

Article

Reservoir Computing Using Measurement-Controlled Quantum Dynamics

A. H. Abbas *  and Ivan S. Maksymov 

Artificial Intelligence and Cyber Futures Institute, Charles Sturt University, Bathurst, NSW 2795, Australia; imaksymov@csu.edu.au

* Correspondence: aborae@csu.edu.au

Abstract: Physical reservoir computing (RC) is a machine learning algorithm that employs the dynamics of a physical system to forecast highly nonlinear and chaotic phenomena. In this paper, we introduce a quantum RC system that employs the dynamics of a probed atom in a cavity. The atom experiences coherent driving at a particular rate, leading to a measurement-controlled quantum evolution. The proposed quantum reservoir can make fast and reliable forecasts using a small number of artificial neurons compared with the traditional RC algorithm. We theoretically validate the operation of the reservoir, demonstrating its potential to be used in error-tolerant applications, where approximate computing approaches may be used to make feasible forecasts in conditions of limited computational and energy resources.

Keywords: approximate computing; chaotic time series; machine learning; neural networks; reservoir computing; neuromorphic computing



Citation: Abbas, A.H.; Maksymov, I.S. Reservoir Computing Using Measurement-Controlled Quantum Dynamics. *Electronics* **2024**, *13*, 1164. <https://doi.org/10.3390/electronics13061164>

Academic Editors: Akash Kumar, Antoni Portero, Somnath Mazumdar, Alