to one of the 2^N addressable locations. Simple RAM nodes then output the stored value directly.

In contrast, in a PLN, the content at this address is passed through the output function which converts it into a binary node output. Such a content could be either 0's, 1's, or u's. The undefined state u implies on the node flipping its output between 0 and 1 with equal probability. The use of a third logic value, u (undefined), makes possible the use of an "unknown" state in the operation of WNNs architectures. This value is stored in all the memory contents before the learning phase, indicating the ignorance of the network before it was trained. The output of the PLN node is given by:

$$y=\begin{cases}0 & \text{if } C[\mathbf{x}]=0\\ 1 & \text{if } C[\mathbf{x}]=1\\ \text{random}(0,1) & \text{if } C[\mathbf{x}]=u\end{cases} \tag{5}$$

5. Quantum weightless neuron node

Out of many proposed models of a quantum neuron in the literature [39–47] some work with actual quantum strategies abiding the laws of Quantum Physics while others are only quantum *inspired* strategies. The latter are classical models of computation that uses ideas from quantum computing.

In this section we present the qRAM Node, the qPLN Node and the $|\psi\rangle$-RAM, as quantisation of the weightless nodes from the previous section.

5.1. qRAM node

A quantisation of the RAM neuron was proposed in [48,49] by de Oliveira et al. through a technique called *Mathematical Quantisation* [50]. The fundamental idea of this method is that sets are replaced with Hilbert spaces. The elements of a set are put in a

one-to-one correspondence with a orthonormal basis of the Hilbert space.

The RAM Node stores in a memory position a unique bit. In the initial description of the qRAM Node [49] the matrix A simulates the behaviour of a classical RAM as can be seen in the example below in the case of one input and one output:

$$A=\begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} \quad \begin{array}{l}\text{where} \\ A|0\rangle|y\rangle = |0\rangle I|y\rangle \\ A|1\rangle|y\rangle = |1\rangle X|y\rangle\end{array} \tag{6}$$

We can interpret the matrix A as an operator which "selects" which one of the X or I operator is going to be applied to the second qubit accordingly weather the first qubit is respectively $|0\rangle$ or $|1\rangle$. We call then the first qubit *selector register* or just *selector* and the second qubit *parameter register* or just *parameter*. By setting the parameter register to $|0\rangle$, A can be seen as a sort of memory returning the *memory content* $|0\rangle$ or $|1\rangle$ if the selector is respectively $|0\rangle$ or $|1\rangle$. In this case we call A a *memory matrix*. To change the memory content is to change the value of the selector register.

A qRAM is a collection of input addressable matrix memories. The input $|i\rangle$ "selects" or "addresses" the memory matrix A_i which returns its memory content. An example of the one qubit addressable qRAM is depicted in Fig. 4. The memory A_0 or A_1 is addressed weather the input qubit $|\psi\rangle$ is respectively $|0\rangle$ or $|1\rangle$. The *output register* $|o\rangle$ depends on the value of the selector $|s_i\rangle$ as explained above. Note that a general state, in a superposition of the basis states, in the input may address both memories simultaneously resulting in a superposition of their contents.

The qRAM of n input qubits has a set of 2^n of that A's, consequently it has 2^n selectors but one qubit of output $|o\rangle$ since we are not dealing here with multiqubits stored values. A representation of the circuit can be seen in Fig. 4. A collection of CNOT operators helps us to choose which of the operators A_i, $i=1,\ldots,2^n$, will be activated depending on the input qubit as can be seen in [34, p. 185]. A classical and quantum learning that change the values of the selectors is proposed in [51].

5.2. qPLN

In de Oliveira et al. [48] is shown that applying a Hadamard operator in the output qubit the qRAM can simulate a PLN Node. qPLN Node is proposed with more features beyond the PLN Node. The qPLN model in the circuit representation is shown in Fig. 5 where the matrix A is detailed in Eq. (7). The randomness of the PLN is ensured through the use of the state $|u\rangle = H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, for quantum principles [34].

$$A=\begin{pmatrix} I & 0 & 0 & 0 \\ 0 & X & 0 & 0 \\ 0 & 0 & H & 0 \\ 0 & 0 & 0 & U \end{pmatrix} \quad \begin{array}{l}\text{where} \\ A|000\rangle = |00\rangle I|0\rangle \\ A|010\rangle = |01\rangle X|0\rangle \\ A|100\rangle = |10\rangle H|0\rangle \\ A|110\rangle = |11\rangle U|0\rangle\end{array} \tag{7}$$

With N input qubits, the qPLN has 2^{N+1} selectors $|s\rangle$, 1 output qubit $|o\rangle = |0\rangle$ and 2^N matrices A. If the selectors are in computational basis, the qPLN Node applies the Identity operator (if $|s\rangle = |00\rangle$), or apply the X operator (if $|s\rangle = |01\rangle$), or apply

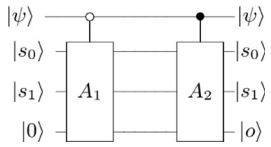


Fig. 4. qRAM Node of 1 input.

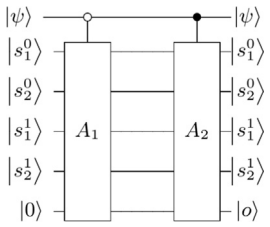


Fig. 5. qPLN Node of 1 input.

Hadamard operator (if $|s\rangle = |10\rangle$).[1] When the selectors are in superposition, the node activates more than one operation and executes a parallel process applying these operators. A qPLN training algorithm is proposed in [48]. A representation of qPLN circuit can be seen in Fig. 5.

5.3. $|\psi\rangle$-RAM

Da Silva et al. [52] generalises the qPLN node in such a way that it becomes a universal quantum gate. They proved in [52] that qPLN Node, with the parameter register being always $|0\rangle$, cannot implement the Hadamard gate. In the $|\psi\rangle$-RAM proposed in [52] the parameter register can take any value and now it is possible to show that this node simulates the Hadamard and Toffoli operators [52] thus being universal for quantum computation.

6. Chaotic models of dynamics

The unitary of quantum transformations prevents exponential sensitivity to initial conditions, despite the controversy surrounding the following so-called exceptions [15–19]. However, measurement in quantum systems affects its dynamics [20] and a nonlinear behaviour can emerge from the system. Bechmann et al., in [13], propose a measurement based nonlinear quantum transformation and they argue that though this transformation is nonlinear (and it does not preserve the trace), it is physically realisable.

Before we present our model of dynamics, we explain the workings of the dynamics over one qubit proposed in [20] which has inspired our model. After that, we present the proof that it is not possible to verify completely the quantum state of the target bit after an U-Controlled Operator over 2 qubits. Then our method is presented and analysed both analytically and experimentally.

6.1. KJAV model of dynamics over one qubit

Kiss et al. [20] propose a model which we dub KJAV model to analyse the dynamics over one qubit coupled with the nonlinear operator proposed by Bechmann et al. [13]. This operator is employed to distinguish optimally between two non-orthogonal spin-1/2 states. This dynamics is mapped by Eq. (8), where N is the function that normalises the qubit with a factor. In this qubit, the factor is $1/\sum \rho_{ij}^2$.

$$\rho = S\rho, \quad \rho_{ij} \xrightarrow{S} N\rho_{ij}^2 \tag{8}$$

Thus, if we have one qubit, its transformation S proposed by Bechmann et al. is:

$$|\psi\rangle_{input} = \alpha|0\rangle + \beta|1\rangle \xrightarrow{S} |\psi\rangle_{output} = N(\alpha^2|0\rangle + \beta^2|1\rangle) \tag{9}$$

[1] Since we must have a power of two many block 2×2 matrices in A we fill it in with an extra possibly not used arbitrary 2×2 unitary U.

Kiss et al. [20] propose to include this transformation S during the dynamics over one qubit. The rotation operator U is a generic rotation operator in the Hilbert space, with x and ϕ variables, as shown in the following equation:

$$U(x,\phi)=\begin{pmatrix}\cos(x) & \sin(x)e^{i\phi}\\ -\sin(x)e^{-i\phi} & \cos(x)\end{pmatrix} \tag{10}$$

Given a $|\psi\rangle$ qubit, a initial pure state:

$$|\psi\rangle = N(z|0\rangle+|1\rangle) \tag{11}$$

where N is the renormalisation factor $1/\sqrt{1+|z|^2}$. The quadratic operator S [13] and a generic rotation operator are applied. The application of these operations are as described below:

$$\begin{aligned}&|\psi\rangle = N(z|0\rangle+|1\rangle)\xrightarrow{S}|\psi_2\rangle = N(z^2|0\rangle+|1\rangle)\\ &U|\psi_2\rangle = |\psi_3\rangle = N((z^2\cos(x)+\sin(x)e^{i\phi})|0\rangle\\ &\qquad +(-\sin(x)e^{-i\phi}z^2+\cos(x))|1\rangle)\end{aligned} \tag{12}$$

The state $|\psi_3\rangle$ is the quantum state after the first iteration. To recovery the z value after this dynamics, it is necessary to transform the state to the original format of pure state.

$$|\psi_{output}\rangle = N(z'|0\rangle+|1\rangle) \tag{13}$$

In $|\psi_3\rangle$ qubit, the $|0\rangle$ amplitude value is divided by the $|1\rangle$ amplitude value, the value of z after one iteration is encountered:

$$z' = \frac{z^2\cos(x)+\sin(x)e^{i\phi}}{-\sin(x)e^{-i\phi}z^2+\cos(x)} \tag{14}$$

It is easy to see that the normalisation rate does not need to be considered because it will be cancelled after the division of the amplitudes.

Considering $p=\tan(x)e^{i\phi}$, the analytic formula of this dynamical model that is studied in [20] is:

$$F_p(z)=\frac{z^2+p}{-p^*z^2+1} \tag{15}$$

Then the dynamics proposed in [20] has one free variable z and one constant p whose initial condition determines the existence or not of chaos during its dynamics.

6.2. Extraction model dynamics

The qRAM Node and the ψ-RAM Node are composed of a circuit with various Controlled Operators. We will review here that we do not have a canonical way to always recover the target qubit after a U-Controlled Operator, for a unitary U, and in particular the CNOT Operator. We next state and prove a well known result but that has been neither stated nor proved in a general formulation as we give below.

Theorem 6.1. *Superposed target qubits after the application of Controlled-U operators can become entangled.*

Proof. If we consider the qubits $|u\rangle$ and $|v\rangle$:

$$|u\rangle = a|0\rangle+b|1\rangle,\quad |v\rangle = c|0\rangle+d|1\rangle \tag{16}$$

Their tensor product is:

$$|u\rangle|v\rangle = ac|00\rangle+ad|01\rangle+bc|10\rangle+bd|11\rangle \tag{17}$$

If we apply to this product state an U-Controlled operation, where U is a generic rotation operator:

$$U(\alpha)=\begin{pmatrix}\cos(\alpha) & -\sin(\alpha)\\ \sin(\alpha) & \cos(\alpha)\end{pmatrix} \tag{18}$$

Then:

$$A_U|u\rangle|v\rangle = \begin{pmatrix}ac\\ ad\\ bc*\cos(\alpha)-bd*\sin(\alpha)\\ bc*\sin(\alpha)+bd*\cos(\alpha)\end{pmatrix} \tag{19}$$

The qubit $|u\rangle$ should not change for this operation and it seems simple to think that the second qubit $|v\rangle$ can be recovered in its quantum state. But this is not possible in general unless the operator U is very restricted.

Considering then that only the second qubit is modified with arbitrary value $|v\rangle_{t+1}=x|0\rangle+y|1\rangle$ and that the qubit $|u\rangle$ has not changed. Their tensor product is: $|uv_{t+1}\rangle = ax|00\rangle+ay|01\rangle+bx|10\rangle+by|11\rangle$. Equating $A_U|u\rangle|v\rangle$ with $|uv_{t+1}\rangle$, under the condition that $|u\rangle$ and $|v\rangle$ are not in the basis states, we have $c=x$, $d=y$, $c*\cos(\alpha)-d*\sin(\alpha)=x$ and $c*\sin(\alpha)+d*\cos(\alpha)=y$.

As we can see, the terms in function of the first qubit disappear the final conditions and there are only conditions over the second qubit. One of the conditions is that the target qubit before and after the operation of the U-Controlled should be equal. This is only possible if $\alpha=0$, in other words, $U=I$ (Identity operator) and that $c=d=x=y$, where $|u\rangle=|v\rangle=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle)$ are equal and have equal amplitudes in their states.□

With that restrictions, how can we iterate the output qubit if we cannot directly recovery the amplitudes of the output qubit $|o\rangle$? The possibility to generate entangled states in general after the application of an operator is also problematic since we cannot isolate the output after this operation.

6.2.1. Extracting the output qubit

The proposed method extracts the output qubit amplitude values through a mathematical algorithm. For example, if we have a state of 2 qubits $|\phi\rangle = a_0|00\rangle+a_1|01\rangle+a_2|10\rangle+a_3|11\rangle$, the quantum theory claims that the second qubit can take with probability $|a_0|^2+|a_2|^2$ the value $|0\rangle$ and $|a_1|^2+|a_3|^2$ the value $|1\rangle$.

So in general, to obtain the last qubit amplitudes for $|0\rangle$ and $|1\rangle$ one takes the square root of the sum of the probability of all the n-qubit states ending with $|0\rangle, |x_1,..,x_{n-2},0\rangle$ and all ending with $|1\rangle, |x_1,\ldots,x_{n-2},1\rangle$, respectively. For the probability one sums up the squared norm of the even positions of the vector in the $|0\rangle$ case. The same is done with odd positions for the $|1\rangle$. Thus, the output $|o\rangle$ is calculated as:

$$|o\rangle = \alpha|0\rangle+\beta|1\rangle \tag{20}$$

where,

$$|\alpha|^2=\sum_{i_{even}}|\psi_i|^2 \quad\text{and}\quad |\beta|^2=\sum_{i_{odd}}|\psi_i|^2 \tag{21}$$

Amongst the solutions (positive or negative real, complex) to Eq. (20) we use the simplest one by taking the positive real solution:

$$\alpha=\sqrt{\sum_{i_{even}}|\psi_i|^2} \quad\text{and}\quad \beta=\sqrt{\sum_{i_{odd}}|\psi_i|^2}$$

An example is shown below to recover the amplitudes in qRAM node dynamics if registers $|i\rangle, |s\rangle=|s_0\rangle|s_1\rangle$ and $|o\rangle$ are:

$$|i\rangle=\begin{pmatrix}i_0\\ i_1\end{pmatrix}, |s_0\rangle=\begin{pmatrix}s_0^0\\ s_0^1\end{pmatrix},\quad |s_1\rangle=\begin{pmatrix}s_1^0\\ s_1^1\end{pmatrix}, |o\rangle=\begin{pmatrix}\alpha\\ \beta\end{pmatrix} \tag{22}$$

where all amplitudes are complex numbers. The $|\psi\rangle=|i\rangle|s\rangle|o\rangle$ as

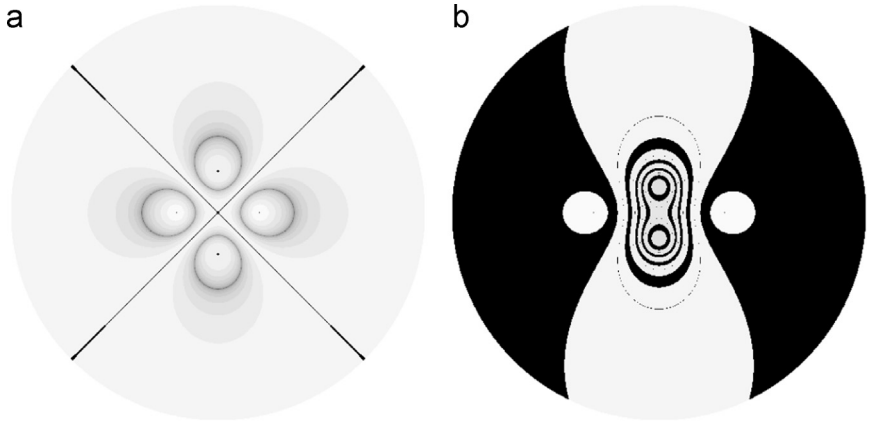


Fig. 6. Filled Julia Set of the $|\psi\rangle$-EM Dynamics to the some different c-values of the generic operator U. In this figure, dark regions indicate fast convergence, grey regions represent slow convergence and white regions indicate no convergence. (a) $|\psi\rangle$-EM-1 Dynamics where $\tan(t_U)=1.0$ and $|\mathbf{s}\rangle = |11\rangle$. (b) $|\psi\rangle$-EM-2 Dynamics where $\tan(t_U)=1.0$ and $|\mathbf{s}\rangle = |11\rangle$.

entry of the qRAM Node is:

$$|\psi\rangle = \begin{pmatrix} i_0 * s_0^0 * s_1^0 * \alpha \\ i_0 * s_0^0 * s_1^0 * \beta \\ i_0 * s_0^0 * s_1^1 * \alpha \\ i_0 * s_0^0 * s_1^1 * \beta \\ i_0 * s_0^1 * s_1^0 * \alpha \\ i_0 * s_0^1 * s_1^0 * \beta \\ i_0 * s_0^1 * s_1^1 * \alpha \\ i_0 * s_0^1 * s_1^1 * \beta \\ i_1 * s_0^0 * s_1^0 * \alpha \\ i_1 * s_0^0 * s_1^0 * \beta \\ i_1 * s_0^0 * s_1^1 * \alpha \\ i_1 * s_0^0 * s_1^1 * \beta \\ i_1 * s_0^1 * s_1^0 * \alpha \\ i_1 * s_0^1 * s_1^0 * \beta \\ i_1 * s_0^1 * s_1^1 * \alpha \\ i_1 * s_0^1 * s_1^1 * \beta \end{pmatrix} \tag{23}$$

When applying the entry $|\psi\rangle$ in the qRAM Node, we have the following output:

$$qRAM|\psi\rangle = \begin{pmatrix} i_0 * s_0^0 * s_1^0 * \alpha \\ i_0 * s_0^0 * s_1^0 * \beta \\ i_0 * s_0^0 * s_1^1 * \alpha \\ i_0 * s_0^0 * s_1^1 * \beta \\ i_0 * s_0^1 * s_1^0 * \beta \\ i_0 * s_0^1 * s_1^0 * \alpha \\ i_0 * s_0^1 * s_1^1 * \beta \\ i_0 * s_0^1 * s_1^1 * \alpha \\ i_1 * s_0^0 * s_1^0 * \alpha \\ i_1 * s_0^0 * s_1^0 * \beta \\ i_1 * s_0^0 * s_1^1 * \beta \\ i_1 * s_0^0 * s_1^1 * \alpha \\ i_1 * s_0^1 * s_1^0 * \alpha \\ i_1 * s_0^1 * s_1^0 * \beta \\ i_1 * s_0^1 * s_1^1 * \beta \\ i_1 * s_0^1 * s_1^1 * \alpha \end{pmatrix} \tag{24}$$

The output qubit amplitudes can be extracted via Eq. (21):

$$\begin{aligned} |\alpha|^2_{t+1} = {} & |i_0 s_0^0 s_1^0 \alpha|^2 + |i_0 s_0^0 s_1^1 \alpha|^2 + |i_0 s_0^1 s_1^0 \beta|^2 + |i_0 s_0^1 s_1^1 \beta|^2 \\ & + |i_1 s_0^0 s_1^0 \alpha|^2 + |i_1 s_0^0 s_1^1 \beta|^2 + |i_1 s_0^1 s_1^0 \alpha|^2 + |i_1 s_0^1 s_1^1 \beta|^2 \end{aligned} \tag{25}$$

$$\begin{aligned} |\beta|^2_{t+1} = {} & |i_0 s_0^0 s_1^0 \beta|^2 + |i_0 s_0^0 s_1^1 \beta|^2 + |i_0 s_0^1 s_1^0 \alpha|^2 + |i_0 s_0^1 s_1^1 \alpha|^2 \\ & + |i_1 s_0^0 s_1^0 \beta|^2 + |i_1 s_0^0 s_1^1 \alpha|^2 + |i_1 s_0^1 s_1^0 \beta|^2 + |i_1 s_0^1 s_1^1 \alpha|^2 \end{aligned} \tag{26}$$

resulting in the output qubit of the next iteration:

$$|o\rangle_{t+1} = \alpha_{t+1}|0\rangle + \beta_{t+1}|1\rangle \tag{27}$$

The output qubit can be entangled as seen in Theorem 6.1. This way of obtaining the amplitudes of the output qubit is just a mathematical operation which will help us to analyse the system as a mathematical entity but it is not at all physically realisable operation since the amplitudes cannot be checked or obtained in general quantum physical systems. We can only obtain an approximation of the amplitudes by repetitive (finitely many) measurements, according to the Quantum Mechanics Postulates. Note that after the extraction step the output qubit must have real amplitudes as a result of our approach.

6.2.2. Experiments and analysis

The proposed Extraction Model (EM) uses the recover procedure presented in the last section to feed the output qubit back into the quantum node. We can do it in two ways: (a) in the EM-1 Dynamics the output qubit $|o\rangle$ is fed back into itself and (b) in the EM-2 Dynamics the output qubit $|o\rangle$ is fed back into output and input qubits.

We run two sets of experiments. In the first set, the $|\psi\rangle$-RAM Filled Julia Set having the U operator of the $|\psi\rangle$-RAM node as parameter for a family of functions is generated and depicted in Fig. 6. The feedback value is always a real number, due to our extraction method, and the variable z of the dynamics obtained from the output qubit $|o\rangle = N(z|0\rangle + |1\rangle)$ as the independent variable is used, inspired by the KJAV model [20]. When the parameter register is initially set to $|0\rangle$ no chaos is observed in the dynamics since the behaviour is restricted to a subspace of the solution set in which all orbits that have seeds $a+bj$ are in the Filled Julia Set, where $a, b \in [-2, 2]$. So we consider that dynamics starts with the parameter register initialised with $N(z|0\rangle + |1\rangle)$. During the dynamics Bechmann et al.'s nonlinear operator [13] is applied to the input, the result of which is fed to the node. In Fig. 6 the (grey) color, symmetry and shapes show the

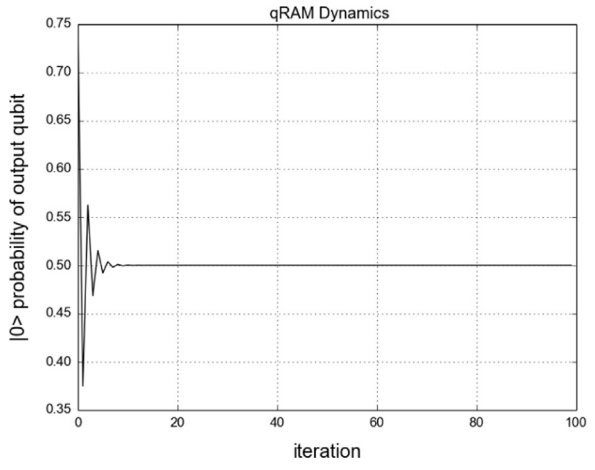


Fig. 7. qRAM-EM-1, initial condition: $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|s_0\rangle = |1\rangle$, $|s_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|o\rangle = |1\rangle$.

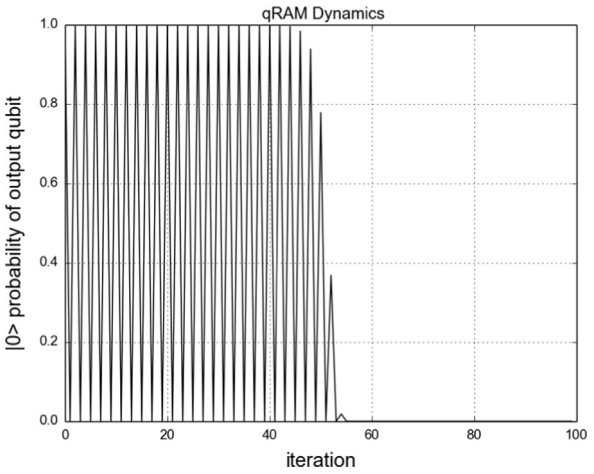


Fig. 9. qRAM-EM-2, initial condition: $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|s_0\rangle = |1\rangle$. $|s_1\rangle = |1\rangle$, $|o\rangle = |1\rangle$.

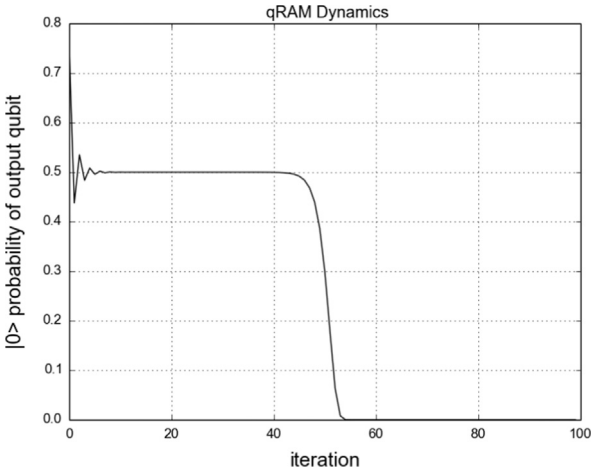


Fig. 8. qRAM-EM-2, initial condition: $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|s_0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|s_1\rangle = |1\rangle$, $|o\rangle = |1\rangle$.

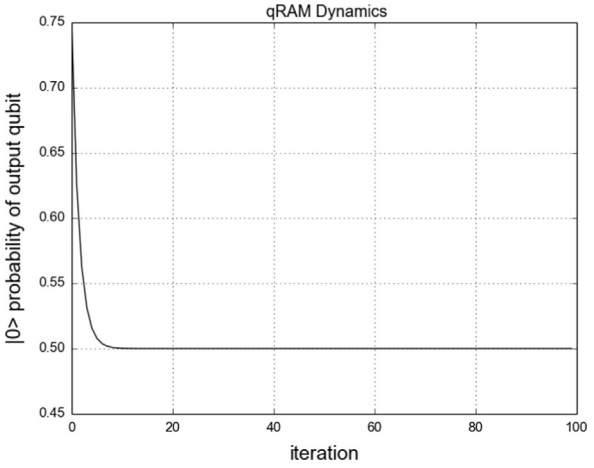


Fig. 10. qRAM-EM-1, initial condition: $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|s_0\rangle = |0\rangle$, $|s_1\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|o\rangle = |0\rangle$.

sensitivity to initial conditions and the boundary regions exhibit where there is convergence.

In the second set of experiments, we analyse the dynamic of a single qRAM and a single $|\psi\rangle$-RAM both with one input for EM-1 and EM-2 models totalising four experiment configurations. Input, output and selectors registers were initialised with the qubits $|0\rangle$, $|1\rangle$ and $H|0\rangle$ totalising 81 runs for each experiment configuration. In the Appendix four tables are presented containing a summary of statistical information for each possible input, one table for each neural model and each extraction model.

The output qubit amplitudes during the dynamics are plotted for the $|\psi\rangle$-RAM model and qRAM model. The orbits can be classified having three main behaviours – undamped, under-damped and over-damped. Small sensitivity to initial conditions and sensitivity to the amplitudes decimal expansion values during the dynamic are clearly depicted. Some orbits are plotted to show the sensitivity to initial conditions of the models.

The under-damped behaviour in the qRAM node can be seen in Figs. 7–9. As one can see in Fig. 7 the dynamics converges to one value and stays in this fixed value. On the other hand, in Fig. 8, the under-damped dynamics abruptly change its convergence to the zero value. This sensitivity is presented with the decimal values variation and next to 40th iteration, a fast damped is presented.

In Fig. 9, the damping is not fast and has a long settling time, presenting a non-oscillating behaviour, once the dynamics is limited by the vertical axis in its damping. During the first iterations, the dynamics is undamped but again next 40th iteration, the decimal values variations change the dynamics to an under-damped dynamics, showing the robustness of the sensitivity of its values in dynamics.

Fig. 10 shows one case of the over-damped dynamics, showing an exponential damping. The permanent oscillation, the undamped case, is presented in Fig. 11.

In $|\psi\rangle$-RAM node, the three behaviours are also encountered varying the rise time, settling time and maximum values. Fig. 12 shows the over-damped situation where the increasing and decreasing behaviour of the over-damped are presented. Next to the 30th iteration the dynamics is sensitive to the decimal values and appears to converge, however it does not occur and the dynamics is exponentially damped.

In almost all cases of the qRAM EM-1 and EM-2 dynamics, the orbits converge in the first iteration. The summations of that dynamics are in the majority cases 50.0, in the qRAM EM-1, and

next to 25.0 in the qRAM EM-2. In both qRAM models, the value of convergence is in most cases 0.5. The orbits of the $|\psi\rangle$-RAM models EM-1 and EM-2 are presented respectively. The convergence values have more diversity in $|\psi\rangle$-RAM than qRAM models. The values of convergence in $|\psi\rangle$-RAM model are not necessarily 0.5 or 1.0 as in the qRAM dynamics, and have more diversity of values.

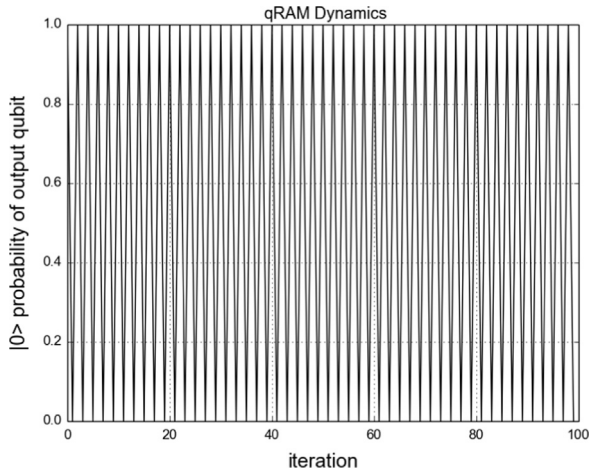


Fig. 11. qRAM-EM-1, initial condition: $|i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, $|s_0\rangle = |1\rangle$, $|s_1\rangle = |1\rangle$, $|o\rangle = |1\rangle$.

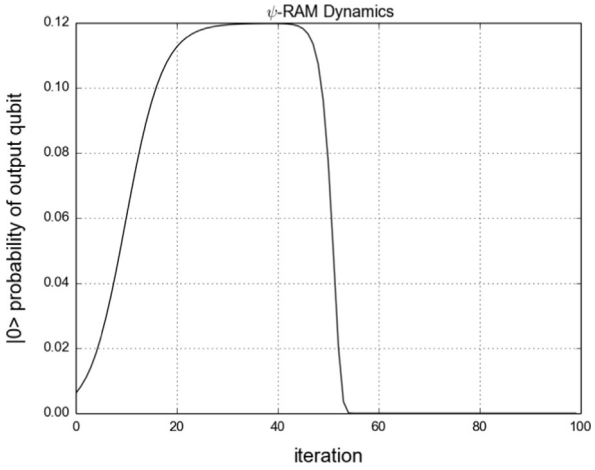


Fig. 12. $|\psi\rangle$-RAM-EM-2, initial condition: $|i\rangle = |0\rangle$, $|s_0\rangle = |s_1\rangle = |1\rangle$, $|s_2\rangle = |s_3\rangle = |0\rangle$, $|o\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$.

In EM-1 Dynamics, the qRAM model has average values next to 0.5 and 1.0, despite the $|\psi\rangle$-RAM model behaviour that has varying averages. In EM-2 Dynamics, the qRAM model has average values next to 0.25, 0.5 and 1.0. $|\psi\rangle$-RAM model has diversity of averages. The EM-2 Dynamics have more minimum values next to 0.0 than EM-1 Dynamics in both neuron models.

Fixing the initial input register to a predetermined value and varying the selectors values, we can examine the convergence behaviour of the dynamics of the point of view of the selectors. Figs. 13 and 14 show the density graph of quantity convergence for two different values of the selectors and two initial input register configuration. Fig. 13 presents the quantity of amplitude convergent cases varying the selector $|s_0\rangle$ for qRAM model and varying $|s_0\rangle = |s_1\rangle$ for ψ-RAM model. Fig. 14 is the same approach in the case of variation of $|s_1\rangle$ for the qRAM model and the variation of $|s_2\rangle = |s_3\rangle$ for ψ-RAM model.

As can be seen, the central values of probability, next to 0.5, are the configuration which has more convergence cases, to both selector variations, and in the extreme values the divergent cases occur more. This is because the central values for selectors cause more superposition of the operators applied in the qubit once the selectors assigned in central values are more in superposition. Extreme values to selectors decrease the participation of more operators in the qubit dynamics.

6.3. δ-Value measure

In order to compare the models in a quantitative way, we propose a variation measure, δ-Value. This measure represents the degree of change of the amplitude values during the dynamics. It is defined as the sum of the derivatives at the sampled points:

$$\delta = \sum_t \frac{\partial \alpha(t)}{\partial t} \tag{28}$$

And in its absolute representation:

$$\delta_{\text{abs}} = \sum_t \left| \frac{\partial \alpha(t)}{\partial t} \right| \tag{29}$$

where $\partial\alpha(t)/\partial t$ is calculated by numerical approximation

$$\frac{\partial \alpha(t)}{\partial t} = \frac{\alpha[t+\Delta t] - \alpha[t]}{\Delta t} \tag{30}$$

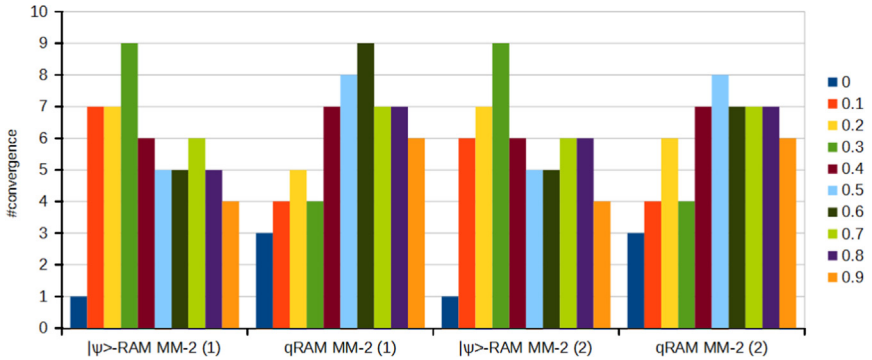


Fig. 13. Density graph to represent the quantity of convergent orbits of the Dynamics Models in the case of variation of the selector values $|s_0\rangle$ in the qRAM model and $|s_0\rangle = |s_1\rangle$ in the ψ-RAM model.

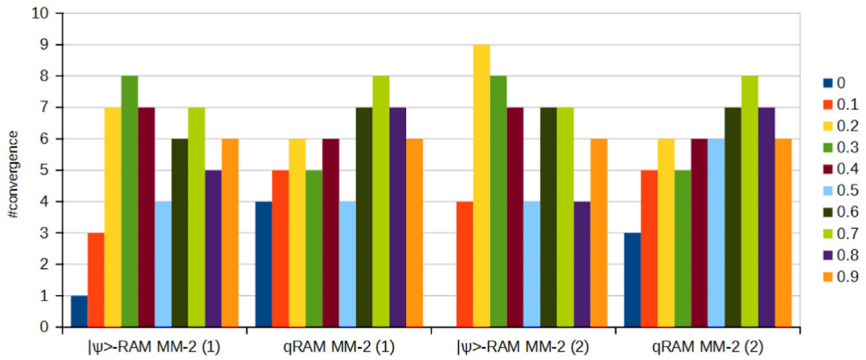


Fig. 14. Density graph to represent the quantity of convergent orbits of the Dynamics Models in the case of variation of the selector values $|s_1\rangle$ in the qRAM model and $|s_2\rangle = |s_3\rangle$ in the ψ-RAM model.

Table 1
Table of the maximum and minimum values, average and variance of the δ-Value considering 81 initial configurations.

Model	Max (δ)	Min (δ)	Avg (δ)	Var (δ)
q-EM-1	1.0	−1.0	0	0.18
q-EM-2	1.0	−1.0	−0.40	0.18
$\|\psi\rangle$-EM-1	0.15	−0.67	−0.09	0.06
$\|\psi\rangle$-EM-2	0	−1.0	−0.41	0.07

Table 2
Table of the maximum and minimum values, average and variance of the δ_{abs}-Value considering 81 initial configurations.

Model	Max (δ_{abs})	Min (δ_{abs})	Avg (δ_{abs})	Var (δ_{abs})
q-EM-1	99.0	0.0	17.16	1399.55
q-EM-2	99.0	0.0	6.70	504.23
$\|\psi\rangle$-EM-1	41.78	0.0	8.73	226.48
$\|\psi\rangle$-EM-2	21.83	0.0	2.42	29.07

considering Δt one step of iteration and α is the $|0\rangle$ output qubit probability.

Varying the registers $|i\rangle, |s_0\rangle, |s_1\rangle$ and $|o\rangle$ as $H|0\rangle, |0\rangle$ and $|1\rangle$, 81 initial configurations of the qRAM are investigated. As $|\psi\rangle$-RAM has 4 selectors, we repeat the selector values in pairs: $|s_0\rangle = |s_1\rangle$ and $|s_2\rangle = |s_3\rangle$ and 81 initial configuration experiments are realised.

From these 81 initial configurations for each model, we extracted the maximum and minimum values, the average and variance of the δ-Value and δ_{abs}-Value. Tables 1 and 2 show the results.

δ-Values in the EM-2 Dynamics have greater variance (in magnitude) than in EM-1 Dynamics. On the other hand, the variance in EM-1 Dynamics is greater than EM-2 Dynamics in the δ_{abs}-Value mainly because the qRAM has basically CNOT operators that makes to oscillate the output qubit amplitudes.

The negative averages in δ-Value imply that the dynamics behaviour is tendentiously decreasing. When we disregard the derivative sign, i.e. considering δ_{abs}-Value, the variance, average and the maximum values are greater than δ-Value, since the sum does not have minus signs to cancel the positive ones.

7. Conclusion

In this work, we proposed a method to iterate quantum weightless neuron nodes by mathematically extracting the qubit to be fed back. The fact that we cannot generally recover the output qubit amplitudes after an application of a Controlled-Operator restricts the recovery procedure of the output qubit of a neuron node. The extraction procedure proposed makes the dynamics sensitive to initial conditions and, through Amplitude Graphs, we visualised nonlinearity and curves highly sensitive to decimal expansion values. The Julia Set images presented showed areas upon which the independent variable is more sensitive to initial conditions.

As future work, we intend to investigate the dynamics of more than one neuron. A numerical analysis in terms of sensitivity to initial conditions and a general classification of the dynamics taking into account many parameters of these models are also envisaged. The applications of nonlinear and chaotic methods to quantum computing seem to be a promising line of research particularly by the results that nonlinear quantum mechanics can solve NP-complete problems in polynomial many steps [14,53].

The search for a physically realisable nonlinear quantum operator and their mathematical classification may lead to real breakthrough in Quantum and Classical Complexity Theory.

Acknowledgements

This work is supported by research grants from CNPq, CAPES and FACEPE (Brazilian research agencies).

Appendix A. Orbits table

Tables A1 and A2 present details of the orbits from qRAM EM-1 and EM-2 dynamics respectively. And the orbits of the $|\psi\rangle$-RAM models EM-1 and EM-2 are presented respectively in Tables A3 and A4.

Table A1
qRAM EM-1 Dynamics Results – values of maximum, minimum, summation, average, variance of the orbits of the iteration of each initial conditional configuration. The convergence is assigned True if the difference of two orbit values is less than 10^{-10}. The iteration and the value of the convergence are showed.

Initial configuration	Max	Min	Sum	Avg	Var	Converging	Iteration of convergence	Value of convergence
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|000H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0010\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0011\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|001H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|00H_00\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|00H_01\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|00H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0100\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0101\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|010H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0110\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0111\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|011H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|01H_00\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|01H_01\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|01H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_000\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_001\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_00H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_010\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_011\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_01H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_0H_00\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_0H_01\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_0H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|100H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1010\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1011\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|101H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|10H_00\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|10H_01\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|10H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1100\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1101\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|110H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1110\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1111\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|111H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|11H_00\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|11H_01\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|11H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_00H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_010\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_011\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_01H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_0H_00\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_0H_01\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_0H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_00H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0010\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0011\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_01H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_00\rangle$	0.75	0.5	50.5	0.505	0.08083	True	31.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_01\rangle$	0.5	0.25	49.5	0.495	0.08083	True	31.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0100\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5

Table A1 (*continued*)

Initial configuration	Max	Min	Sum	Avg	Var	Converging	Iteration of convergence	Value of convergence
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0101\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_010H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0110\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0111\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_011H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_01H_00\rangle$	0.625	0.25	49.83333	0.49833	0.08306	True	32.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_01H_01\rangle$	0.75	0.375	50.16667	0.50167	0.08306	True	32.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_01H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_000\rangle$	0.75	0.5	50.5	0.505	0.08083	True	31.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_001\rangle$	0.5	0.25	49.5	0.495	0.08083	True	31.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_00H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_010\rangle$	0.625	0.25	49.83333	0.49833	0.08306	True	32.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_011\rangle$	0.75	0.375	50.16667	0.50167	0.08306	True	32.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_01H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_00\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_01\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5

Table A2
qRAM EM-2 Dynamics Results – values of maximum, minimum, summation, average, variance of the orbits of the iteration of each initial conditional configuration. The convergence is assigned True if the difference of two orbit values is less than 10^{-10}. The iteration and the value of the convergence are showed.

Initial configuration	Max	Min	Sum	Avg	Var	Converging	Iteration of convergence	Value of convergence
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|000H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0010\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0011\rangle$	1.0	0.0	99.0	0.99	0.99	True	1.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|001H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|00H_00\rangle$	1.0	0.0	50.21329	0.50213	23.99955	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|00H_01\rangle$	0.5	0.0	24.79114	0.24791	5.99956	True	1.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|00H_0H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0100\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0101\rangle$	1.0	0.0	1.0	0.01	0.99	True	1.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|010H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0110\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0111\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|011H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|01H_00\rangle$	0.5	0.0	24.79114	0.24791	5.99956	True	1.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|01H_01\rangle$	1.0	0.0	25.17995	0.2518	6.49968	True	2.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|01H_0H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_000\rangle$	0.5	0.0	25.58363	0.25584	5.99659	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_001\rangle$	0.5	0.0	25.58363	0.25584	5.99659	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_00H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_010\rangle$	0.5	0.0	25.58363	0.25584	5.99659	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_011\rangle$	0.5	0.0	25.58363	0.25584	5.99659	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_01H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_0H_00\rangle$	0.5	0.0	24.82152	0.24822	5.99968	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_0H_01\rangle$	0.5	0.0	24.82152	0.24822	5.99968	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|0H_0H_0H_0\rangle$	0.5	0.0	24.82152	0.24822	5.99968	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|100H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1010\rangle$	1.0	0.0	99.0	0.99	0.99	True	1.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1011\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|101H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|10H_00\rangle$	0.5	0.0	25.58363	0.25584	5.99659	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|10H_01\rangle$	0.5	0.0	25.58363	0.25584	5.99659	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|10H_0H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1100\rangle$	1.0	0.0	1.0	0.01	0.99	True	1.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1101\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|110H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1110\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1111\rangle$	1.0	0.0	50.0	0.5	25.0	False	–	–

Table A2 (*continued*)

Initial configuration	Max	Min	Sum	Avg	Var	Converging	Iteration of convergence	Value of convergence
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|111H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|11H_00\rangle$	0.5	0.0	25.58363	0.25584	5.99659	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|11H_01\rangle$	0.5	0.0	25.58363	0.25584	5.99659	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|11H_0H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_000\rangle$	1.0	0.0	25.79114	0.25791	6.49374	True	1.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_00H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_010\rangle$	1.0	0.0	25.17995	0.2518	6.49968	True	2.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_011\rangle$	1.0	0.0	25.79114	0.25791	6.49374	True	1.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_01H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_0H_00\rangle$	0.5	0.0	24.82152	0.24822	5.99968	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_0H_01\rangle$	0.5	0.0	24.82152	0.24822	5.99968	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|1H_0H_0H_0\rangle$	0.5	0.0	24.82152	0.24822	5.99968	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0000\rangle$	1.0	0.0	51.66725	0.51667	23.9722	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_00H_0\rangle$	0.5	0.0	25.44794	0.25448	5.99799	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_010\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_011\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_01H_0\rangle$	0.5	0.0	25.44794	0.25448	5.99799	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_00\rangle$	0.75	0.0	25.82282	0.25823	6.49118	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_01\rangle$	0.5	0.0	24.86352	0.24864	5.88456	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0100\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0101\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_010H_0\rangle$	0.5	0.0	25.44794	0.25448	5.99799	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0110\rangle$	1.0	0.0	25.58355	0.25584	18.53853	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0111\rangle$	1.0	0.0	26.08371	0.26084	18.77995	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_011H_0\rangle$	0.5	0.0	25.44794	0.25448	5.99799	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_01H_00\rangle$	0.5625	0.0	25.09566	0.25096	5.96191	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_01H_01\rangle$	0.75	0.0	25.18704	0.25187	6.1513	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_01H_0H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_00\rangle$	0.75	0.0	25.57731	0.25577	6.24667	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_01\rangle$	0.5	0.0	24.36585	0.24366	5.74598	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_010\rangle$	0.6875	0.0	24.99026	0.2499	6.0533	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_011\rangle$	0.75	0.0	25.51967	0.2552	6.1713	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_01H_0\rangle$	0.5	0.0	25.08363	0.25084	5.99993	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_00\rangle$	0.5	0.0	24.82152	0.24822	5.99968	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_01\rangle$	0.5	0.0	24.82152	0.24822	5.99968	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|o\rangle = \|H_0H_0H_0H_0\rangle$	0.5	0.0	24.76893	0.24769	5.99947	True	0.0	0.5

Table A3
ψ-RAM EM-1 Dynamics Results – values of maximum, minimum, summation, average, variance of the orbits of the iteration of each initial conditional configuration. The convergence is assigned True if the difference of two orbit values is less than 10^{-10}. The iteration and the value of the convergence are showed.

Initial configuration	Max	Min	Sum	Avg	Var	Converging	Iteration of convergence	Value of convergence
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|00000H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000110\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000111\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|00011H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000H_0H_00\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000H_0H_01\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000H_0H_0H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011000\rangle$	0.57797	0.02432	15.20368	0.15204	1.72195	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011001\rangle$	0.42203	0.0	21.10141	0.21101	4.45269	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|01100H_0\rangle$	0.34596	0.00612	17.60406	0.17604	2.8874	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011110\rangle$	0.57797	0.02432	15.20368	0.15204	1.72195	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011111\rangle$	0.42203	0.0	21.10141	0.21101	4.45269	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|01111H_0\rangle$	0.34596	0.00612	17.60406	0.17604	2.8874	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011H_0H_00\rangle$	0.57797	0.02432	15.20368	0.15204	1.72195	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011H_0H_01\rangle$	0.42203	0.0	21.10141	0.21101	4.45269	False	–	–
	0.34596	0.00612	17.60406	0.17604	2.8874	False	–	–

Table A3 (*continued*)

Initial configuration	Max	Min	Sum	Avg	Var	Converging	Iteration of convergence	Value of convergence
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011H_0H_0H_0\rangle$								
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0000\rangle$	0.51949	0.50159	50.17776	0.50178	0.00032	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0001\rangle$	0.50159	0.48051	50.13719	0.50137	0.00044	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_000H_0\rangle$	0.50159	0.50153	50.15906	0.50159	0.0	True	5.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0110\rangle$	0.51949	0.50159	50.17776	0.50178	0.00032	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0111\rangle$	0.50159	0.48051	50.13719	0.50137	0.00044	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_011H_0\rangle$	0.50159	0.50153	50.15906	0.50159	0.0	True	5.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0H_0H_00\rangle$	0.51949	0.50159	50.17776	0.50178	0.00032	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0H_0H_01\rangle$	0.50159	0.48051	50.13719	0.50137	0.00044	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0H_0H_0H_0\rangle$	0.50159	0.50153	50.15906	0.50159	0.0	True	5.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|10000H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100110\rangle$	0.57797	0.02432	15.20368	0.15204	1.72195	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100111\rangle$	0.42203	0.0	21.10141	0.21101	4.45269	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|10011H_0\rangle$	0.34596	0.00612	17.60406	0.17604	2.8874	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100H_0H_00\rangle$	0.51949	0.50159	50.17776	0.50178	0.00032	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100H_0H_01\rangle$	0.50159	0.48051	50.13719	0.50137	0.00044	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100H_0H_0H_0\rangle$	0.50159	0.50153	50.15906	0.50159	0.0	True	5.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|11100H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111110\rangle$	0.57797	0.02432	15.20368	0.15204	1.72195	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111111\rangle$	0.42203	0.0	21.10141	0.21101	4.45269	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|11111H_0\rangle$	0.34596	0.00612	17.60406	0.17604	2.8874	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111H_0H_00\rangle$	0.51949	0.50159	50.17776	0.50178	0.00032	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111H_0H_01\rangle$	0.50159	0.48051	50.13719	0.50137	0.00044	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111H_0H_0H_0\rangle$	0.50159	0.50153	50.15906	0.50159	0.0	True	5.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_000H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0110\rangle$	0.57797	0.02432	15.20368	0.15204	1.72195	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0111\rangle$	0.42203	0.0	21.10141	0.21101	4.45269	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_011H_0\rangle$	0.34596	0.00612	17.60406	0.17604	2.8874	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0H_0H_00\rangle$	0.51949	0.50159	50.17776	0.50178	0.00032	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0H_0H_01\rangle$	0.50159	0.48051	50.13719	0.50137	0.00044	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0H_0H_0H_0\rangle$	0.50159	0.50153	50.15906	0.50159	0.0	True	5.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_000000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_000001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_00000H_0\rangle$	0.5	0.5	50.0	0.5	0.0	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_000110\rangle$	0.78899	0.11988	13.13405	0.13134	0.56728	True	6.0	0.11988
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_000111\rangle$	0.21101	0.11988	12.09072	0.12091	0.00833	True	4.0	0.11988
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_00011H_0\rangle$	0.25306	0.11988	12.14449	0.12144	0.01801	True	4.0	0.11988
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_00H_0H_00\rangle$	0.75975	0.50159	50.69607	0.50696	0.08835	True	32.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_00H_0H_01\rangle$	0.50159	0.24025	49.61494	0.49615	0.09063	True	32.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_00H_0H_0H_0\rangle$	0.50159	0.50076	50.15741	0.50157	0.0	True	24.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_011000\rangle$	0.78899	0.11988	13.13405	0.13134	0.56728	True	6.0	0.11988
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_011001\rangle$	0.21101	0.11988	12.09072	0.12091	0.00833	True	4.0	0.11988
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_01100H_0\rangle$	0.25306	0.11988	12.14449	0.12144	0.01801	True	4.0	0.11988
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_011110\rangle$	0.57797	0.02432	15.20368	0.15204	1.72195	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_011111\rangle$	0.42203	0.0	21.10141	0.21101	4.45269	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_01111H_0\rangle$	0.34596	0.00612	17.60406	0.17604	2.8874	False	–	–
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_011H_0H_00\rangle$	0.54873	0.25974	26.34257	0.26343	0.08222	True	11.0	0.26055
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_011H_0H_01\rangle$	0.45127	0.25025	26.23706	0.26237	0.03615	True	12.0	0.26055
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_011H_0H_0H_0\rangle$	0.26174	0.25382	26.04936	0.26049	5e-05	True	11.0	0.26055
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_000\rangle$	0.75975	0.50159	50.69607	0.50696	0.08835	True	32.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_001\rangle$	0.50159	0.24025	49.61494	0.49615	0.09063	True	32.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_00H_0\rangle$	0.50159	0.50076	50.15741	0.50157	0.0	True	24.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_0110\rangle$	0.54873	0.25974	26.34257	0.26343	0.08222	True	11.0	0.26055
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_0111\rangle$	0.45127	0.25025	26.23706	0.26237	0.03615	True	12.0	0.26055
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_011H_0\rangle$	0.26174	0.25382	26.04936	0.26049	5e-05	True	11.0	0.26055
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_0H_0H_00\rangle$	0.51949	0.50159	50.17776	0.50178	0.00032	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_0H_0H_01\rangle$	0.50159	0.48051	50.13719	0.50137	0.00044	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0H_0H_0H_0H_0H_0\rangle$	0.50159	0.50153	50.15906	0.50159	0.0	True	5.0	0.50159

Table A4
ψ-RAM EM-2 Dynamics Results – values of maximum, minimum, summation, average, variance of the orbits of the iteration of each initial conditional configuration. The convergence is assigned True if the difference of two orbit values is less than 10^{-10}. The iteration and the value of the convergence are showed.

Initial configuration	Max	Min	Sum	Avg	Var	Converging	Iteration of convergence	Value of convergence
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|00000H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000110\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000111\rangle$	0.42203	0.0	6.6133	0.06613	0.48011	True	58.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|00011H_0\rangle$	0.5	0.0	6.76432	0.06764	0.56439	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000H_0H_00\rangle$	1.0	0.0	49.91844	0.49918	23.99993	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000H_0H_01\rangle$	0.50159	0.0	24.4529	0.24453	6.01344	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|000H_0H_0H_0\rangle$	0.50159	0.0	24.82886	0.24829	6.03596	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011000\rangle$	0.57797	0.0	7.21512	0.07215	0.67313	True	58.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011001\rangle$	0.42203	0.0	7.014	0.07014	0.53157	True	58.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|01100H_0\rangle$	0.11981	0.0	4.80489	0.04805	0.28175	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011110\rangle$	0.57797	0.0	8.2827	0.08283	1.55907	True	58.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011111\rangle$	0.42203	0.0	11.12623	0.11126	3.36908	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|01111H_0\rangle$	0.34596	0.0	8.80801	0.08808	2.15839	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011H_0H_00\rangle$	0.57797	0.0	16.58469	0.16585	2.79488	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011H_0H_01\rangle$	0.42203	0.0	16.53523	0.16535	2.69112	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|011H_0H_0H_0\rangle$	0.48016	0.0	16.38449	0.16384	2.72023	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_000\rangle$	0.51949	0.0	24.87717	0.24877	6.05647	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_001\rangle$	0.50159	0.0	24.81217	0.24812	6.0156	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_00H_0\rangle$	0.50159	0.0	24.89069	0.24891	6.03783	True	18.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0110\rangle$	0.51949	0.0	7.9796	0.0798	0.77056	True	15.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0111\rangle$	0.48051	0.0	7.92639	0.07926	0.72922	True	17.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_011H_0\rangle$	0.50153	0.0	7.94106	0.07941	0.75103	True	16.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0H_0H_00\rangle$	0.51949	0.0	24.59575	0.24596	6.04557	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0H_0H_01\rangle$	0.50159	0.0	24.60256	0.24603	6.02542	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|0H_0H_0H_0H_0H_0\rangle$	0.50159	0.0	24.53733	0.24537	6.03528	True	5.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100000\rangle$	1.0	1.0	100.0	1.0	0.0	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|10000H_0\rangle$	0.5	0.0	25.94794	0.25948	5.99101	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100110\rangle$	0.57797	0.0	6.94702	0.06947	0.67807	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100111\rangle$	0.42203	0.0	6.6133	0.06613	0.48011	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|10011H_0\rangle$	0.34389	0.0	6.21134	0.06211	0.42096	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100H_0H_00\rangle$	0.51949	0.0	24.93063	0.24931	6.05768	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100H_0H_01\rangle$	0.50159	0.0	24.80868	0.24809	6.01663	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|100H_0H_0H_0\rangle$	0.50159	0.0	24.92745	0.24927	6.03796	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111000\rangle$	1.0	0.0	8.21512	0.08215	1.51883	True	59.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|11100H_0\rangle$	0.5	0.0	6.99805	0.06998	0.60015	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111110\rangle$	0.57797	0.0	8.2827	0.08283	1.55907	True	58.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111111\rangle$	0.42203	0.0	11.12623	0.11126	3.36908	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|11111H_0\rangle$	0.34596	0.0	8.80801	0.08808	2.15839	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111H_0H_00\rangle$	0.51949	0.0	16.61976	0.1662	2.7492	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111H_0H_01\rangle$	0.48051	0.0	16.5853	0.16585	2.7231	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|111H_0H_0H_0\rangle$	0.50153	0.0	16.66332	0.16663	2.73674	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_000\rangle$	1.0	0.0	25.50404	0.25504	6.55349	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_00H_0\rangle$	0.50159	0.0	24.82542	0.24825	6.03595	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0110\rangle$	0.57797	0.0	8.0997	0.081	0.84109	True	17.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0111\rangle$	0.42203	0.0	7.83464	0.07835	0.67596	True	17.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_011H_0\rangle$	0.34679	0.0	7.62818	0.07628	0.62135	True	18.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0H_0H_00\rangle$	0.51949	0.0	24.5586	0.24559	6.0452	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0H_0H_01\rangle$	0.50159	0.0	24.52748	0.24527	6.02464	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|1H_0H_0H_0H_0H_0\rangle$	0.50159	0.0	24.53733	0.24537	6.03528	True	5.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_000000\rangle$	1.0	0.0	51.66725	0.51667	23.9722	True	0.0	1.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_000001\rangle$	0.0	0.0	0.0	0.0	0.0	True	0.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0000H_0\rangle$	0.5	0.0	25.44794	0.25448	5.99799	True	0.0	0.5
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_000110\rangle$	0.78899	0.0	7.70889	0.07709	1.35229	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_000111\rangle$	0.21101	0.0	6.1908	0.06191	0.3659	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0011H_0\rangle$	0.25306	0.0	6.26432	0.06264	0.37953	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_00H_0H_00\rangle$	0.75975	0.0	25.83173	0.25832	6.58231	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_00H_0H_01\rangle$	0.50159	0.0	24.68056	0.24681	5.91258	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_00H_0H_0H_0\rangle$	0.50159	0.0	24.82557	0.24826	6.03674	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_011000\rangle$	0.78899	0.0	7.30021	0.073	0.94741	True	57.0	0.0

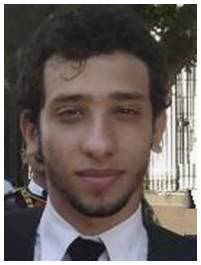
Table A4 (*continued*)

Initial configuration	Max	Min	Sum	Avg	Var	Converging	Iteration of convergence	Value of convergence
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 11001\rangle$	0.21101	0.0	6.49274	0.06493	0.39207	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 1100 H_0\rangle$	0.25306	0.0	6.49805	0.06498	0.41763	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 11110\rangle$	0.57797	0.0	7.93186	0.07932	1.52721	True	56.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 11111\rangle$	0.42203	0.0	10.95365	0.10954	3.3335	True	57.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 1111 H_0\rangle$	0.34596	0.0	8.71117	0.08711	2.14234	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 11 H_0 H_0 0\rangle$	0.54873	0.0	16.605	0.16605	2.7711	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 11 H_0 H_0 1\rangle$	0.45127	0.0	16.43846	0.16438	2.70564	True	54.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 11 H_0 H_0 H_0\rangle$	0.37498	0.0	16.42626	0.16426	2.6517	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 000\rangle$	0.75975	0.0	25.33199	0.25332	6.30605	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 001\rangle$	0.50159	0.0	24.15084	0.24151	5.75755	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 00 H_0\rangle$	0.50159	0.0	24.90932	0.24909	6.0371	True	55.0	0.0
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 110\rangle$	0.54873	0.0	8.01227	0.08012	0.80474	True	16.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 111\rangle$	0.45127	0.0	7.86117	0.07861	0.70142	True	17.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 11 H_0\rangle$	0.25382	0.0	7.6445	0.07644	0.57643	True	15.0	0.1515
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 H_0 H_0 0\rangle$	0.51949	0.0	24.59575	0.24596	6.04557	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 H_0 H_0 1\rangle$	0.50159	0.0	24.60256	0.24603	6.02542	True	6.0	0.50159
$\|i\rangle\|s_0\rangle\|s_1\rangle\|s_2\rangle\|s_3\rangle\|o\rangle = \|H_0 H_0 H_0 H_0 H_0 H_0\rangle$	0.50159	0.0	24.53733	0.24537	6.03528	True	5.0	0.50159

References

[1] L.A. Aguirre, C. Letellier, Modeling nonlinear dynamics and chaos: a review, Math. Probl. Eng. 2009 (2009) 35.
[2] S. Haykin, J. Principe, Making sense of a complex world [chaotic events modeling], Signal Process. Mag. 15 (1998) 66–81.
[3] J.H. Poincaré, J. Kovalevsky, Les Méthodes Nouvelles de la Mécanique Céleste, Grands classiques Gauthier-Villars, Blanchard, Paris, 1892.
[4] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (1963) 130–141.
[5] S.H. Strogatz, Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity), Westview Press, Cambridge, UK, 1994.
[6] A.V. M Herz, T. Gollisch, C.K. Machens, D. Jaeger, Modeling single-neuron dynamics and computations: a balance of detail and abstraction, Science 314 (5796) (2006) 80–85.
[7] F. Pasemann, Dynamics of a single model neuron, Int. J. Bifurc. Chaos 3 (02) (1993) 271–278.
[8] A. Garliauskas, Neural network chaos analysis, nonlinear anal, Model. Control 3 (02) (1998) 43–57.
[9] D. Gross, The Importance of Chaos Theory in the Development of Artificial Neural Systems. ⟨http://www.lycaeum.org/sputnik/Misc/chaos.html⟩.
[10] M. Schlosshauer, Decoherence, the measurement problem, and interpretations of quantum mechanics, Rev. Mod. Phys. 76 (2005) 1267–1305.
[11] D.R. Terno, Nonlinear operations in quantum-information theory, Phys. Rev. A 59 (5) (1999) 3320.
[12] T. Kiss, S. Vymětal, L. Tóth, A. Gábris, I. Jex, G. Alber, Measurement-induced chaos with entangled states, Phys. Rev. Lett. 107 (10) (2011) 100501.
[13] H. Bechmann-Pasquinucci, B. Huttner, N. Gisin, Non-linear quantum state transformation of spin-1/2, Phys. Lett. A 242 (10) (1998) 198–204.
[14] D.S. Abrams, S. Lloyd, Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and P problems, Phys. Rev. Lett. 81 (18) (1998) 3992–3995.
[15] B. Chirikov, F. Izrailev, D. Shepelyansky, Quantum chaos: localization vs. ergodicity, Physica D: Nonlinear Phenom. 33 (1–3) (1988) 77–88.
[16] M. Berry, True quantum chaos: an instructive example, in: Y. Abe, H. Horiuchi, K. Matsuyanagi (Eds.), New Trends in Nuclear Collective Dynamics, Springer Proceedings in Physics, vol. 58, Springer, Berlin, Heidelberg, 1992, pp. 183–186.
[17] R. Blümel, Exponential sensitivity and chaos in quantum systems, Phys. Rev. Lett. 73 (1994) 428–431.
[18] R. Schack, Comment on exponential sensitivity and chaos in quantum systems, Phys. Rev. Lett. 75 (1995) 581.
[19] R. Blümel, Blümel replies, Phys. Rev. Lett. 75 (1995) 582.
[20] T. Kiss, I. Jex, G. Alber, S. Vymetal, Complex chaos in the conditional dynamics of qubits, Phys. Rev. A 74 (2006) 23.
[21] H. Stöckmann, Quantum Chaos: An Introduction, Cambridge University Press, New York, USA, 2007.
[22] L.K. Grover, Quantum mechanics helps in searching for a needle in a haystack, Phys. Rev. Lett. 79 (2) (1997) 325–328.
[23] P. Benioff, Foundational aspects of quantum computers and quantum robots, Superlattices Microstruct. 23 (2) (1998) 407–417.
[24] P. Benioff, Quantum robots and environments, Phys. Rev. A 58 (2) (1998) 893–904.
[25] P. Cabauy, P. Benioff, Cyclic networks of quantum gates, Phys. Rev. A 68 (2) (2003) 032315.
[26] S. Gammelmark, K. Mølmer, Quantum learning by measurement and feedback, New J. Phys. 11 (3) (2009) 33017.
[27] S. Lloyd, Coherent quantum feedback, Phys. Rev. A 62 (2000) 022108.
[28] S. Lloyd, J.-J. Slotine, Quantum feedback with weak measurements, Phys. Rev. A 62 (2000) 012307.
[29] T. Kiss, I. Jex, G. Alber, E. Kollár, Properties of complex chaos in conditional qubit dynamics, Int. J. Quant. Inf. 06 (01) (2008) 695–700.
[30] J. Novotný, G. Alber, I. Jex, Random unitary dynamics of quantum networks, J. Phys. A: Math. Theor. 42 (28) (2009) 282003.
[31] G. Casati, B. Chirikov, I. Guarneri, D. Shepelyansky, Dynamical stability of quantum chaotic motion in a hydrogen atom, Phys. Rev. Lett. 56 (28) (1986) 2437–2440.
[32] K. Furuya, M. Nemes, G. Pellegrino, Quantum dynamical manifestation of chaotic behavior in the process of entanglement, Phys. Rev. Lett. 80 (28) (1998) 5524–5527.
[33] B.H. Nguyen, Quantum dynamics of quantum bits, Adv. Nat. Sci.: Nanosci. Nanotechnol. 2 (3) (2011) 033002.
[34] M.A. Nielsen, I.L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, UK, 2000.
[35] R.L. Devaney, A First Course in Chaotic Dynamical Systems: Theory and Experiment, Westview Press, Hanoi, Vietnam, 1992.
[36] I. Aleksander, Self-adaptive universal logic circuits, Electron. Lett. 2 (8) (1966) 321–322.
[37] M.C.P. de Souto, T.B. Ludermir, W.R. de Oliveira, Equivalence between RAM-based neural networks and probabilistic automata, IEEE Trans. Neural Netw. 16 (2005) 996–999.
[38] W.K. Kan, I. Aleksander., A probabilistic logic neuron network for associative learning, in: IEEE First International Conference on Neural Networks (ICNN87) II, 1987, pp. 541–548.
[39] A.J. da Silva, T.B. Ludermir, W.R. Oliveira, Superposition based learning algorithm, in: Brazilian Symposium on Neural Networks, 2010, pp. 1–6.
[40] M. Panella, G. Martinelli, Neural networks with quantum architecture and quantum learning, Int. J. Circuit Theory Appl. 39 (1) (2011) 61–77.
[41] M.V. Altaisky, Quantum Neural Network, Technical Report, Joint Institute for Nuclear Research, Russia, 2001.
[42] M. Schuld, I. Sinayskiy, F. Petruccione, The quest for a quantum neural network, Quant. Inf. Process. 13 (11) (2014) 2567–2586.
[43] P. Li, H. Xiao, F. Shang, X. Tong, X. Li, M. Cao, A hybrid quantum-inspired neural networks with sequence inputs, Neurocomputing 117 (2013) 81–90.
[44] M. Schuld, I. Sinayskiy, F. Petruccione, Quantum walks on graphs representing the firing patterns of a quantum neural network, Phys. Rev. A 89 (2014) 032333.
[45] B. Ricks, D. Ventura, Training a quantum neural network, in: S. Thrun, L. Saul, B. Schölkopf (Eds.), Advances in Neural Information Processing Systems, vol. 16, MIT Press, Montreal, Canada, 2004, pp. 1019–1026.
[46] P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. 113 (2014) 130503.
[47] F.M. De Paula, A. Silva, T. Ludermir, W.R. De Oliveira, Analysis of quantum neural models, in: XI Brazilian Congress of Computational Intelligence, vol. 1, 2013, pp. 1–6.
[48] W.R. de Oliveira, A.J. da Silva, T.B. Ludermir, A. Leonel, W.R. Galindo, J.C. Pereira, Quantum logical neural networks, in: Brazilian Symposium on Neural Networks, 2008, pp. 147–152.
[49] W. de Oliveira, Quantum RAM based neural networks, in: ESANN, 2009, pp. 22–24.
[50] N. Weaver, Mathematical Quantization, Studies in Advanced Mathematics, Taylor & Francis, Boca Raton, USA, 2001.

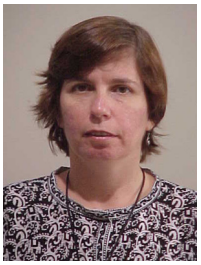
[51] A.J. da Silva, W.R. de Oliveira, T.B. Ludermir, Classical and superposed learning for quantum weightless neural networks, Neurocomputing 75 (1) (2012) 52–60.
[52] A.J. da Silva, T.B. Ludermir, W.R. de Oliveira, On the universality of quantum logical neural networks, in: 2012 Brazilian Symposium on Neural Networks, IEEE, Curitiba, Brazil, 2012, pp. 102–106.
[53] S. Aaronson, NP-complete problems and physical reality, Phys. Rev. A 74 (36) (2005) 30–52.



Fernando M. de Paula Neto received the Bachelor's degrees in computer engineering (cum laude) from the Center of Informatics, Universidade Federal de Pernambuco - UFPE, Brazil, in 2014. He is currently working toward the M.Sc. degree in Computer Science at the Universidade Federal de Pernambuco, Brazil. His current research interests include Dynamics Systems, Intelligent Computing, Neural Networks, Quantum Computing and Chaos.

Wilson R. de Oliveira has completed his B.SC degree (1982), M.Sc. degree (1985) and Ph.D. degree (2004) in Computer Science at Universidade Federal de Pernambuco (Brazil) specialising in Computing Theory of Artificial Neural Networks and Automata Theory. He is on sabbatical leave at the School of Computer Science, University of Birmingham, UK (2008) working on Matroid Theory and Discrete Differential Geometry. He has been working recently on Quantum Weightless Neural Networks, Combinatorial Hopf Algebras, Matroid Representations and Non-linearity and Chaos in Natural Temporal Series. He has published over 50 articles in scientific journals and conferences. He joined the "Departmento de Estatística e Informática" at the "Universidade Federal Rural de Pernambuco" in 2000 where he is now an Associate Professor.

Adenilton J. da Silva received the Licentiate degree in mathematics and M.Sc. degree in computer science from Universidade Federal Rural de Pernambuco, Brazil, in 2009, and Universidade Federal de Pernambuco, Brazil, in 2011, respectively. He is currently working toward the Ph.D. degree in computer science at the Universidade Federal de Pernambuco, Brazil. He joined the "Departmento de Estatística e Informática" at the "Universidade Federal Rural de Pernambuco" in 2014 where is now Assistant Professor. His current research interests include quantum computation, weightless neural networks and hybrid neural systems.



Teresa B. Ludermir received the Ph.D. degree in Artificial Neural Networks in 1990 from Imperial College, University of London, UK. From 1991 to 1992, she was a lecturer at Kings College London. She joined the Center of Informatics at Federal University of Pernambuco, Brazil, in September 1992, where she is currently a Professor and head of the Computational Intelligence Group. She has published over a 200 articles in scientific journals and conferences, three books in Neural Networks and organised two of the Brazilian Symposium on Neural Networks. She is one of the editors-in-Chief of the International Journal of Computation Intelligence and Applications. Her research interests include weightless Neural Networks, hybrid neural systems and applications of Neural Networks.

3

Fitting Parameters on Quantum Weightless Neuron Dynamics

In this chapter, we present our proposal of the analytical equations analysis for the qRAM neuron dynamics which uses the information extraction model from the qubit amplitudes, *Extraction Model Dynamics* (EMD), described in the previous Chapter. This paper was published in the *Brazilian Congress on Intelligent Systems* (BRACIS) 2015 (PAULA NETO et al., 2015a). We detail the system behaviour in the parameter space where it is possible visualize how the dynamics behaves itself to different possible selectors values (Section V). A general view of the dynamics behaviours according the selectors variation is shown (Section VI). The results from that work allow to understand the system behaviour in function of its parameters.

Fitting Parameters on Quantum Weightless Neuron Dynamics

Fernando M. de Paula Neto, Teresa B. Ludermir
Centro de Informática
Universidade Federal de Pernambuco
Recife, Pernambuco
Email: {fmpn2, tbl}@cin.ufpe.br

Wilson R. de Oliveira, Adenilton J. da Silva
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco
Recife, Pernambuco
Email: wilson.rosa@gmail.com, ajs@deinfo.ufrpe.br

***Abstract*—A parametric analysis of a quantum neuron node dynamics is proposed using an output qubit extraction protocol employed in previous works. The proposed classification method is composed of 5 different types of dynamics which explain the parameter changing behavior. The initial conditions analysis for each dynamics type is investigated. This parametric analysis gives a better understanding of the dynamics suggesting how fine-tune the system parameters can influence the system expected behaviour allowing *e.g.* for applications in patten recognition and information retrieval.**

***Keywords*-Dynamical systems; Control systems; Quantum computing; Convergent Dynamics; Weightless neuron; Learnability;**

I. Introduction

Complex and non-linear systems are everywhere around us, emerging from natural and artificial systems interacting with each other. Dynamical systems research has grown to understand these systems from their initial conditions and parameters aspects [1] [2].

The Poincaré three-body problem [3] and Lorenz work on weather forecasting through numerical solutions in systems of differential equations [4] landmarks in the field of Dynamical Systems, introducing the chaos concept where systems have sensibility of initial conditions. Lorenz showed how small changes in the initial conditions could lead to far apart differential equation solutions in such systems.

The studies of fractals and dynamical systems has had an exponential increase with the visualization tools introduced by computer graphics technologies and advances in Physics. Dynamical system have been categorized according to number of variables, nonlinearity and parameter behavior [5].

Many classical/quantum algorithms follow a cyclic iterative dynamics. For instance, the Grover algorithm repeats the Grover operator $\Theta(\sqrt{n})$ times [6]. Dynamics are important in quantum control systems [7] [8] as well in classical control systems. To feedback the results to guarantee error decreasing and system adjustment, we need to understand parameters behaviour in that system. One has to be careful since properties of cyclic algorithm can be reduced to te Halting Problems for Turing machines [9].

Neural networks have been used to solve a variety of problems in many areas. Quantum models of neuron and neural networks were proposed, using parallelism and other characteristics of quantum computation [10]. The iterative neuron dynamics process has been used to better understand the relationship of learnability and non linear and chaotic behavior of that dynamics. A short survey wrote by Dave Gross, in an electronically available article [11] elucidate the relationship chaos versus learning. A study of convergence and learnability is introduced in [12] from which some important concepts are used in this paper in terms of useful behaviors for neuron learning capability.

In [13], a qubit extraction model of a quantum neuron node was proposed. The extraction model dynamics presents great parametric sensibility. In this paper we give a detailed analysis of the system behavior under the extraction model dynamical system parameters configuration. The parameters are separated in regions of behavior each detailed in relation to the initial conditions and the changing of their parameters.

This paper is organized as follows. In Section II, we give basic concepts of control systems. In Section III, basic quantum computing concepts and the quantum weightless neuron nodes are presented. In Section IV, the extraction method of the quantum neuron node output is explained. The proposed model of parameters fitting is explained in Section V. Finally, the conclusion is presented in Section VII.

II. Control System

The basic concepts of control systems is presented here to fix the notation used in the analysis in the next sections. We follow [14] closely. A control system consists of subsystems and processes assembled for the purpose of obtaining a desired output with a desired performance, given a specified input [14]. The second order control systems are classified according to their

parameters which are related to their behavior in time, as shown in Figure 1.

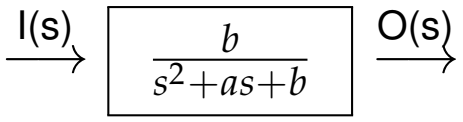


Figure 1. Second order system block diagram in the Laplacian representation where parameters 'a' and 'b' are not defined.

In the second order control systems, when the roots of the denominator are real, negative and different, the system in time is over-damped (or critically damped, if the roots are real, negative and equal). When the denominator roots have real and complex parts, it has a under-damped behavior in time, because it has a oscillator component included by the complex root part. When the roots have only complex parts, the behavior is undamped.

The three types of behavior are shown in Figure 2. Control systems engineers change these parameters also to vary the settling time (time to stabilize the signal in a constant value), peak value (maximum value) and others shapes parameters of the signal in control.

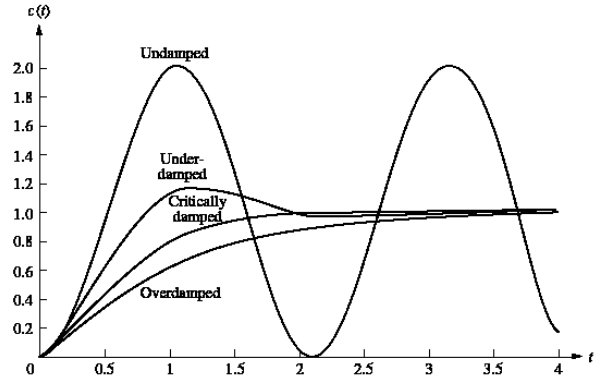


Figure 2. Over, under and undamped behaviors plots comparison.

III. Quantum Computing

A quantum bit, qubit, is a complex bidimensional unitary vector. The computational basis for a single qubit is $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$ vectors. A qubit $|\psi\rangle$ can be written as shown in Equation (1), where α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$. Quantum systems are qubit composition using tensor products: $|ij\rangle = |i\rangle \otimes |j\rangle$.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

A quantum system is altered by a unitary operator or a set of operators that change the qubit amplitude values. A quantum operator **U** over n qubits is a $2^n \times 2^n$ complex unitary matrix. Some main operators over one qubit are the identity operator **I**, not operator **X** and Hadamard **H** operator, described in Equation (2) and Equation (3). A quantum circuit is a combination of unitary operators applied to one or some set of qubits.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{matrix} \mathbf{I}|0\rangle = |0\rangle \\ \mathbf{I}|1\rangle = |1\rangle \end{matrix} \quad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{matrix} \mathbf{X}|0\rangle = |1\rangle \\ \mathbf{X}|1\rangle = |0\rangle \end{matrix} \tag{2}$$

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{matrix} \mathbf{H}|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle) \\ \mathbf{H}|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle) \end{matrix} \tag{3}$$

The identity operator I outputs the input; flip operator X behaves as the classical NOT on the computational basis; Hadamard transformation H generates superposition of states. The **CNOT** operator has 2 input qubits and 2 output qubits and flips the second qubit if the first one is 1 as show in Figure III.

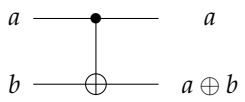


Figure 3. Circuit representation of **CNOT** operator.

A. qRAM - Quantum Neuron RAM-based Node

In [15] and [16], the quantum RAM based neuron was defined as quantization of the weightless neural networks proposed in [17].

The RAM node stores in its memory one bit addressed by an input bit string. The qRAM represents that bit storage by the gate A, as showed below. The gate A is the classical behavior of a memory addressed by one bit:

$$A = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} \quad \begin{matrix} \text{where} \\ A|00\rangle = |0\rangle I|0\rangle \\ A|10\rangle = |1\rangle X|0\rangle \end{matrix} \tag{4}$$

If the first input bit is zero, A matrix outputs $|0\rangle$ in the second position. Otherwise, A matrix outputs $|1\rangle$ in the second position. Then, the first bit called selector is changed to modify the content loaded. The second bit is always zero and is considered a output register because is only accessed in the final. A qRAM of input n-qubits has a collection of 2^n A's. Each operator A has its selector to change its content loaded. qRAM circuit representation is showed in Figure 4. To train a qRAM circuit one needs to change the selectors values to answer correctly with the input.

In general, given an input qubit, qRAM model chooses the selectors that will be applied in the output. If the chosen selector value is one, the output qubit will be one. Otherwise, the output qubit will be zero. If the input qubit is in superposition state, the selectors will be chosen in superposition, resulting in an output qubit in superposition.

To train a qRAM circuit one needs to change the selectors values to output correctly for an input in a pattern set [18].

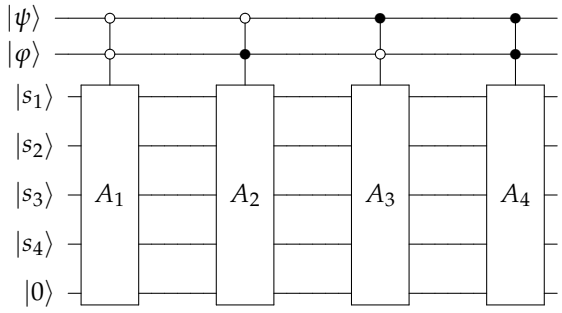


Figure 4. qRAM node with two inputs, four selectors and one output representation.

IV. Quantum Output Extraction Model

In [13], an extraction model of the output qubit during a qRAM neuron dynamics was proposed. The proposed method extracts the even and odds amplitudes of the final state during an iteration to recover the output qubit. The recovered output qubit can be fed back in any input register. The example below shows an output extraction of a qRAM node with one input (see Figure 5).

$$|i\rangle = \begin{pmatrix} a \\ b \end{pmatrix}, |s_0\rangle = \begin{pmatrix} c \\ d \end{pmatrix}, \quad |s_1\rangle = \begin{pmatrix} e \\ f \end{pmatrix}, |o\rangle = \begin{pmatrix} g \\ h \end{pmatrix} \tag{5}$$

where amplitudes are complex numbers. Making $|s\rangle = |s_0\rangle|s_1\rangle$, the $|\psi\rangle = |i\rangle|s\rangle|o\rangle$ as entry of the qRAM Node is:

$$\begin{aligned}|\psi\rangle = {} & aceg|0000\rangle + aceh|0001\rangle + acfg|0010\rangle + \\ & acfh|0011\rangle + adeg|0100\rangle + adeh|0101\rangle + adfg|0110\rangle + \\ & adfh|0111\rangle + bceg|1000\rangle + bceh|1001\rangle + bcfg|1010\rangle + \\ & bcfh|1011\rangle + bdeg|1100\rangle + bdeh|1101\rangle + bdfg|1110\rangle + \\ & bdfh|1111\rangle\end{aligned} \tag{6}$$

When the qRAM node N is applied in the $|\psi\rangle$, we have:

$$\begin{aligned}\mathsf{N}|\psi\rangle = |\psi'\rangle = {} & aceg|0000\rangle + aceh|0001\rangle + acfg|0010\rangle + \\ & acfh|0011\rangle + adeh|0100\rangle + adeg|0101\rangle + \\ & adfh|0110\rangle + adfg|0111\rangle + bceg|1000\rangle + \\ & bceh|1001\rangle + bcfh|1010\rangle + bcfg|1011\rangle + \\ & bdeg|1100\rangle + bdeh|1101\rangle + bdfh|1110\rangle + \\ & bdfg|1111\rangle\end{aligned} \tag{7}$$

The output qubit amplitudes can be extracted by:

$$\alpha^2 = \sum_{i\,even} |\psi'_i|^2 \tag{8}$$

$$\beta^2 = \sum_{i\,odd} |\psi'_i|^2 \tag{9}$$

since the output qubit is $|0\rangle$ in the even $|\psi'\rangle$ amplitude positions and is $|1\rangle$ in odd $|\psi'\rangle$ amplitude positions.

For the case described above, the output qubit recovered amplitudes α and β are:

$$\begin{aligned}|\alpha|^2_{t+1} = {} & |aceg|^2 + |acfg|^2 + |adeh|^2 + |adfh|^2 + \\ & |bceg|^2 + |bcfh|^2 + |bdeg|^2 + |bdfh|^2\end{aligned} \tag{10}$$

$$\begin{aligned}|\beta|^2_{t+1} = {} & |aceh|^2 + |acfh|^2 + |adeg|^2 + |adfg|^2 + \\ & |bceh|^2 + |bcfg|^2 + |bdeh|^2 + |bdfg|^2\end{aligned} \tag{11}$$

It is possible to recover the output qubit for feed back itself in the next iteration:

$$|o\rangle_{t+1} = \alpha_{t+1}|0\rangle + \beta_{t+1}|1\rangle \tag{12}$$

V. Proposed Model of Parameters Fitting

The extraction model dynamics finds an function of the input variables. As qRAM node is defined in [16] as always having output qubit set to zero, *i.e.* $g = 1$ and $h = 0$, and by definition that a quantum qubits have unit norm, we can find b, d,f, g, by its respective amplitude values, *i.e.* $b = \sqrt{-a^2+1}$, $d = \sqrt{-c^2+1}$, $f = \sqrt{-e^2+1}$, $g = \sqrt{-g^2+1}$. Then the function can be represented only by:

$$\alpha^2(a,c,e) = a^2c^2 - a^2e^2 + e^2 \tag{13}$$

To study the dynamics of this function with three variables and to understand the parameters variations behavior, we restrict the function to fewer parameters by setting some variable to constants. Considering the parameter configuration, resetting or making the selectors equal, the dynamics behavior takes over a transition of shapes that can be studied as general behavior categories.

In the studied dynamics, the output extracted via Equation 13 (EM) is fedback in the qRAM input register, as shown in Figure 5. That feedback process is called iteration and it is done many times. The values set for these iterations is called in studies of Dynamical System as orbit [19]. The orbit plotting shows the dynamics shape.

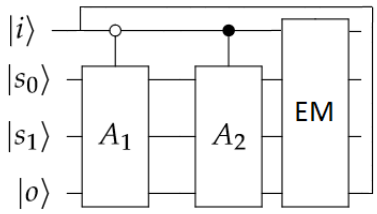


Figure 5. In extraction model dynamics, the output qubit extracted via EM block and it is feed back in input register many times (iterations), as explained in Section IV.

The dynamics types names are chosen to represent the dynamics behavior when the function parameter is increased. We talk about these five types below. We see the over-damped, under-damped and undamped function behaviors as is shown in [13]. The undamped behavior is also called "nondamped" to help in abbreviations. Here we talk of the function behavior in function of the parameters, showing the analytical function extracted from the reduced parameter space.

A. Type 1 - Fixed-Under-damped-Nondamped dynamics

In this type, the selector $|s_0\rangle$ is set to $|1\rangle$, *i.e.* $c = 0$. The iteration function is then:

$$f_1(x) = x_{t+1}^2 = e^2(1 - x_t^2) \tag{14}$$

Type-1 dynamics starts with 0 as fixed point, when $e = 0$. The orbits behavior becomes under-damped as e is increased. When e is maximum, *i.e.* $e = 1$, the dynamics is non-damped. The range of dynamics values in undamped behaviour depends of the initial value of x, x_0. Settling time and maximum value increase as x_0 increasing.

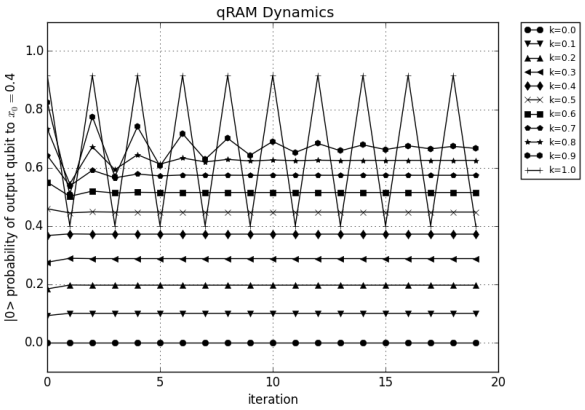


Figure 6. Type-1 dynamics orbits to different values of $e = k$ and when $x_0 = 0.4$. We can observe the orbits making an under-damped behavior as k is increased. The non-damped is reached when k is maximum, $k = 1$.

B. Type 2 - Fixed-Over-damped-Fixed dynamics

When the selector $|s_1\rangle$ is set as $|1\rangle$, *i.e.* $e = 0$, the function is:

$$f_2(x) = x_{t+1}^2 = c^2 x_t^2 \tag{15}$$

In this dynamics type, orbits are fixed in 0, when $c = 0$. Increasing c value, the orbits becomes over-damped, converging to zero. Settling time is increased as x_0 is increased. When c is maximum, *i.e.* $c = 1$, the dynamics orbits is a x_0 fixed point. Therefore, there is a superior and inferior horizontal limits, imposed by the x_0 and 0 respectively as c is increased.

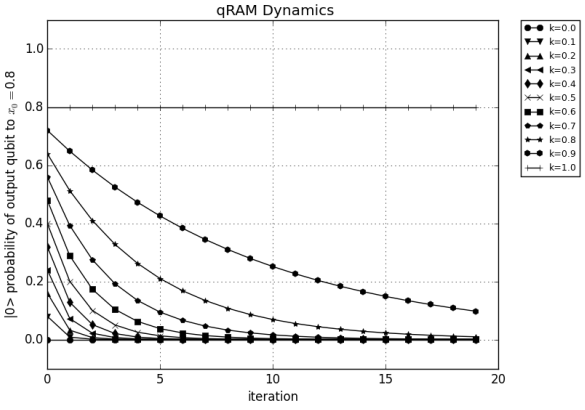


Figure 7. Type-2 orbits to different values of constant $c = k$ and $x_0 = 0.8$. We can observe the orbits reach horizontal axis in zero value when $k = 0$. The over-damped behavior is presented when $0 < k < 1.0$. In $k = 1.0$ configuration, the orbits are fixed to $x_0 = 0.8$.

C. Type 3 - Fixed dynamics

When the selectors are equal, *i.e.* $|s_0\rangle = |s_1\rangle = c|0\rangle + d|1\rangle = e|0\rangle + f|1\rangle$, the iteration function is:

$$f_3(x) = x_{t+1}^2 = c^2 = e^2 \tag{16}$$

This dynamics is always fixed depending of the value of the selectors.

D. Type 4 - Fixed-over-damped-Fixed dynamics

In this type, the selector $|s_0\rangle$ is set to zero, $|s_0\rangle = |0\rangle$, *i.e.* $c = 1$. The iteration function is:

$$f_4(x) = x_{t+1}^2 = x_t^2(1 - e^2) + e^2 \tag{17}$$

This dynamics orbit goes from x_0 as fixed point when $e = 0$, pass to over-damping orbits converging to one value as e is increased, and takes the one fixed point with $e = 1$. However, the over-damping here is the complement of the Type-2, *i.e.* the dynamics orbits is growing to one. Here also there is a superior and inferior horizontal limits, imposed by the 1 and x_0 respectively as c is increased. This type can be understood as the complement of the Type-2.

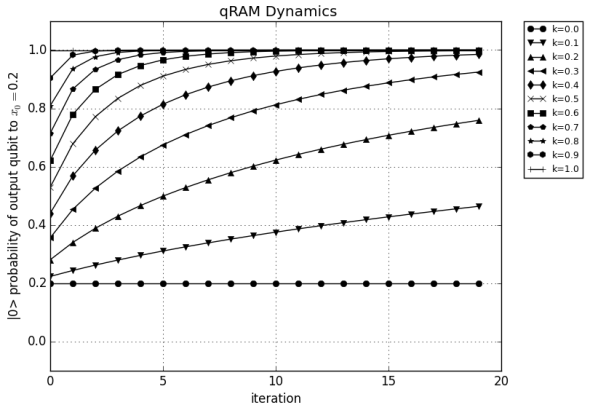


Figure 8. Type-4 orbits to different values of constant $e = k$ and $x_0 = 0.2$. In this figure, we can see to $k = 0$ the orbits are fixed to x_0. Increasing the k value, the orbits are an over-damping dynamics tending to one value. When k is maximum, the orbits dynamics is fixed to one.

E. Type 5 - Nondamped-Under-damped-Fixed dynamics

Finally, the selector $|s_1\rangle$ is set to one, $|s_1\rangle = |0\rangle$, *i.e.* $e = 1$. The iteration function is:

$$f_5(x) = x_{t+1}^2 = x_t^2(c^2 - 1) + 1 \tag{18}$$

This dynamics has non-damped behavior with $c = 0$. The dynamics transforms to an under-damped dynamics as c is increased and finally when $c = 1$ the dynamics is always 1 independently of x_0.

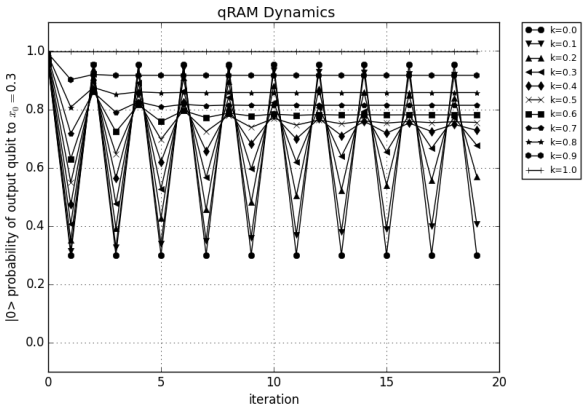


Figure 9. Type-5 orbits to different values of constant $c = k$ and $x_0 = 0.3$. To $k = 0$, the orbits are nondamped dynamics. To $0 < k < 1$ the dynamics shape is under-damped and when $k = 1$ the dynamics is fixed to one.

VI. Class Zones

Figure 10 shows the zones representation with the parameter variations. In the middle of the graph, a line divides over-damped and under-damped zones. In this line, the Type-3 is reached. The zones are labelled with the name according the behavior as parameter values are increased (from $|0\rangle$ to $|1\rangle$).

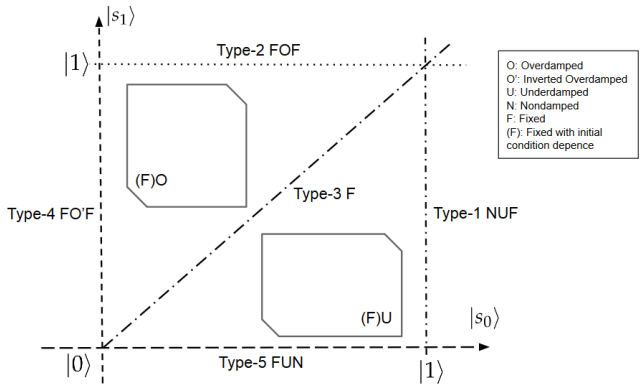


Figure 10. Representation of the parameter configuration zones. Each one classification type discussed in last section is positioned in the graph. In the middle of the graph, a line divides over-damped and under-damped zones. In this line, the Type-3 is reached. Each one zone is labelled according the values of parameter is increased (from $|0\rangle$ to $|1\rangle$). The description "FUN" in the horizontal axis is, for example, what happens when the $|s_1\rangle = |0\rangle$ and $|s_0\rangle$ goes from $|0\rangle$ (Fixed), between $0 < c < 1$ (Under-damped), to $|1\rangle$ (Non-damped).

In [12], a study of convergent activation dynamics in recurrent networks is proposed. Convergent dynamics are useful to accomplish many interesting tasks such as pattern recognition and classification, combinatorial optimization, conversion of printed documents to spoken words and so on. Convergent dynamics are also desired dynamics to retrieve information because it is

natural to want this information to be in the form of a single unchanging n-tuple of numbers. For oscillatory dynamics, given an input value, the periodic orbit saves some type of information. How can we retrieve the information embodied in the cycle? It can be the period of the cycle, the amplitude, or some function of Fourier coefficients of its components, or the average of some function over the cycle [12].

Following that, the expected configuration for a quantum weightless neurons to further increase the information retrieval capabilities lies in all of those which are convergent. In our investigation, this occurs in all space that is not $|s_0\rangle = |1\rangle$ (*i.e.* $c = 0$) and $|s_1\rangle = |0\rangle$ (*i.e.* $e = 1$). Type-1 and Type-5 depends of the c and e parameter setting to be a convergent or not dynamics. Although some dynamics converge to same point, their orbit can be explored to represent different information when an input configuration is iterated, implying in an orbit representation of an input configuration.

VII. Conclusion

In this paper, we showed a parametric classification of the qubit extraction dynamics model proposed in [13]. The understanding of the dynamics by its parameters implies in controlling its behavior and using its outputs in similar systems. Then, the qRAM neuron nodes dynamics can be used coupled with others systems governed by the node parameters.

The dynamics classification proposed divided the space of parameters in 5 classes of behavior and showed how its orbits change with the parameters. These dynamics classes explore the under-damping, over-damping and nondamped behavior, as cited in [13], but here their control is analytically more clear.

As shown in [12], the neuron convergent dynamics is desirable for information retrieval and we show the conditions where it occurs for the quantum neuron.

The use of qRAMs network dynamics is the next step of this work, extending the behavior analysis of only one node to more nodes. The output feedback in quantum registers as the selectors can also be investigated.

Acknowledgment

The authors would like to thank CNPq and FACEPE (Brazilian Research Agencies) for their financial support.

References

[1] L. A. Aguirre and C. Letellier. Modeling Nonlinear Dynamics and Chaos: A Review. *Mathematical Problems in Engineering*, 2009:35, 2009.

[2] S. Haykin and J. Principe. Making sense of a complex world [chaotic events modeling]. *Signal Processing Magazine*, 15:66–81, 1998.

[3] J. H. Poincaré and J. Kovalevsky. *Les Méthodes Nouvelles de la Mécanique Céleste*. Grands classiques Gauthier-Villars. Blanchard, Paris, 1892.

[4] E. N. Lorenz. Deterministic nonperiodic flow. *Journal of the Atmospheric Sciences*, 20:130–141, 1963.

[5] S. H. Strogatz. *Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity)*. Westview Press, 1994.

[6] L. K. Grover. Quantum Mechanics Helps in Searching for a Needle in a Haystack. *Phys. Rev. Lett.*, 79(2):325–328, 1997.

[7] P. Benioff. Foundational Aspects of Quantum Computers and Quantum Robots. *Superlattices and Microstructures*, 23:407–417, 1998.

[8] P. Benioff. Quantum robots and environments. *Phys. Rev. A*, 58:893–904, Aug 1998.

[9] P. Cabauy and P. Benioff. Cyclic networks of quantum gates. *Phys. Rev. A*, 68:032315, Sep 2003.

[10] F. M. de Paula, A. Silva, T. Ludermir, and W. R. De Oliveira. Analysis of Quantum Neural Models. *XI Brazilian Congress of Computational Intelligence*, 1:1–6, 2013.

[11] D. Gross. The importance of chaos theory in the development of artificial neural systems. http://www.lycaeum.org/ sputnik/Misc/chaos.html.

[12] M. W. Hirsch. Convergent activation dynamics in continuous time networks. *Neural Networks*, 2(5):331 – 349, 1989.

[13] F. M. de Paula, W. R. de Oliveira, A. J. da Silva, and T. B. Ludermir. Chaos in Quantum Weightless Neuron Node Dynamics. *Neurocomputing*, 2015.

[14] N. S. Nise. *Control Systems Engineering*. John Wiley and Sons, 2011.

[15] W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, A. Leonel, W. R. Galindo, and J. C. Pereira. Quantum Logical Neural Networks. In *Brazilian Symposium on Neural Networks*, pages 147–152, 2008.

[16] W. R. de Oliveira. Quantum RAM Based Neural Networks. In *ESANN*, pages 22–24, 2009.

[17] I. Aleksander. Self-adaptive universal logic circuits. *Electronics Letters 2*, 8:321–322, 1966.

[18] A. J. da Silva, W. R. de Oliveira, and T. B. Ludermir. Classical and superposed learning for quantum weightless neural networks. *Neurocomputing*, 75(1):52–60, 2012.

[19] R. L. Devaney. *A first course in chaotic dynamical systems: theory and experiment*. Westview Press, 1992.

4

Solving NP-complete Problems using Quantum Weightless Neuron Nodes

In this chapter, we show how a non-unitary operator together with a quantum weightless neuron can solve NP-Complete Problems in polynomial time. This work (PAULA NETO et al., 2015b) was published in the *Brazilian Congress on Intelligent Systems* (BRACIS) 2015. Although the paper result depends on whether that non unitary operator exists or not, this implies the wide computational complexity of the nonunitary gates. The quantum algorithm proposed can be used in practical problems where there is unknown solution in a classical computing. The qRAM neurons are attached to non-unitary operators to solve the 3-SAT Problem representing each one a clause in a logical function (Section V).

Solving NP-complete Problems using Quantum Weightless Neuron Nodes

Fernando M. de Paula Neto, Teresa B. Ludermir
Centro de Informática
Universidade Federal de Pernambuco
Recife, Pernambuco
Email: {fmpn2, tbl}@cin.ufpe.br

Wilson R. de Oliveira, Adenilton J. da Silva
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco
Recife, Pernambuco
Email: wilson.rosa@gmail.com, ajs@deinfo.ufrpe.br

***Abstract*—Despite neural networks have super-Turing computing power, there is no known algorithm for obtaining a classical neural networks that solves NP-complete problems in polynomial time. However this paper shows that a quantum neural networks model coupled with a non-unitary operator can solve 3-SAT in polynomial time. The proposed method uses a network circuit to represent a Boolean logic function and a non-unitary operator to decide the satisfiability. The parameters of the network is set deterministically and manually, accordingly to the problem at hand with neither quantum nor classical learning.**

***Keywords*-Quantum computing; Weightless neuron; NP-complete problem; Non unitary;**

I. Introduction

Classical computers apparently have limitations to efficiently solve complex problems such as those in the NP-complete class. The travelling salesman, the graph coloring, the knapsack and the satisfiability (SAT) problems, are examples that are in NP-complete class and are so hard that solve one them in polynomial time implies we can also solve all problems in NP [18]. These are technically called NP-complete problems.

Neural networks have been used to solve problems inductively, using a training set to adjust their parameters. In the supervised training there are expected targets in each training example so the network has its parameters adjusted for minimising the measured error in relation of the expected value. That inductive learning gives an approximate solution not necessarily the best one. Several neural network models were proposed to solve NP-complete problems but none always gives a precise correct solution due to its stochastic behaviour and limitation of heuristic method of updating the network parameters [11][10].

Quantum neural networks were firstly proposed in the nineties and several works propose concrete models of quantum neural networks [3][6]. Some advantages of quantum neural networks are the capacity of a single neuron to solve non-linearly separable patterns, learning algorithms with polynomial cost in relation to the number of patterns in training set and an exponential gain in memory capacity [7].

Non-unitary and non-linear operators [1][19][20] combined with quantum gates solve analytically NP-complete problems in polynomial time [13][15]. In this paper, we show how to solve 3-SAT problem using a quantum weightless neuron circuit [6] and a non-unitary gate [12].

This paper is organized as follows. In Section II and Section III we discuss quantum computing and quantum weightless neuron nodes, respectively. 3-SAT Problem is defined in Section IV and the 3-SAT proposed solution using qRAM circuit representation for Boolean logic functions is presented in Section V, followed by an example. The conclusions are presented in Section VI.

II. Quantum Computing

A quantum bit, *qubit*, is a special complex bidimensional unitary vector. The computational basis for the vector space, $\mathbb{C}^2$, of all qubits is composed of the vectors written in the Dirac or bra-ket nottion: $|0\rangle = [1,0]^T$ and $|1\rangle = [0,1]^T$. An arbitrary qubit $|\psi\rangle$ can be written as linear combination (*superposition*) of the computational basis as shown in Equation (1), where α and β are complex numbers and subject to the restriction that $|\alpha|^2 + |\beta|^2 = 1$. Qubits represents the states of a single quantum system. Composition of quantum systems are obtained with tensor products: $|ij\rangle = |i\rangle \otimes |j\rangle$.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

Quantum system evolves by a unitary operators that change the state amplitude values. A quantum operator $\mathbf{U}$ over n qubits is a $2^n \times 2^n$ complex unitary matrix. Some main operators over one qubit are the identity operator $\mathbf{I}$, the flip operator $\mathbf{X}$ and Hadamard $\mathbf{H}$ operator, described in Equation (2) and Equation (3). A quantum circuit is a combination of unitary operators applied to one or some set of qubits.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{matrix} \mathbf{I}|0\rangle = |0\rangle \\ \mathbf{I}|1\rangle = |1\rangle \end{matrix} \qquad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{matrix} \mathbf{X}|0\rangle = |1\rangle \\ \mathbf{X}|1\rangle = |0\rangle \end{matrix} \tag{2}$$

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \begin{matrix} \mathbf{H}|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle) \\ \mathbf{H}|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle) \end{matrix} \tag{3}$$

The identity operator **I** outputs the input; the flip operator **X** behaves as the classical **NOT** on the computational basis; Hadamard transformation **H** generates an even superposition of states. The **CNOT** operator has two input qubits and two output qubits and flips the second qubit if the first one is 1 as show in Figure 1.

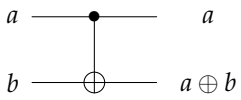


Figure 1. The **CNOT** operator

III. Quantum Neurons

Many quantum neurons have been proposed generalising the (classical) artificial neural networks [7][16]. Some of them are only quantum inspired but their working is not intrinsically quantum once their learning algorithms interfere with network parameters violating some quantum postulates [4]. The qRAM neuron node [6][5] studied here is a generalisation of RAM based nodes, used in many applications of pattern recognition and classification tasks [14]. The RAM based nodes have good generalization capabilities [9] and computational power [8].

A. qRAM - Quantum Neuron RAM-based Node

In [6] and [5], the quantum RAM based neuron was defined as quantization of the weightless neural networks proposed in [2].

The RAM node stores in its memory one bit addressed by an input bit string. The qRAM represents that bit storage by the gate A, as showed below. The gate A is the classical behavior of a memory addressed by one bit:

$$A = \begin{pmatrix} I & 0 \\ 0 & X \end{pmatrix} \quad \begin{matrix} \text{where} \\ A|00\rangle = |0\rangle I|0\rangle \\ A|10\rangle = |1\rangle X|0\rangle \end{matrix} \tag{4}$$

If the first input bit is zero, A matrix outputs $|0\rangle$ in the second position. Otherwise, A matrix outputs $|1\rangle$ in the second position. Then, the first bit called selector is changed to modify the content loaded. The second bit is always zero and is considered an output register because is only accessed in the final. A qRAM of input n-qubits has a collection of 2^n A's. Each operator A has its selector to change its content loaded. qRAM circuit representation is shown in Figure 2. To train a qRAM circuit one needs to change the selectors values to answer correctly with the input.

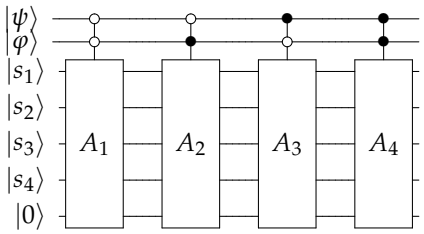


Figure 2. qRAM node with two inputs, four selectors and one output representation.

In general, given an input qubit, the qRAM model chooses the selectors that will be applied to the output. If the chosen selector value is one, the output qubit will be one. Otherwise, the output qubit will be zero. If the input qubit is in superposition state, the selectors will be chosen in superposition, resulting in an output qubit in superposition.

One can represent the qRAM with the implicit selector representation as shown in Figure 3. This notation is introduced in [6] as q-ROM, quantum read-only memory, since the addressed content cannot be modified and the quantum operators U that implicitly represent the selectors are fixed. Training a q-ROM, shown in Figure 3, is to choose the operator U_i for each i positions.

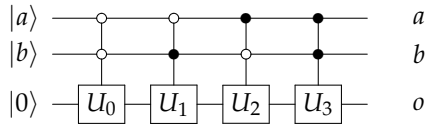


Figure 3. q-PLN node, where U_i, $i = 0, 1, ..., 2^n - 1$, can be I or X gates.

IV. 3-SAT Problem

The Boolean Satisfiability Problem (SAT) is, given a propositional logic expression $\phi(x_1, x_2..., x_n)$, to decide if there are logical values for each x_i that turn the expression ϕ true.

The 3-SAT problem restricts the SAT logic expression format. A literal is a Boolean variable or a negated Boolean variable. A clause is a disjunction of literals and a 3-clause is a disjunction of exactly 3 literals. A 3-SAT expression is formed by conjunctions of 3-clauses. Since each clause contains exactly three literals, the

expression will be at most $(2n)^3 = 8n^3$ clauses, where n is the number of variables.

A problem p is NP-complete if satisfies two conditions (a) it is in NP, and (b) every problem in NP is polynomial time reducible to p. These problems equally difficult to solve and have to date only be solved by algorithms with an exponential number of steps as function of the input size. In this class of problems, there is no known polynomial time solution. 3-SAT problem is NP-complete [17].

V. Solving 3-SAT using qRAM

In this Section, we use the quantum node qRAM in the solution of the 3-SAT using a non-unitary operator. The qRAM nodes are used to represent the logic expressions. The non-unitary operator applied to the qRAM circuit provides a null vector as output only if the logic expression is not satisfiable and provides a superposition vector in case the expression is satisfiable.

A. Logic expression as qRAM circuits

To set a logic expression in a qRAM circuit, we configure the selectors $|\mathbf{s}\rangle$ of each qRAM node to represent a logic operation of each clause. As example, for a logic expression of two variables a e b formed by the disjunction $a \vee b$, the selectors $|s_1\rangle$, $|s_2\rangle$ and $|s_3\rangle$ are set to $|1\rangle$ and the selector $|s_0\rangle$ is set to $|0\rangle$. The disjunction operation qRAM representation is presented in Figure 4. For a conjunction representation, the selector $|s_3\rangle$ is set to $|1\rangle$ and the others selectors are set to $|0\rangle$. That procedure is extended for qRAMs with more than two inputs.

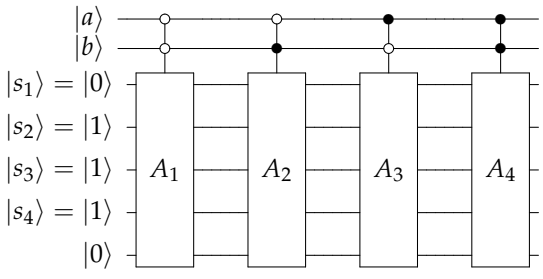


Figure 4. qRAM representing disjunction operation of two inputs $|a\rangle$ and $|b\rangle$.

A logic expression is formed by a circuit with qRAMs where each one represents a clause. Quantum ancillary registers load partial results for each operation. The last qRAM node calculates the conjunction of the partial outputs giving the final output. In a conjunctive normal form which has c clauses, it is necessary $c + 1$ output registers $|o\rangle$.

In a general construction, a clause with n variables, i.e. a logic expression which executes a disjunction operator in the n inputs, the qRAM selector $|s_0\rangle$ is set to $|0\rangle$, and all the others ones $|s_1, s_2..., s_{2^n}\rangle$ are set to $|1, ..., 1\rangle$. In the case the operation is a conjunction, only the selector s_{2^n} is set to $|1\rangle$, and the others are set to $|0\rangle$.

Some Boolean expressions have negated variables $|\hat{x}\rangle$. For representing these variables, it is proposed to use one additional input in the qRAM. The quantum circuit designer is responsible to feed in that input the negated representation of the respective variable $|x\rangle$. This is done simply by the negation of the variable by the NOT gate X. In this way, expressions of n free variables, qRAM circuits will have $2n$ inputs.

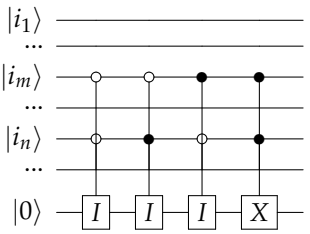


Figure 5. Example of qROM prepared to a conjunctive operation with no selectors representation. The gates I e X do the configuration role. This example considers the m and n-th register as input.

To process a clause in a qRAM circuit, a qRAM node needs to choose what variables will be used as input in the operation. It is possible the neuron does not need use them alls. A qRAM $\frac{OR}{S|k}$ applies its disjunction operation in the set of positions $S \subseteq \{t|1 < t < p\}$, where n is the quantity of input variables, m is the clauses quantity, and $p = 2^n + m + 1$ and the output is load in the k-th output register. Figure 5 represents a qROM setting for a conjunction operation and Figure 6 show the same circuit but in this representation the internal configuration is hidden for better displaying the image.

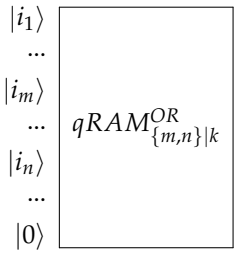


Figure 6. Representation of qRAM node. The selectors and U matrices are hidden in the circuit representation to simplify the notation. This example considers the m and n-th registers of input and put the result of the OR operation in the k-th register

In Figure 7, it is shown an example of qRAM circuit of the Boolean logic that express $\phi(x_1, x_2) = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2) \wedge (x_1 \vee \bar{x_2}) \wedge (\bar{x_1} \vee \bar{x_2})$ using quantum ROMs. For simplify our notation, we use qRAMs notation even in the qROMs configurations, i.e. when the selectors are hidden.

1) Solving 3-SAT: Then, to solve a 3-SAT problem, a qRAM circuit is created to represent a Boolean expression. To calculate all possibles values in a single shot operation, the inputs are prepared in superposition. The output registers are set to zero. The neuron circuit calculates the logic expression and the non-unitary operator verifies if any input configuration turns the expression true [12].

As some inputs of the circuits are in negated representation of some variables, we cannot put all inputs in the superposition as $|x, \hat{x}\rangle$ would be wrongly set to $\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$. In the case which negated inputs are used, a Bell state for this couple qubit is created, as described in Figure 8: $|x, \hat{x}\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$.

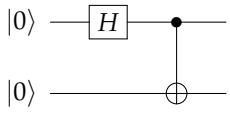


Figure 8. To solve 3-SAT problems, the qRAM circuit receives all inputs in superposition. In the case in which variables and their negated representation are input, a Bell state is set to input these couple of qubits, since the possible values are $|01\rangle$ e $|10\rangle$ and not all values.

After applying the inputs in the qRAM circuit, the next step is apply the non-unitary operator [12] in the last output qubit:

$$O = 2^n \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{5}$$

That operator does a check if some final output qubit was flipped to $|1\rangle$, i.e. if some input value turns the logic expression true. To a circuit with n variables and k clauses, we have $2n$ registers of input, $k+1$ output registers, then $m = 2n + k + 1$ registers in total. To apply the non-unitary operator described in Equation 5 only in the last total output register, the operator above is used:

$$O^{(m)} = (\otimes^{m-1} I_1) \otimes O \tag{6}$$

where I is identity gate. Finally, the operator O plays the role for checking the output. This is done by applying the operator O and verifying if the result is a null vector or not. Since $O|0\rangle = 2^n|1\rangle\langle 1||0\rangle = \mathbf{0}$, then if our final quantum vector is null, no input turns the final output register to $|1\rangle$, i.e. the logic expression is not satisfiable.

Otherwise, if there is one solution, the final output qubit is $|1\rangle$, even in superposition state, our final quantum vector is not a null vector. For example, given a superposition state, applying the operator O results in $O(\alpha_0|0\rangle + \beta_0|1\rangle) = \alpha_0 2^n|1\rangle\langle 1||0\rangle + \alpha_1 2^n|1\rangle\langle 1||1\rangle = \mathbf{0} + \alpha_1 2^n|1\rangle = \alpha_1 2^n|1\rangle$ [12].

As an example, given an expression

$$\begin{aligned} &\phi(x_1, x_2, x_3, x_4) = (\hat{x_1} \vee x_2 \vee x_4) \wedge (\hat{x_2} \vee x_3 \vee x_4) \wedge \\ &(x_1 \vee \hat{x_3} \vee x_4) \wedge (x_1 \vee \hat{x_2} \vee \hat{x_4}) \wedge (x_2 \vee \hat{x_3} \vee \hat{x_4}) \wedge \\ &(\hat{x_1} \vee x_3 \vee \hat{x_4}) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\hat{x_1} \vee \hat{x_2} \vee \hat{x_3}) \end{aligned} \tag{7}$$

We need six qRAMs, five ones are used to do a disjunction in each clause and one qRAM does a conjunctive composition of the partial outputs.

Analytically, the input in superposition to apply in the qRAM circuit:

$$\begin{aligned} |\psi\rangle &= |x_1, \hat{x_1}, x_2, \hat{x_2}, x_3, \hat{x_3}, x_4, \hat{x_4}, o_1, o_2, o_3, o_4, o_5, o_6\rangle \\ &= (\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \otimes \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \otimes \\ &\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \otimes \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)) \otimes (|000000\rangle) \end{aligned} \tag{8}$$

Distributively, the input vector is also equal to:

$$\begin{aligned} |\psi\rangle &= |x_1, \hat{x_1}, x_2, \hat{x_2}, x_3, \hat{x_3}, x_4, \hat{x_4}, o_1, o_2, o_3, o_4, o_5, o_6\rangle \\ &= |01010101\rangle|000000\rangle + |01010110\rangle|000000\rangle + \\ &|01011001\rangle|000000\rangle + |01011010\rangle|000000\rangle + \\ &|01100101\rangle|000000\rangle + |01100110\rangle|000000\rangle + \\ &|01101001\rangle|000000\rangle + |01101010\rangle|000000\rangle + \\ &|10010101\rangle|000000\rangle + |10010110\rangle|000000\rangle + \\ &|10011001\rangle|000000\rangle + |10011010\rangle|000000\rangle + \\ &|10100101\rangle|000000\rangle + |10100110\rangle|000000\rangle + \\ &|10101001\rangle|000000\rangle + |10101010\rangle|000000\rangle \end{aligned} \tag{9}$$

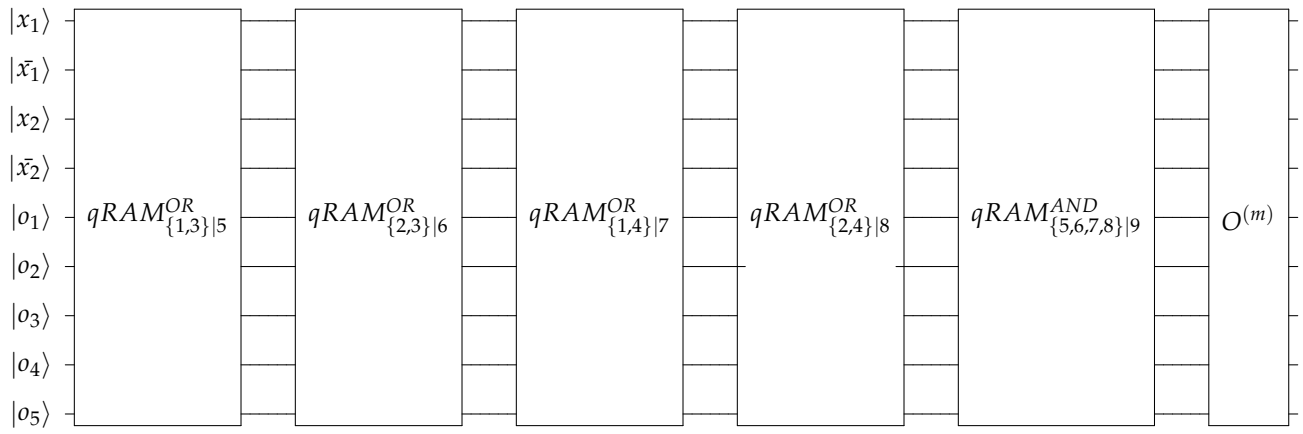


Figure 7. qRAM Circuit representing the logic function $\phi(x_1, x_2) = (x_1 \vee x_2) \wedge (\bar{x_1} \vee x_2) \wedge (x_1 \vee \bar{x_2}) \wedge (\bar{x_1} \vee \bar{x_2})$. Each qRAM node $qRAM_{S|k}$ has its indexes input set and define the k-th output to load its result. The register o_i loads the result of the i-th clause and the register o_5 loads the final output. The non-unitary operator $O^{(m)}$ verifies if the input values satisfies the equation. If the final state is a null vector the input vector does not satisfy the equation. Otherwise, it satisfies.

When the first qRAM processes the first clause, $(\hat{x_1} \vee x_2 \vee x_4)$ the first register $|o_1\rangle$ is updated and the input vector is:

$$
\begin{aligned}
&|x_1, \hat{x_1}, x_2, \hat{x_2}, x_3, \hat{x_3}, x_4, \hat{x_4}, o_1, o_2, o_3, o_4, o_5, o_6\rangle = \\
&|01010101\rangle|100000000\rangle + |01010110\rangle|100000000\rangle + \\
&|01011001\rangle|100000000\rangle + |01011010\rangle|100000000\rangle + \\
&|01100101\rangle|100000000\rangle + |01100110\rangle|100000000\rangle + \\
&|01101001\rangle|100000000\rangle + |01101010\rangle|100000000\rangle + \\
&|10010101\rangle|100000000\rangle + |10010110\rangle|100000000\rangle + \\
&|10011001\rangle|100000000\rangle + |10011010\rangle|100000000\rangle + \\
&|10100101\rangle|000000000\rangle + |10100110\rangle|000000000\rangle + \\
&|10101001\rangle|000000000\rangle + |10101010\rangle|000000000\rangle
\end{aligned} \tag{10}
$$

When the second qRAM processes the second clause, $(\hat{x_2} \vee x_3 \vee x_4)$, the second output register $|o_2\rangle$ is updated and the input vector is:

$$
\begin{aligned}
&|x_1, \hat{x_1}, x_2, \hat{x_2}, x_3, \hat{x_3}, x_4, \hat{x_4}, o_1, o_2, o_3, o_4, o_5, o_6\rangle = \\
&|01010101\rangle|110000000\rangle + |01010110\rangle|110000000\rangle + \\
&|01011001\rangle|110000000\rangle + |01011010\rangle|110000000\rangle + \\
&|01100101\rangle|110000000\rangle + |01100110\rangle|110000000\rangle + \\
&|01101001\rangle|110000000\rangle + |01101010\rangle|110000000\rangle + \\
&|10010101\rangle|110000000\rangle + |10010110\rangle|110000000\rangle + \\
&|10011001\rangle|110000000\rangle + |10011010\rangle|110000000\rangle + \\
&|10100101\rangle|010000000\rangle + |10100110\rangle|010000000\rangle + \\
&|10101001\rangle|010000000\rangle + |10101010\rangle|010000000\rangle
\end{aligned} \tag{11}
$$

When the third qRAM processes the third clause, $(x_1 \vee \hat{x_3} \vee x_4)$, the third register $|o_3\rangle$ is updated and the input vector is:

$$
\begin{aligned}
&|x_1, \hat{x_1}, x_2, \hat{x_2}, x_3, \hat{x_3}, x_4, \hat{x_4}, o_1, o_2, o_3, o_4, o_5, o_6\rangle = \\
&|01010101\rangle|111000000\rangle + |01010110\rangle|111000000\rangle + \\
&|01011001\rangle|111000000\rangle + |01011010\rangle|111000000\rangle + \\
&|01100101\rangle|110000000\rangle + |01100110\rangle|110000000\rangle + \\
&|01101001\rangle|110000000\rangle + |01101010\rangle|110000000\rangle + \\
&|10010101\rangle|111000000\rangle + |10010110\rangle|111000000\rangle + \\
&|10011001\rangle|111000000\rangle + |10011010\rangle|111000000\rangle + \\
&|10100101\rangle|011000000\rangle + |10100110\rangle|011000000\rangle + \\
&|10101001\rangle|011000000\rangle + |10101010\rangle|011000000\rangle
\end{aligned} \tag{12}
$$

Finally, when the sixth first qRAMs process the disjunction of the clauses, the input vector is:

$$
\begin{aligned}
&|x_1, \hat{x_1}, x_2, \hat{x_2}, x_3, \hat{x_3}, x_4, \hat{x_4}, o_1, o_2, o_3, o_4, o_5, o_6\rangle = \\
&|\psi\rangle = \\
&|01010101\rangle|111011110\rangle + |01010110\rangle|111011110\rangle + \\
&|01011001\rangle|111011110\rangle + |01011010\rangle|111011110\rangle + \\
&|01100101\rangle|110111110\rangle + |01100110\rangle|110111110\rangle + \\
&|01101001\rangle|110111110\rangle + |01101010\rangle|110111110\rangle + \\
&|10010101\rangle|111110110\rangle + |10010110\rangle|111110110\rangle + \\
&|10011001\rangle|111110110\rangle + |10011010\rangle|111110110\rangle + \\
&|10100101\rangle|011111110\rangle + |10100110\rangle|011111110\rangle + \\
&|10101001\rangle|011111110\rangle + |10101010\rangle|011111110\rangle
\end{aligned} \tag{13}
$$

As can be seen, in each state there is at least one register set to zero in some output register $|o_1, ..., o_8\rangle$. Then, the last qRAM in the circuit will not alter the final output register $|o_9\rangle$ by the conjunction operation.

Applying Equation 5 in the output quantum state, the state $|\psi\rangle$ transforms to $|\psi\rangle'$:

$$O^{(m)} = (\otimes^{m-1} I_1) \otimes O \tag{14}$$

$$|\psi\rangle' = O^{(m)}|\psi\rangle \tag{15}$$

Verifying the value of $|\psi\rangle'$, we check that is a null vector $\mathbf{0}$, as expected, since the expression is not satisfiable: $|\psi'\rangle = \mathbf{0}$.

VI. Conclusion

In this paper, we demonstrate that it is possible to solve the 3-SAT problem using a non-unitary operator quantum gate and a quantum weightless neural network in polynomial time. Each logic function clause is represented by a qRAM node, setting the selectors according its operation. The non-unitary operator verifies weather the function is satisfiable or not. No one knows if there is a configuration of a classical weightless neuron networks to solve 3-SAT, this would implies that $P = NP$, but using a quantization of weightless neuron node, 3-SAT problem is solved in polynomial time. There are controversies about physical implementation of non-unitary operators, notwithstanding there are some papers showing its working [19] [20].

The logic function representation can be studied further improved as regarding the circuit size, although it grows polynomially in the input size. Further studies in quantum approximated heuristic solutions for NP-complete problems are envisioned [20].

Acknowledgment

The authors would like to thank CNPq and FACEPE (Brazilian Research Agencies) for their financial support.

References

[1] D. S. Abrams and S. Lloyd. Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems. *Phys. Rev. Lett.*, 81(18):3992–3995, 1998.

[2] I. Aleksander. Self-adaptive universal logic circuits. *Electronics Letters 2*, 8:321–322, 1966.

[3] M. V. Altaisky. Quantum neural network. Technical report, Joint Institute for Nuclear Research, Russia, 2001.

[4] A. J. da Silva, W. R. de Oliveira, and T. B. Ludermir. Comments on "quantum m-p neural network". *International Journal of Theoretical Physics*, 54(6):1878–1881, 2015.

[5] W. R. de Oliveira. Quantum RAM Based Neural Networks. In *ESANN*, pages 22–24, 2009.

[6] W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, A. Leonel, W. R. Galindo, and J. C. Pereira. Quantum Logical Neural Networks. In *Brazilian Symposium on Neural Networks*, pages 147–152, 2008.

[7] F. M. de Paula, A. J. da Silva, T. B. Ludermir, and W. R. de Oliveira. Analysis of Quantum Neural Models. *XI Brazilian Congress of Computational Intelligence*, 1:1–6, 2013.

[8] M. C. P. de Souto, T. B. Ludermir, and W. R. de Oliveira. Equivalence between ram-based neural networks and probabilistic automata. *Neural Networks, IEEE Transactions on*, 16(4):996–999, 2005.

[9] A. F. De Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, L. Veronese, and C. Badue. Automated multi-label text categorization with vg-ram weightless neural networks. *Neurocomput.*, 72(10-12):2209–2217, June 2009.

[10] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biological cybernetics*, 52(3):141–152, 1985.

[11] J.-S. Lai, S.-Y. Kuo, and I.-Y. Chen. Neural networks for optimization problems in graph theory. In *Circuits and Systems, 1994. ISCAS '94., 1994 IEEE International Symposium on*, volume 6, pages 269–272 vol.6, May 1994.

[12] A. Leporati and S. Felloni. Three "quantum" algorithms to solve 3-sat. *Theoretical Computer Science*, 372(2–3):218 – 241, 2007.

[13] A. Leporati, C. Zandron, and G. Mauri. Simulating the fredkin gate with energy-based p systems. *J. UCS*, 10(5):600–619, 2004.

[14] T. B. Ludermir, A. de Carvalho, A. P. Braga, and M. C. P. de Souto. Weightless neural models: a review of current and past works. *Neural Computing Surveys*, 2:41–61, 1989.

[15] M. Ohya and I. Volovich. Quantum Computing, NP-complete Problems and Chaotic Dynamics. *eprint arXiv:quant-ph/9912100*, Dec. 1999.

[16] M. Schuld, I. Sinayskiy, and F. Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13(11):2567–2586, 2014.

[17] M. Sipser. *Introduction to the Theory of Computation*. 2006.

[18] N. Srinivasan. Tutorial: Current state of p vs np problem. In *Recent Trends in Information Technology (ICRTIT), 2011 International Conference on*, pages 1–1, June 2011.

[19] H. Terashima and M. Ueda. Nonunitary quantum circuit. *eprint arXiv:quant-ph/0304061*, Apr. 2003.

[20] C. Williams and R. Gingrich. Non-unitary probabilistic quantum computing circuit and method, Aug. 4 2005. US Patent App. 11/007,792.

5

Quantum Neuron Chaotic Filtering Dynamics

In this chapter, we propose a new quantum dynamics information extraction method. That method was inspired in the quadratic operator S proposed by BECHMANN-PASQUINUCCI; HUTTNER; GISIN (1998). For this reason we name it the **Quadratic Extraction Model**. Trough that method, it is possible to visualize quadratic functions which consider complex numbers during its dynamics. We use this method to set up a chaotic configuration of the qRAM dynamics and to observe bifurcation and chaos as the parameters are changed. The presence of chaos in orbit diagrams is observed.

We show how the **Quadratic Extraction Model** works (Section 5) and we present the qRAM dynamics orbit diagrams for that model which have bifurcation and chaos when its parameter is varied (Section 6). The fixed points movement is presented during parameters fitting and any learning algorithms can be use that information to make the neuron learns or adapts itself. The contribution of this work in mainly the complex values iteration for qRAM dynamics where present chaos and bifurcation in orbit diagrams. In Section 6.1 we show the Quadratic Extraction Model (QEM) circuit can represent logic operations training its parameters from a simple delta rule.

An extended version of this chapter is being prepared to be sent to a computing journal.

Quantum Neuron Chaotic Filtering Dynamics

Fernando M. de Paula Neto[a], Wilson R. de Oliveira[b], Teresa B. Ludermir[a], Adenilton J. da Silva[b]

[a] *Centro de Informática*
Universidade Federal de Pernambuco
[b] *Departamento de Estatística e Informática*
Universidade Federal Rural de Pernambuco

Abstract

Natural neuron dynamics activities have shown that phase transition and chaos provide optimal behaviour for information processing. In artificial neural models, that behaviour is expected also to maximize the neuron learnability. In this work, we propose a new quantum information extraction model during the quantum node RAM dynamics where complex values are iterated and chaos is shown in its orbit diagram. In that way, we show experimentally that it is possible to view bifurcation and chaos varying their parameters.

Keywords: Quantum weightless neurons, Weightless neurons, Dynamical systems, Chaos, Dynamics parameterization

1. Introduction

Biological neurons receive the attention of dynamical systems researchers because the brain is a self-feedback system. The information about the parameters, bifurcations and phase change of neurons helps scientists to understand how information processing can be maximized in the brain activities [1, 2, 3, 4]. In the artificial world, there is further evidence pointing agreements between nonlinearity and chaotic environments and artificial neuron networks learnability [5, 6, 3]. It is argued that neuron populations are predisposed to instability and bifurcation that depend on external input and internal parameters [7, 8].

To find the analytical expression of a particular system does not necessarily imply to understand its parametric dependency, its possible temporal convergence, chaos and fractality existence, bifurcation, and other dynamics

behaviour [9]. The natural (chemical, biological, physics) or artificial (artificial neural networks, computing networks, data centers) systems dynamics need to be understood in quantitative and qualitative behaviours to be used as useful subsystems. A dynamics that is understood can be induced to produce an expected behavior, e.g. either serving as input signal generator or working as specific task module in a complex system. Encrypt texts may involve a chaotic module [10]. Fractals can help either to generate or to segment textures [11, 12]. Biological neurons can learn fast in chaotic regime, bifurcation or phase change [13, 14, 5].

Existing quantum neurons are inspired by quantum computing or are quantum models [15, 16, 17, 18]. Quantum neurons have been used to solve pattern recognition problems and machine learning tasks [19, 17, 20, 21].

The fact that biological neurons can learn more under certain dynamics conditions leads us to investigate a kind of neuron model and its behaviour in time. In this work, we analyze the dynamics of a quantum artificial weightless neuron model named qRAM in [22]. The qRAM parameters need to be adjusted and influence their internal characteristics. The parameter choice can induce certain set of behaviors in qRAMİn this work we define a extraction model that search for a chaotic qRAM parametric setting. There is evidence that in a chaotic configuration a neural model will work at maximum efficiency for specific tasks of machine learning, pattern recognition and artificial intelligence in general [1, 2, 3, 4, 7, 8].

In [23] we show that neuron operation will entangle the qubits representing the neuron input, output and parameters. The neuron entangled configuration does not allow to view the qRAM output result separated from the other qubits. The output can be estimated by the Real Extraction Model (**REM**) proposed in [23]. The estimated output is feedback in the input. The **REM** estimates qubits amplitudes norms and it does not enable complex values manipulation. Other works investigate chaos in quantum systems but the states are in predetermined format to increase their entanglement [24, 25, 26]. The **REM** experiments shown that real values iteration is not enough to generate chaos in orbit diagrams of this qRAM system.

Based on **REM**, this paper presents a new quantum qubit extraction model to work with complex values named Quadratic Extraction Model (**QEM**). We use **QEM** to find a qRAM configuration that generates chaos and bifurcations in neurons dynamics. The advantage of **QEM** over **REM** is the capacity to generate a qRAM chaotic configuration that can be visualized through a orbit diagram. The system parameters are the neuron selectors parameters.

The system initial condition is the initial neuron input. The qRAM chaotic configuration found in this paper should help artificial computing designers to fit neurons to invoke specific chaotic behaviour or oscillatory dynamics. The experimental results show the qualitatively changing of dynamical critical points during its parameter variation.

This paper is organized as follows: Section 2 presents basic concepts of quantum computing; Section 3 describes the working of quantum weightless neuron nodes; Section 4 explains the **REM** dynamics and in Section 5 the proposed method **QEM** dynamics is presented; Section 6 shows the experiments of the **QEM** dynamics chaotic configuration where it is possible to see orbit diagrams showing bifurcations and chaos. The conclusion is presented in Section 7.

2. Quantum Computing

A quantum bit, qubit, is a complex bidimensional unit vector. In spite of fact that 0 and 1 bits can be represented by any orthogonal base of $\mathbb{C}^2$, the mostly used one is the *canonical* (or *computational*) basis defined as the pair of vectors $|0\rangle = [1, 0]^T$ and $|1\rangle = [0, 1]^T$. A qubit $|\psi\rangle$ can be written as shown in Equation 1, where α and β are complex numbers and $|\alpha|^2 + |\beta|^2 = 1$. In quantum computing it is not possible to make a copy of an unknown state [27]. Composite quantum systems are formed using tensor product: $|ij\rangle = |i\rangle \otimes |j\rangle$.

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

A quantum system is altered by operator or a set of operators that change the qubit amplitude values. A quantum operator $\mathbf{U}$ over n qubits is a $2^n \times 2^n$ complex unitary matrix. Some operators over one qubit are the identity operator $\mathbf{I}$, not operator $\mathbf{X}$ and Hadamard $\mathbf{H}$ operator, described in Equation (2) and Equation (3). A quantum circuit is a combination of operators applied to one or some set of qubits.

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{matrix} \mathbf{I}|0\rangle = |0\rangle \\ \mathbf{I}|1\rangle = |1\rangle \end{matrix} \quad \mathbf{X} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{matrix} \mathbf{X}|0\rangle = |1\rangle \\ \mathbf{X}|1\rangle = |0\rangle \end{matrix} \tag{2}$$

$$\mathbf{H} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{matrix} \mathbf{H}|0\rangle = 1/\sqrt{2}(|0\rangle + |1\rangle) \\ \mathbf{H}|1\rangle = 1/\sqrt{2}(|0\rangle - |1\rangle) \end{matrix} \tag{3}$$

The identity operator $\mathbf{I}$ outputs the input; flip operator $\mathbf{X}$ behaves as the classical NOT on the computational basis; Hadamard transformation $\mathbf{H}$

generates superposition of states. The **CNOT** operator has 2 input qubits and 2 output qubits and flips the second qubit if the first one is 1 as show in Figure 1.

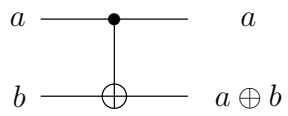


Figure 1: Circuit representation of **CNOT** operator.

Usually the operators are unitary but many non-unitary operators have been used [28]. The non-linearity is useful for overcoming "the difficulties connected with the ordinary linear quantum components" [16]. In this paper, a non-unitary operator is used to filter the dynamics amplitudes.

3. Quantum Weightless Networks

The quantum RAM based neuron qRAM [29] was defined as the quantisation of the RAM node which is the neural unit of the weightless neural networks first proposed in [30] and reviewed in [31].

In its simplest form a RAM node stores in its memory one bit addressed by an input bit string. The corresponding qRAM represents the stored bit by *selectors*. To change that selector values will simulate the changing of the stored value in the RAM memory. In spite of the simplicity of the RAM-based nodes, RAM-based networks have good generalisation and computational power and they have been used in several of machine learning applications [32, 33, 34, 35, 36, 37].

Rather than giving a general definition (see [29]) let us look at the particular one input neuron which we have been analysing [23, 18, 38, 39]. The circuit for the qRAM node is represented in Figure 2. Given a qubit, $|\psi\rangle$, in the input register, a qRAM node uses a controlled operation to choose which the operator, A_1 or A_2, will be applied to the selector registers, $|s_0 s_1\rangle$, and output register (a qubit which will hold the output, $|o'\rangle$), initially set to $|o\rangle = |0\rangle$. Each A_i is a CNOT operation and we use this notation A because the possible generalizations described in [40]. If the chosen selector value, $|s_0\rangle$ or $|s_1\rangle$, is one, the output register will be one, $|1\rangle$. Otherwise, the output register will be zero, $|0\rangle$, as shown in Figure 3. If the input qubit is in

superposition state, the selectors will be chosen in superposition, resulting in an output register in superposition.

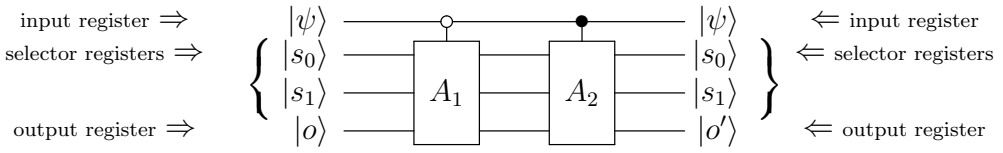


Figure 2: qRAM of one input.

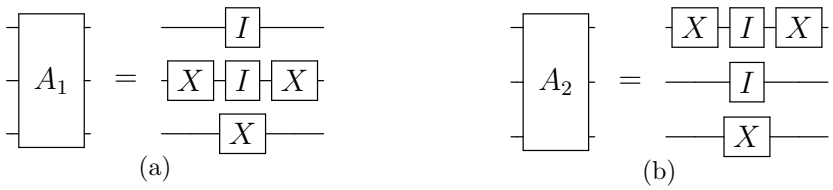


Figure 3: Structure of qRAM A operators

To train a qRAM circuit one needs to change the selectors values to correctly output to an input in a pattern set [17]. That method was generalised for arbitrary operators acting in a convenient vector space in [41].

4. Real Extraction Model

For iterate a system, we need to extract the system output and feed it back in the system input. An extraction model of the output qubit for using in its dynamics was proposed in [23]. The **REM** method extracts the amplitudes norms to build the output qubit. That model is explained in this section. In the next section, another qRAM dynamics model named **QEM** is proposed considering the summation of the squared amplitudes to build the output qubit. **QEM** allows that the dynamics manipulates complex numbers generated during its iteration.

Given a quantum state of size n represented by its superposition basis states

$$|\psi\rangle = a_0|\mathbf{0}\rangle + a_1|\mathbf{1}\rangle + a_2|\mathbf{2}\rangle ... a_n|\mathbf{n}\rangle \tag{4}$$

to check the real values of $|0\rangle$ and $|1\rangle$ amplitudes of the last qubit, we calculate the probabilities of those events occur. It can be done for

$$\alpha^2 = \sum_{i\ even} |a_i|^2 \qquad\qquad \beta^2 = \sum_{i\ odd} |a_i|^2$$

In that way, the output quantum state of the last qubit recovered is

$$|o\rangle_{t+1} = \alpha_{t+1}|0\rangle + \beta_{t+1}|1\rangle \tag{5}$$

The example below shows an output extraction of a qRAM node with one input. Given the following qubits

$$|i\rangle = \begin{pmatrix} a \\ b \end{pmatrix}, |s_0\rangle = \begin{pmatrix} c \\ d \end{pmatrix}\ , \quad |s_1\rangle = \begin{pmatrix} e \\ f \end{pmatrix}, |o\rangle = \begin{pmatrix} g \\ h \end{pmatrix} \tag{6}$$

where those amplitudes are complex numbers. Considering the composition of qubits $|\psi\rangle = |i\rangle|s_0\rangle|s_1\rangle|o\rangle$:

$$\begin{aligned}|\psi\rangle = {} & aceg|0000\rangle + aceh|0001\rangle + acfg|0010\rangle + acfh|0011\rangle + adeg|0100\rangle + \\ & adeh|0101\rangle + adfg|0110\rangle + adfh|0111\rangle + bceg|1000\rangle + bceh|1001\rangle + \\ & bcfg|1010\rangle + bcfh|1011\rangle + bdeg|1100\rangle + bdeh|1101\rangle + bdfg|1110\rangle + bdfh|1111\rangle\end{aligned} \tag{7}$$

When the qRAM node operator N [22] is applied in the $|\psi\rangle$, we have:

$$\begin{aligned}\mathsf{N}|\psi\rangle = |\psi'\rangle = {} & aceg|0000\rangle + aceh|0001\rangle + acfg|0010\rangle + acfh|0011\rangle + adeh|0100\rangle + \\ & adeg|0101\rangle + adfh|0110\rangle + adfg|0111\rangle + bceg|1000\rangle + bceh|1001\rangle + bcfh|1010\rangle \\ & + bcfg|1011\rangle + bdeg|1100\rangle + bdeh|1101\rangle + bdfh|1110\rangle + bdfg|1111\rangle\end{aligned} \tag{8}$$

The output qubit amplitudes can be extracted by:

$$\alpha^2 = \sum_{i\ even} |\psi'_i|^2 \qquad\qquad \beta^2 = \sum_{i\ odd} |\psi'_i|^2$$

because the last qubit or output qubit is $|0\rangle$ in the even $|\psi'\rangle$ amplitude positions and is $|1\rangle$ in odd $|\psi'\rangle$ amplitude positions. For the case described above, the output qubit recovered amplitudes α and β are:

$$\begin{aligned}|\alpha|^2_{t+1} =&|aceg|^2 + |acfg|^2 + |adeh|^2 + |adfh|^2 + \\ &|bceg|^2 + |bcfh|^2 + |bdeg|^2 + |bdfh|^2\end{aligned} \tag{9}$$

$$\begin{aligned}|\beta|^2_{t+1} =&|aceh|^2 + |acfh|^2 + |adeg|^2 + |adfg|^2 + \\ &|bceh|^2 + |bcfg|^2 + |bdeh|^2 + |bdfg|^2\end{aligned} \tag{10}$$

Then it is possible to recover the output qubit for feed back itself as input qubit in the next iteration:

$$|o\rangle_{t+1} = \alpha_{t+1}|0\rangle + \beta_{t+1}|1\rangle \tag{11}$$

The main characteristic of **REM** model is the feedback real values. The parameters of the system is the neuron parameters and the input of the system is the input neuron state. The iterated quantum system state is not in predetermined format as it is done in other works [24, 25, 26]. In the next Section, a modification of that model is proposed, where the summation of squared amplitudes are considered, being possible to manipulate complex values and to enrich the dynamics.

5. Quadratic Extraction Model

The Real Extraction Method allows the dynamics to deal with only real values, since only the module of values are added. Here, the **REM** method is extended to recover the value of the output register through the summation of the squared amplitudes and not the summation of squared module of the amplitudes. This model is not statistical but it is built in a quantum circuit with projective operators.

The following steps show firstly how to square the amplitudes of a quantum state. After that, we see how to filter the amplitudes and to sum some of them. The main idea is to use a pair of the same neuron output qubit as in [42, 24] and to flag the states of interest to erase all other ones with a projective operator. Finally, we use a non-unitary operator to choose the amplitudes of interest to build the output qubit.

5.1. Squaring amplitudes

We use a pair of the same quantum state after the qRAM operation [42] to square its amplitudes. Considering the result of the qRAM operation: qRAM $|\psi\rangle = a_0|\mathbf{0}\rangle + a_1|\mathbf{1}\rangle + \cdots + a_{15}|\mathbf{15}\rangle$, when we put together the pair (qRAM $|\psi\rangle$) (qRAM $|\psi\rangle$) $= |\phi\rangle$, the resulting state is:

$$|\phi\rangle = a_0a_0|\mathbf{0}\rangle|\mathbf{0}\rangle + a_0a_1|\mathbf{0}\rangle|\mathbf{1}\rangle + \cdots + a_0a_{15}|\mathbf{0}\rangle|\mathbf{15}\rangle + \cdots + a_{15}a_{15}|\mathbf{15}\rangle|\mathbf{15}\rangle \quad (12)$$

or in its binary representation

$$|\phi\rangle = a_0a_0|0000\rangle|0000\rangle + a_0a_1|0000\rangle|0001\rangle + \cdots + a_{15}a_{15}|1111\rangle|1111\rangle. \quad (13)$$

In Equation 13, we need to erase the amplitudes a_ia_j where $i \neq j$. One see that the cases where $|\phi\rangle$ has a_ja_j amplitudes are in the states $|\mathbf{j}\rangle|\mathbf{j}\rangle$.

5.2. Flagging interested states

One need to flag the states where the first four qubits are the same as the last four qubits. We use the CNOT operator four times, where the control qubits are the b-th qubits and their target qubits respectively are the $(b+4)$-th qubits, for $0 \leq b < 4$. The CNOT operator will set the last four qubits of the $|\mathbf{j}\rangle|\mathbf{j}\rangle$ states to $|0000\rangle$.

5.3. Erasing non-interesting states

To erase the states which their amplitudes are not in the form $a_j * a_j$, we use the projective operator $I^{\otimes 2}P_0^{\otimes 2}$ inspired in [42]. Those states that are flagged in the previous step will not be altered, keeping themselves after the projective operator and all other ones will be erased. The $I^{\otimes 2}$ operator is the 4x4 identity matrix and $P_0^{\otimes 2}$ is the matrix $(|0\rangle\langle 0|)^{\otimes 2}$.

The result of the projective operation is the $\left|\hat{\psi}\right\rangle|0000\rangle$ qubit state, where the $\left|\hat{\psi}\right\rangle$ state is the qRAM $|\psi\rangle$ state with squared amplitudes. The squared operation is represented in circuit in Figure 4.

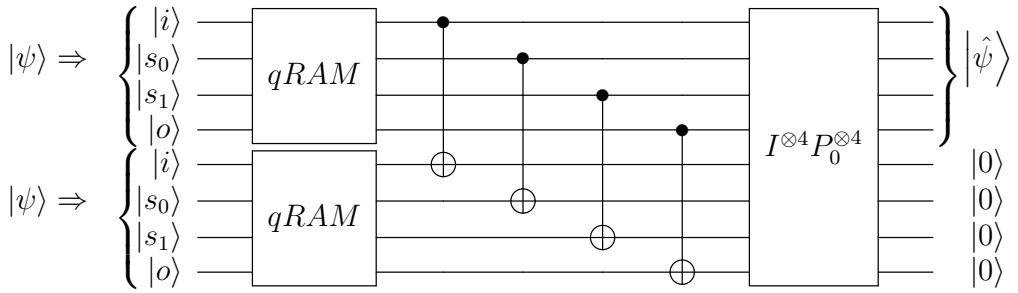


Figure 4: Quantum circuit to square the output amplitudes of the qRAM neuron. The $|\psi\rangle$ state is created from known initial conditions - this is not a cloning.

Finally, to extract the output qubit of the $\left|\hat{\psi}\right\rangle$, we apply the non-unitary O_f operator. The output filtering is done by O_f Operator which selects some amplitudes and combines them.

$$O_f = \begin{pmatrix} p_{(0,0)} & p_{(0,1)} & p_{(0,2)} & \cdots & p_{(0,j)} & \cdots & p_{(0,n)} \\ p_{(1,0)} & p_{(1,1)} & p_{(1,,2)} & \cdots & p_{(1,j)} & \cdots & p_{(1,n)} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{14}$$

When one needs to add the even (odd) amplitudes and assign as $|0\rangle$ ($|1\rangle$) state, the configuration of O_f matrix is $p_{(0,i)} = p_{(1,i+1)} = 1$ and $p_{(0,j)} = p_{(1,j+1)} = 0$, where i is even and j is odd, and $0 \leq i, j \leq 15$.

The circuit of filtering operation is represented in Figure 5 showing the position of the extracted output .

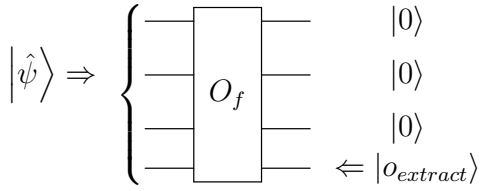


Figure 5: Isolating the output qubit for the dynamics.

5.4. Normalizing the output qubit

In **REM** dynamics, the dynamics feedback is always a real value, restricting the annihilation of complex values. Complex values is possible in the dynamics of **QEM**, but it has so many divergent orbits. To control that divergence, we normalize the extracted output quantum state as used in quantum dynamics in previous works [25, 24].

Then the final output qubit $|o_{extract}\rangle$ is

$$|o_{extract}\rangle = N(\sum_i a_i^2 p_{(0,i)}|0\rangle + \sum_i a_i^2 p_{(1,i)}|1\rangle) \tag{15}$$

which can be feed it back in the input circuit again for the dynamics iteration as represented by Figure 6. The N operator is the normalize operator. Through that extraction method (QEM), the extracted output qubit is now the sum of the squared complex amplitudes, in contrast to [23], where only the real values are feedback as input.

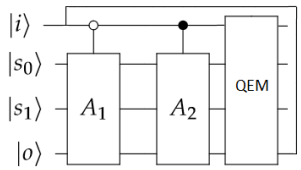


Figure 6: Quadratic Extraction Method (QEM) of a qRAM dynamics.

5.5. Generalisation of Quadratic Quantum Circuit Dynamics

It is easy to generalise the **QEM** for any quantum circuit C with n inputs. This method can then be used to generate an arbitrary nonlinear chaotic dynamics [9].

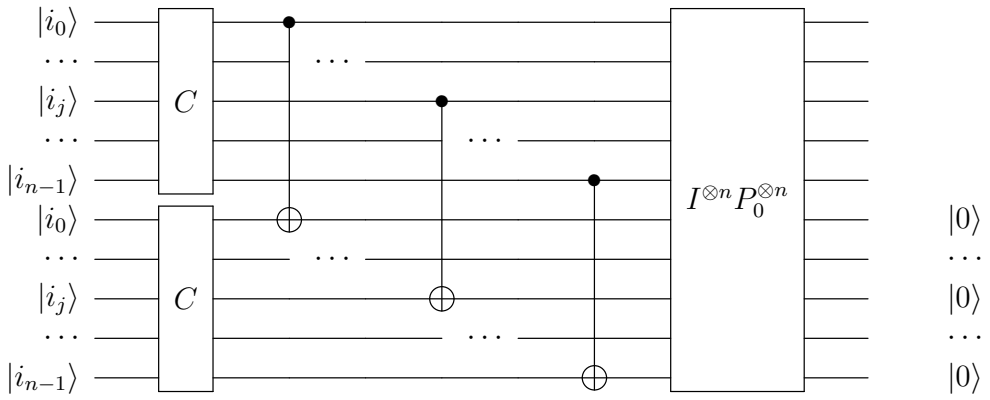


Figure 7: Quantum circuit to square the output state amplitudes of an arbitrary quantum circuit.

If the quantum circuit C has an output register, as the qRAM circuit, it is possible to extract it using a $2^n \times 2^n$ O_f operator. Through that proposed extraction (QEM), the extracted output qubit is now the sum of the quadratic of the amplitudes, being possible to explore the complex values during its working for any circuit.

6. Experiments

In this Section, experiments are done using **QEM** dynamics environment to invoke chaos during its dynamics. During the **REM** dynamics, we did not encounter chaos in orbit diagrams. In this Section we show the orbit diagrams with chaos when the **QEM** dynamics is used. Three steps of **QEM** dynamics are followed: (a) first, the amplitudes are squared through the circuit explained in previous section; (b) after that, the qubit is filtered by the O_f operator which some output qubit amplitudes are passed to build $|0\rangle$ and $|1\rangle$ states; (c) the output extracted qubit is normalized.

Values for the p index of the O_f are chosen experimentally. Considering $p_{(0,k)} = 1$, $k \in \{0, 1, 2, 7, 8, 11, 12\}$, $p_{(0,m)} = -1$, $m \in \{5, 15\}$, $p_{(1,g)} = 1$, for $g \in \{1, 5, 11\}$, $p_{(1,w)} = -1$, for $w \in \{7, 15\}$, all other positions equal to zero of filtering operator O_f, the **QEM** dynamics formula for output qubit in time $t + 1$ is

$$\begin{cases} x_{c,e}^{t+1} = f(x_t, y_t, c, e) & = 2.0c^2e^2x_t^2 - 2.0c^2e^2y_t^2 + 2.0c^2y_t^2 \\ & -2.0e^2x_t^2 + 2.0e^2y_t^2 + 1.0x_t^2 - 1.0y_t^2 \\ y_{c,e}^{t+1} = g(x_t, y_t, c, e) & = c^2y_t^2(-e^2+1) + e^2x_t^2(-c^2+1) \\ & -x_t^2(c^2-1)(e^2-1) - y_t^2(c^2-1)(e^2-1) \\ |o\rangle_{c,e}^{t+1} & = x_{c,e}^{t+1}|0\rangle + y_{c,e}^{t+1}|1\rangle \end{cases} \tag{16}$$

where the output qubit in time t ($|o\rangle_t = x_t|0\rangle + y_t|1\rangle$) is feedback in the input qubit $|i\rangle$ in the time $t+1$ and the selectors have the configuration $|s_0\rangle = c|0\rangle + \sqrt{1-|c|^2}|1\rangle$ and $|s_1\rangle = e|0\rangle + \sqrt{1-|e|^2}|1\rangle$.

Note that many other combinations result is possible through the O_f filtering operation. Further works must be targeted in future to generalize the O_f generation. Those values for this experiment were chosen empirically. Even using a combination of amplitudes, as it is done in O_f operator, the **REM** dynamics has not encounter chaos in its orbit diagrams.

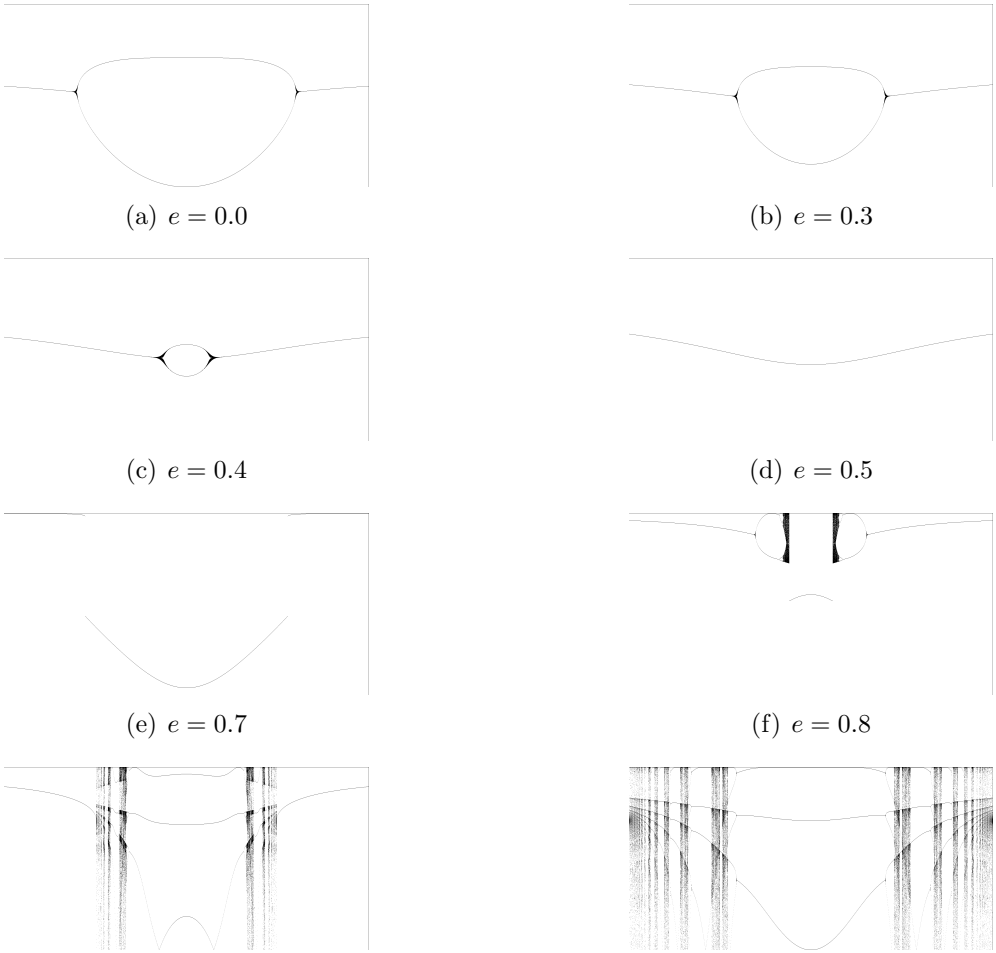
To analyse the dynamics behaviour that has two parameters (c and e) we set one as constant and variate the other one [38]. The experiments show chaotic behaviour from the neuron setting proposed. Through the orbit diagrams we can check the critical bifurcation points modifications qualitatively and quantitatively.

In the orbit diagram we plot the parameters c (Figure 8) or e (Figure 9) (selectors amplitude values) on the horizontal axis and in vertical axis the asymptotic values of $x_{c,e}^t$. By asymptotic points it means that one do not plot the first few iterations (usually 100, as in this experiment) and plot the next 1000 points. By not plotting the first few iterations, we have eliminated the "transient behavior" of the orbit [43]. In this Section, j is the complex number $\sqrt{-1}$.

Analysing the orbit diagram of **QEM** dynamics, one see that the bifurcation points vary by the parameters changing as in the bifurcation points in Figures 8(a-c). The movement of critical points allows to collapse themselves and also to generate chaos. In Figure 8(f-h), one see the chaos appearance when parameter e is initialized in range $[0.8, 1.0]$ value and c is modified between $-j$ and j. In Figures 8(a-e) the orbit diagrams present few critical points and for $e > 0.7$ chaos and many critical points appear in Figures 8(f-h).

In Figure 9, the orbit diagrams are presented varying the e parameter and considering the c parameter as a constant. All orbit diagram have $x_0 = 1.0$. The chaos orbits starts to appear in the extremities of e values Figures 9(b-f)

(either next to $e = 0.0$ or next to $e = j$). As c is increased, the movement of critical points does the chaos orbits collapse themselves and appear only one critical point after $c > 0.8$.



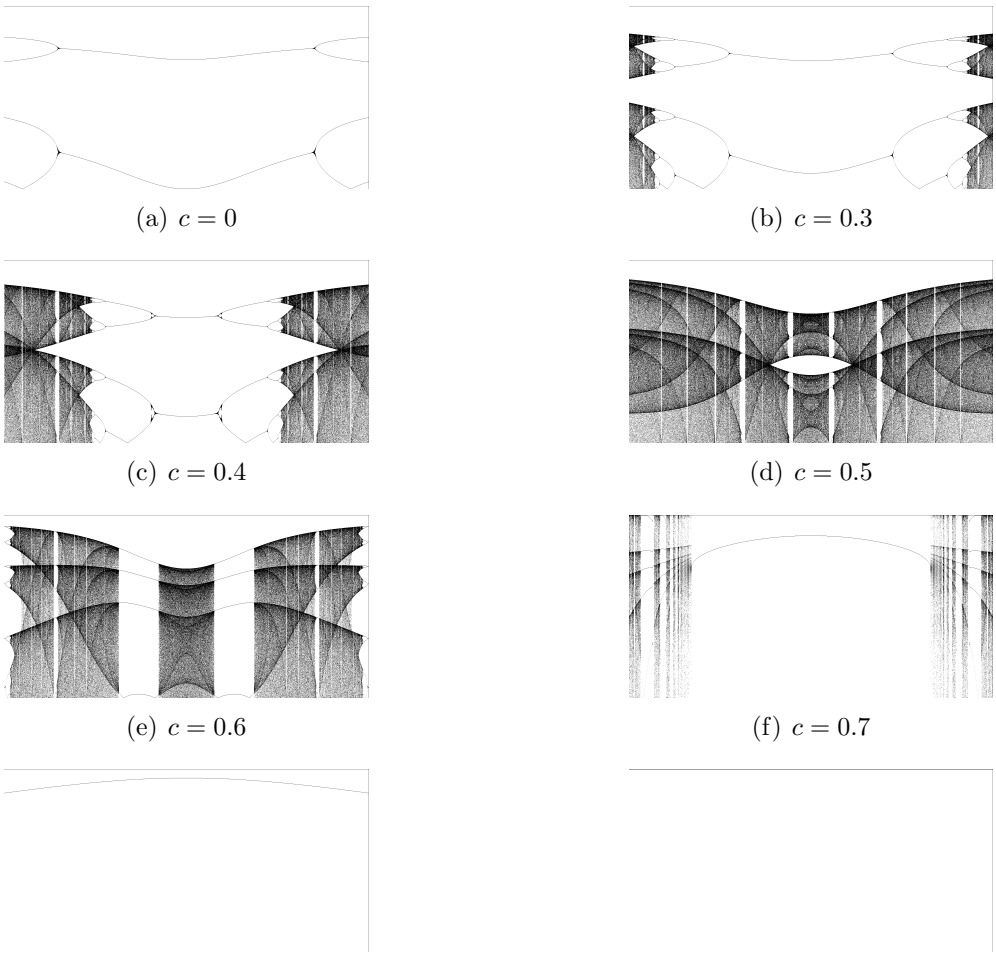
(a) $e = 0.0$ (b) $e = 0.3$

(c) $e = 0.4$ (d) $e = 0.5$

(e) $e = 0.7$ (f) $e = 0.8$

(g) $e = 0.9$ (h) $e = 1.0$

Figure 8: Orbit diagrams where e parameter value is fixed and c varies between $-j$ and j. There is a movement of dynamics critical points as the e parameter is increased and the appearance of chaos in $e > 0.7$.



(a) $c = 0$ (b) $c = 0.3$

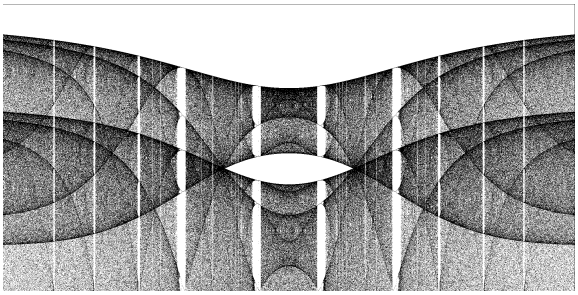
(c) $c = 0.4$ (d) $c = 0.5$

(e) $c = 0.6$ (f) $c = 0.7$

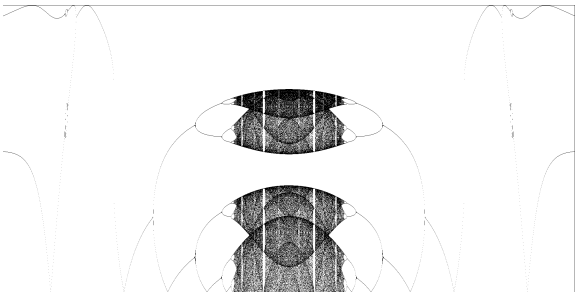
(g) $c = 0.8$ (h) $c = 1.0$

Figure 9: Orbit diagrams which c parameter is fixed and e varies between $-j$ and j. Bifurcations and chaos are present in $c \leq 0.7$. After that the dynamics is transformed in an unique critical point.

We plot in Figure 10 two orbit diagrams where $c = 0.5$ and e varies between $[-j, j]$ and $[-1, 1]$ respectively. It is possible to see that the dynamics is very sensitive to complex values, amplifying its bifurcation power.



(a) $c = 0.5$ and e varies between $-j$ and j.



(b) $c = 0.5$ and e varies between -1 and 1.

Figure 10: Neuron Chaotic Orbit Diagrams showing its bifurcation sketches and chaos zones where e is varied in real and complex values respectively. In Figure 10(a) one observes the complexity of chaos orbits in almost all e value, where the dark lines are the bifurcation points flattened as c is increased. In Figure 10(b) the e varies in real values showing amount of bifurcation points and chaos in its center. The figures show the sensibility of that dynamics to its parameter values.

Two orbits are plotted in Figure 11 to show the dynamics sensibility to initial conditions. A difference of initial conditions in a decimal case invokes different behaviour during its dynamics.

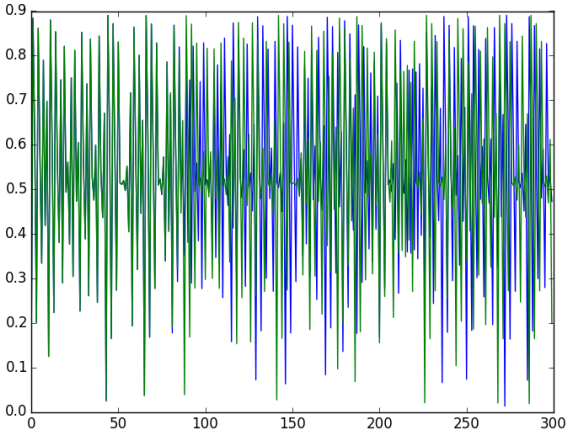


Figure 11: Two orbit series starting with initial conditions $x = 0.1$ and $x = 0.1000000000000001$ as chaotic parameter $c = 0.5$ and $e = 0.96j$ and differing completely with the other next to $90{-}th$ iteration.

6.1. Logic gates representation

Here we show that the **QEM** circuit can represent logic gates after a simple classical training algorithm that learns the qRAM parameters. We use the simple delta rule [44] to train the parameters of the qRAM circuit, c and e selectors values which are set initially with random values. This experiment considers the two input variables i_1 and i_2 as the input quantum state in the format $|i\rangle = i_1|0\rangle + i_2|1\rangle$. The **QEM** circuit outputs the logic operation result in the $|0\rangle$ amplitude of the extracted output state.

The training sets are $\Omega_{AND} = \{\{0,0\},0\}, \{\{0,1\},0\}, \{\{1,0\},0\}, \{\{1,1\},1\}$, $\Omega_{OR} = \{\{0,0\},0\}, \{\{0,1\},1\}, \{\{1,0\},1\}, \{\{1,1\},1\}$ and $\Omega_{XOR} = \{\{0,0\},0\}, \{\{0,1\},1\}, \{\{1,0\},1\}, \{\{1,1\},0\}$. The procedure to train the parameters of the qRAM is updating the c and e values through delta rule described in Equation 17 below

$$w_{t+1} = w_t + \eta * (\hat{y} - y) * x_i \tag{17}$$

where w is the parameter c and e, η is the learning rate empirically defined as 0.1, $\hat{y}$ the system output and y the expected value from the training set. In some steps of learning, the c and e parameters are adjusted and the desired configuration is encountered. To compare the result with the training set examples the amplitude of $|0\rangle$ is considered 1 if it is bigger than 0.5 or 0

otherwise. If all the examples, in the training set, match with the amplitude output qubit, the algorithm ends.

In that way, the **QEM** circuit can represent the AND, OR and XOR logic gates. As example, if the parameters are set randomly as $c = 0.502741811814$ and $e = 0.923462993432$, the algorithm executes 14 times to encounter the parameters to match the Ω_{AND} training set, $c = 0.323571712677$ and $e = 0.744292894296$. Similarly, if the parameters are set as $c = 0.00812963753783$ and $e = 0.830240020655$, in 4 iterations the training algorithm encounters $c = -0.600648681131$ and $e = 0.221461701987$ to solve the Ω_{OR} training set. For the last, if $c = 0.852644066708$ and $e = 0.974984538593$, the algorithm executes 13 times to find $c = 1.55419414381$ and $e = 1.6765346157$ and solves the Ω_{XOR} problem.

7. Conclusion

In this work, we present a new quantum qubit extraction model named **QEM** to evaluate a qRAM dynamics. **QEM** allows complex values iterations and a qRAM chaotic dynamics configuration is encountered in its orbit diagram, unlike the **REM** model. The result of this work should be used as parameter configuration for neuron learning since the chaos and bifurcation are present and many previous works demonstrated the relation between chaos and learnability.

The **QEM** dynamics was inspired in [23, 38, 25] works where dynamics were analytically studied. The results show chaotic behaviour easily viewed experimentally and the critical points collapsing during the parameter variation. The chaos sensibility of initial conditions as shown in Figure 11 allows either to cryptography images or texts [45, 46] or to understand the role of biological networks [47, 48]. The extraction model showed to represent the AND, OR and XOR logic gates through a simple training algorithm.

A possible future is about a generalization of the filtering operator O_f to understand quantitatively the producing of chaos in that dynamics. The analysis of quantum weightless neuron networks dynamics are also expected. Another possible future work is to train (quantum weightless) neural networks using the information of chaotic parametrization found in this work [49].

Bibliography

[1] A. V. M. Herz, T. Gollisch, C. K. Machens, D. Jaeger, Modeling Single-Neuron Dynamics and Computations: A Balance of Detail and Abstraction, Science 314 (5796) (2006) 80–85.

[2] F. Pasemann, Dynamics of a single model neuron, International Journal of Bifurcation and Chaos 3 (02) (1993) 271–278.

[3] A. Garliauskas, Neural network chaos analysis, Nonlinear Anal., Model. and Control 3 (1998) 43–57.

[4] K. Matsumoto, I. Tsuda, Calculation of information flow rate from mutual information, Journal of Physics A: Mathematical and General 21 (6) (1988) 1405.

[5] D. Gross, The importance of chaos theory in the development of artificial neural systems (Online in 2015).
URL `http://www.lycaeum.org/ sputnik/Misc/chaos.html`

[6] A. Garliauskas, An influence of nonlinearities to storage capacity of neural networks., Informatica, Lith. Acad. Sci. 16 (2) (2005) 159–174.

[7] W. J. Freeman, Tutorial on neurobiology: from single neurons to brain chaos, International journal of bifurcation and chaos 2 (03) (1992) 451–482.

[8] W. J. Freeman, S. Jakubith, Bifurcation analysis of continuous time dynamics of oscillatory neural networks, Brain theory, Elsevier Science Publishers BV, Berlin, Heidelberg (1993) 183–207.

[9] S. H. Strogatz, Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity), Westview Press, 1994.

[10] M. Baptista, Cryptography with chaos, Physics Letters A 240 (1–2) (1998) 50 – 54.

[11] B. Chaudhuri, N. Sarkar, Texture segmentation using fractal dimension, Pattern Analysis and Machine Intelligence, IEEE Transactions on 17 (1) (1995) 72–77.

[12] J. M. Keller, S. Chen, R. M. Crownover, Texture description and segmentation through fractal geometry, Computer Vision, Graphics, and Image Processing 45 (2) (1989) 150 – 166.

[13] O. Kinouchi, M. Copelli, Optimal dynamical range of excitable networks at criticality, Nature Physics 2 (5) (2006) 348–352.

[14] J. Torres, J. Marro, Brain performance versus phase transitions, Scientific Reports 5 (12216) (2015) 1–10.

[15] R. Zhou, Q. Ding, Quantum m-p neural network, International Journal of Theoretical Physics 46 (12) (2007) 3209–3215.

[16] M. Panella, G. Martinelli, Neural networks with quantum architecture and quantum learning, International Journal of Circuit Theory and Applications 39 (1) (2011) 61–77.

[17] A. J. da Silva, W. R. de Oliveira, T. B. Ludermir, Classical and superposed learning for quantum weightless neural networks, Neurocomputing 75 (1) (2012) 52–60.

[18] F. M. de Paula Neto, A. J. da Silva, T. B. Ludermir, W. R. De Oliveira, Analysis of quantum neural models, 11th Brazilian Congress on Computational Intelligence CBIC (2013) 1–6.

[19] F. Li, S. Zhao, B. Zheng, Quantum neural network in speech recognition, in: Signal Processing, 2002 6th International Conference on, Vol. 2, IEEE, 2002, pp. 1267–1270.

[20] A. Manju, M. J. Nigam, Applications of quantum inspired computational intelligence: a survey, Artificial Intelligence Review 42 (1) (2014) 79–156.

[21] C.-J. Lin, I.-F. Chung, C.-H. Chen, An entropy-based quantum neuro-fuzzy inference system for classification applications, Neurocomputing 70 (13–15) (2007) 2502–2516.

[22] W. R. de Oliveira, Quantum RAM Based Neural Networks, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (2009) 22–24.

[23] F. M. de Paula Neto, W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, Chaos in Quantum Weightless Neuron Node Dynamics, Neurocomputing 183 (2015) 23–38.

[24] T. Kiss, S. Vymětal, L. Tóth, A. Gábris, I. Jex, G. Alber, Measurement-induced chaos with entangled states, Physical review letters 107 (10) (2011) 100501.

[25] T. Kiss, I. Jex, G. Alber, S. Vymetal, Complex chaos in the conditional dynamics of qubits, Phys. Rev. A 74 (2006) 040301.

[26] T. Kiss, I. Jex, G. Alber, S. Vymêtal, Complex chaos in conditional qubit dynamics and purification protocols, Acta Physica Hungarica Series B, Quantum Electronics 26 (3) (2006) 229–235.

[27] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, 2000.

[28] D. S. Abrams, S. Lloyd, Nonlinear Quantum Mechanics Implies Polynomial-Time Solution for NP-Complete and P Problems, Phys. Rev. Lett. 81 (18) (1998) 3992–3995.

[29] W. R. de Oliveira, Quantum RAM based neural networks, in: M. Verleysen (Ed.), ESANN'09: Advances in Computational Intelligence and Learning, 2009, pp. 331–336.

[30] I. Aleksander, Self-adaptive universal logic circuits, Electronics Letters 2 8 (1966) 321–322.

[31] F. M. G. França, M. D. Gregorio, P. M. V. Lima, W. R. de Oliveira, Advances in weightless neural systems, in: European Symposium on Artificial Neural Networks: Computational Intelligence and Machine, pp. 497–504.

[32] M. C. P. de Souto, T. B. Ludermir, W. R. de Oliveira, Equivalence between RAM-based neural networks and probabilistic automata, IEEE Transactions on Neural Networks 16 (2005) 996–999.

[33] T. B. Ludermir, A. de Carvalho, A. P. Braga, M. C. P. de Souto, Weightless neural models: a review of current and past works, Neural Computing Surveys 2 (1989) 41–61.

[34] D. O. Cardoso, D. S. Carvalho, D. S. Alves, D. F. Souza, H. C. Carneiro, C. E. Pedreira, P. M. Lima, F. M. França, Financial credit analysis via a clustering weightless neural classifier, Neurocomputing (2016) 1–26.

[35] M. D. Gregorio, M. Giordano, S. Rossi, M. Staffa, Experimenting {WNN} support in object tracking systems, Neurocomputing (2015) 1–11.

[36] N. Nedjah, F. P. da Silva, A. O. de Sá, L. M. Mourelle, D. A. Bonilla, A massively parallel pipelined reconfigurable design for m-pln based neural networks for efficient image classification, Neurocomputing (2015) 1–17.

[37] M. Berger, A. F. D. Souza, J. de Oliveira Neto, E. de Aguiar, T. Oliveira-Santos, Visual tracking with vg-ram weightless neural networks, Neurocomputing (2016) 1–16.

[38] F. M. de Paula Neto, T. B. Ludermir, W. R. De Oliveira, A. J. da Silva, Fitting parameters on quantum weightless neuron dynamics, in: Intelligent Systems, 2015 Brazilian Conference on, Vol. 4, 2015, pp. 169–174.

[39] F. M. de Paula Neto, T. B. Ludermir, W. R. De Oliveira, A. J. da Silva, Solving np-complete problems using quantum weightless neuron nodes, in: Intelligent Systems, 2015 Brazilian Conference on, 2015, pp. 258–263.

[40] W. Oliveira, A. Silva, T. Ludermir, A. Leonel, W. Galindo, J. Pereira, Quantum logical neural networks, in: Neural Networks, 2008. SBRN '08. 10th Brazilian Symposium on, 2008, pp. 147–152.

[41] W. R. de Oliveira, A. J. da Silva, T. B. Ludermir, Vector space weightless neural networks, ESANN (2014) 535–540.

[42] H. Bechmann-Pasquinucci, B. Huttner, N. Gisin, Non-linear quantum state transformation of spin-1/2, Physics Letters A 242 (1998) 198–204.

[43] R. L. Devaney, A first course in chaotic dynamical systems: theory and experiment, Westview Press, 1992.

[44] B. Widorw, M. E. Hoff, Adaptive switching circuits., Institute of Radio Engineers, Western Electronic Show and Convention Record (Part 4) (1960) 96–104.

[45] Y. Wang, K.-W. Wong, X. Liao, T. Xiang, G. Chen, A chaos-based image encryption algorithm with variable control parameters, Chaos, Solitons & Fractals 41 (4) (2009) 1773 – 1783.

[46] Y. Wang, K.-W. Wong, X. Liao, G. Chen, A new chaos-based fast image encryption algorithm, Applied Soft Computing 11 (1) (2011) 514 – 522.

[47] Y. Yao, W. J. Freeman, Model of biological pattern recognition with spatially chaotic dynamics, Neural Networks 3 (2) (1990) 153–170.

[48] T. J. Gordon, Chaos in social systems, Technological Forecasting and Social Change 42 (1) (1992) 1–15.

[49] I. Tsuda, Chaotic itinerancy and its roles in cognitive neurodynamics, Current Opinion in Neurobiology 31 (2015) 67–71, sI: Brain rhythms and dynamic coordination.

6

Conclusion

The artificial intelligence area has proposed increasingly varied and complexes intelligent models. On the other hand, there are few studies about how those models behave in time with their parameter variations and initial conditions. In this work, we studied the iteration and some parametrization conditions of the quantum weightless neuron qRAM dynamics.

To investigate a neuron dynamics, we need to feedback the system output in the input iteratively. In this work, we showed in Theorem 6.1 (Chapter 2) that there is no way to recovery the output qubit after a Controlled-Not operation, even in analytical mathematical manipulation, because the system can be entangled. The output system information is then recovered for two proposed novel methods in this work: (a) in Chapter 2, trough the Extraction Model, where the recovered output is get by a statistical method which has phase loss; (b)in Chapter 5, trough a Quadratic Extraction Method, which sum the squared amplitudes considering the phase information using a non-unitary operator.

These two dynamics models were studied in this dissertation. The main contributions can be encountered: (a) in Chapter 2, the Real Extraction Dynamics Model is studied from the phase space where 3 behaviour are encountered: undamped / oscillatory, underdamped, overdamped. A measure to compare different dynamics is also proposed in the Chapter 2; (b) In Chapter 3, we analyzed analytical equations of the Real Extraction Model presented in Chapter 2 generating a configuration behaviour map in function of selectors values; (b) In Chapter 4, we use the qRAM attached with a nonunitary operator to solve a 3-SAT Problem. Each qRAM represents the logic function clause use to represent a 3-SAT expression. With that representation, we showed the powerful of non-unitary operators when used with neuron nodes; (c) In Chapter 5, a chaotic dynamics configuration is presented using the Quadratic Extraction Model. Orbits diagram are showed presenting chaos behaviour and bifurcation sensibility. We perceived that as parameters are increased, the fixed points are in movement and chaotic zones appears with parametric sensibility. Chaotic zones can be used in future to solve complex problems as OHYA; VOLOVICH (1999) did.

6.1 Main Results

The main contributions of this works:

- From the identification of physics structure limitation due information entanglement generated by the CNOT Operators, the proposition of two methods of quantum information extraction for any quantum system. One model is purely real in its iteration and other one uses non-unitary operators where amplitudes of quantum states are squared and a dynamics with complex values is possible to be done;
- Behaviour categorization of convergence inside a phase space trough parametric changing of the Real Extraction Model Dynamics;
- Proposition of a quantity measure to compare dynamical systems of the Extraction Dynamics Model;
- Proposition of a non-unitary algorithm to solve NP-Complete Problems using qRAM nodes.
- Proposition of a chaotic configuration using the proposed Quadratic Extraction Model in the qRAM dynamics where there are bifurcation which can be used to improve the qRAM learning.

6.2 Future Works

This study can be continued in many branches of investigations:

- Dynamics analysis of a qRAM network considering how its topology influences the information management, entropy and information maximization (YUAN; XIONG; HUAI, 2003; KOUTSOUGERAS; PAPACHRISTOU, 1988; MöLLER; PEIXOTO, 2015).
- Dynamics categorical extrapolation on the view of the quantum operators, considering more than one qRAM and extending for circuit connexions with many layers (DEVANEY, 1992; STROGATZ, 1994; STÖCKMANN, 2007).
- Investigation of the recurrent quantum neuron networks and the proposition of models which training using with chaotic behaviour (GANDHI et al., 2015; GANJEFAR; TOFIGHI; KARAMI, 2015; ZHAO et al., 2015; STROGATZ, 1994).

References

BAPTISTA, M. Cryptography with chaos. **Physics Letters A**, Amsterdam, v.240, n.1–2, p.50 – 54, 1998.

BECHMANN-PASQUINUCCI, H.; HUTTNER, B.; GISIN, N. Non-linear quantum state transformation of spin-1/2. **Physics Letters A**, Amsterdam, v.242, p.198–204, 1998.

BEHERA, L.; KAR, I.; ELITZUR, A. C. A recurrent quantum neural network model to describe eye tracking of moving targets. **Foundations of Physics Letters**, USA, v.18, n.4, p.357–370, 2005.

BEHRMAN, E. et al. Simulations of quantum neural networks. **Information Sciences**, California, USA, v.128, n.3–4, p.257 – 269, 2000.

BRAGA, A. d. P.; CARVALHO, A.; LUDERMIR, T. B. **Redes neurais artificiais**: teoria e aplicações. [S.l.]: Livros Técnicos e Científicos, 2000.

CHAUDHURI, B.; SARKAR, N. Texture segmentation using fractal dimension. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, USA, v.17, n.1, p.72–77, Jan 1995.

DE SOUZA, A. F. et al. Automated multi-label text categorization with VG-RAM weightless neural networks. **Neurocomputing**, Amsterdam, v.72, n.10, p.2209–2217, 2009.

DEVANEY, R. L. **A first course in chaotic dynamical systems: theory and experiment**. USA: Westview Press, 1992.

FREEMAN, W. J. Tutorial on neurobiology: from single neurons to brain chaos. **International journal of bifurcation and chaos**, Singapore, v.2, n.03, p.451–482, 1992.

FREEMAN, W. J.; JAKUBITH, S. Bifurcation analysis of continuous time dynamics of oscillatory neural networks. **Brain theory**, Berlin, Heidelberg, p.183–207, 1993.

GANDHI, V. et al. Evaluating Quantum Neural Network filtered motor imagery brain-computer interface using multiple classification techniques. **Neurocomputing**, Amsterdam, v.170, p.161 – 167, 2015.

GANJEFAR, S.; TOFIGHI, M.; KARAMI, H. Fuzzy wavelet plus a quantum neural network as a design base for power system stability enhancement. **Neural Networks**, Amsterdam, v.71, p.172 – 181, 2015.

GARLIAUSKAS, A. Neural network chaos analysis. **Nonlinear Anal., Model. and Control**, [S.l.], v.3, p.43–57, 1998.

GROSS, D. **The Importance of Chaos Theory in the Development of Artificial Neural Systems**. Online em 2015.

HERZ, A. V. M. et al. Modeling Single-Neuron Dynamics and Computations: a balance of detail and abstraction. **Science**, USA, v.314, n.5796, p.80–85, 2006.

HOPFIELD, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. **Proceedings of the national academy of sciences**, USA, v.81, n.10, p.3088–3092, 1984.

KAK, S. On quantum neural computing. **Information Sciences**, California, USA, v.83, n.3–4, p.143 – 160, 1995.

KELLER, J. M.; CHEN, S.; CROWNOVER, R. M. Texture description and segmentation through fractal geometry. **Computer Vision, Graphics, and Image Processing**, Amterdam, v.45, n.2, p.150 – 166, 1989.

KINOUCHI, O.; COPELLI, M. Optimal dynamical range of excitable networks at criticality. **Nature Physics**, United Kingdom, v.2, n.5, p.348–352, 2006. cited By 193.

KISS, T. et al. Complex chaos in the conditional dynamics of qubits. **Phys. Rev. A**, Maryland, USA, v.74, p.040301, 2006.

KISS, T. et al. Complex chaos in conditional qubit dynamics and purification protocols. **Acta Physica Hungarica Series B, Quantum Electronics**, USA, v.26, n.3, p.229–235, 2006.

KISS, T. et al. Measurement-Induced Chaos with Entangled States. **Physical review letters**, Maryland, USA, v.107, n.10, p.100501, 2011.

KOUTSOUGERAS, C.; PAPACHRISTOU, C. Training of a neural network for pattern classification based on an entropy measure. In: NEURAL NETWORKS, 1988., IEEE INTERNATIONAL CONFERENCE ON, San Diego, California, USA. **Anais...** [S.l.: s.n.], 1988. v.1, p.247–254.

LI, F.; ZHAO, S.; ZHENG, B. Quantum neural network in speech recognition. In: SIGNAL PROCESSING, 2002 6TH INTERNATIONAL CONFERENCE ON, Beijing, China. **Anais...** IEEE, 2002. v.2, p.1267–1270.

LIN, C.-J.; CHUNG, I.-F.; CHEN, C.-H. An entropy-based quantum neuro-fuzzy inference system for classification applications. **Neurocomputing**, Amsterdam, Netherlands, v.70, n.13–15, p.2502 – 2516, 2007.

LUDERMIR, T. B. et al. Weightless neural models: a review of current and past works. **Neural Computing Surveys**, [S.l.], v.2, p.41–61, 1989.

MANJU, A.; NIGAM, M. J. Applications of quantum inspired computational intelligence: a survey. **Artificial Intelligence Review**, Berlin, Heidelberg, v.42, n.1, p.79–156, 2014.

MöLLER, M.; PEIXOTO, T. P. Maximum-entropy large-scale structures of Boolean networks optimized for criticality. **New Journal of Physics**, Bristol, England, v.17, n.4, p.043021, 2015.

MONTEIRO, L. **Sistemas dinâmicos**. São Paulo, Brazil: Editora Livraria da Física, 2006.

OHYA, M.; VOLOVICH, I. V. **Quantum Computing, NP-complete Problems and Chaotic Dynamics**. 1999.

OLIVEIRA, W. R. de. Quantum RAM Based Neural Networks. **ESANN**, Bruges, Belgium, p.22–24, 2009.

PANELLA, M.; MARTINELLI, G. Neural networks with quantum architecture and quantum learning. **International Journal of Circuit Theory and Applications**, [S.l.], v.39, n.1, p.61–77, 2011.

PASEMANN, F. Dynamics of a single model neuron. **International Journal of Bifurcation and Chaos**, Singapore, v.3, n.02, p.271–278, 1993.

PAULA NETO, F. M. de et al. Analysis of Quantum Neural Models. **11th Brazilian Congress on Computational Intelligence CBIS**, Porto de Galinhas, Brazil, 2013.

PAULA NETO, F. M. de et al. Fitting Parameters on Quantum Weightless Neuron Dynamics. In: INTELLIGENT SYSTEMS, 2015 BRAZILIAN CONFERENCE ON, Natal, Brazil. **Anais...** IEEE, 2015. v.4, p.169–174.

PAULA NETO, F. M. de et al. Solving NP-Complete Problems using Quantum Weightless Neuron Nodes. In: INTELLIGENT SYSTEMS, 2015 BRAZILIAN CONFERENCE ON, Natal, Brazil. **Anais...** IEEE, 2015. v.4, p.258–263.

PAULA NETO, F. M. de et al. Chaos in Quantum Weightless Neuron Node Dynamics. **Neurocomputing**, Amsterdam, v.183, p.23–38, 2016.

SCHLOSSHAUER, M. Decoherence, the measurement problem, and interpretations of quantum mechanics. **Rev. Mod. Phys.**, Maryland, USA, v.76, p.1267–1305, Feb 2005.

SILVA, A. J. da; LUDERMIR, T. B.; DE OLIVEIRA, W. R. On the universality of quantum logical neural networks. In: NEURAL NETWORKS (SBRN), 2012 BRAZILIAN SYMPOSIUM ON, Curitiba, Brazil. **Anais...** [S.l.: s.n.], 2012. p.102–106.

SILVA, A. J. da; OLIVEIRA, W. R. de; LUDERMIR, T. B. Classical and superposed learning for quantum weightless neural networks. **Neurocomputing**, Amsterdam, Netherlands, v.75, n.1, p.52–60, 2012.

SOUTO, M. C. P. de; LUDERMIR, T. B.; OLIVEIRA, W. R. de. Equivalence between RAM-based neural networks and probabilistic automata. **IEEE Transactions on Neural Networks**, New Jersey, USA, v.16, p.996–999, 2005.

STÖCKMANN, H. **Quantum Chaos**: an introduction. Cambridge, England: Cambridge University Press, 2007.

STROGATZ, S. H. **Nonlinear Dynamics And Chaos: with applications to physics, biology, chemistry, and engineering (studies in nonlinearity)**. Colorado,USA: Westview Press, 1994.

STROGATZ, S. H. **A Matemática do dia a dia**. Amsterdam: Elsevier Editora Ltda, 2012.

TERNO, D. R. Nonlinear operations in quantum-information theory. **Physical Review A**, Maryland, USA, v.59, n.5, p.3320, 1999.

TORRES, J.; MARRO, J. Brain Performance versus Phase Transitions. **Scientific Reports**, United Kingdom, v.5, 2015.

YUAN, H.; XIONG, F.; HUAI, X. A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy. **Computers and Electronics in Agriculture**, Amsterdam, v.40, n.1–3, p.57 – 64, 2003. Artificial Intelligence in Agriculture (Budapest).

ZHAO, P. et al. **Hyper-structure recurrent neural networks for text-to-speech**. 2015. n.20150364128.

ZHOU, R.; DING, Q. Quantum M-P Neural Network. **International Journal of Theoretical Physics**, Berlin, v.46, n.12, p.3209–3215, 2007.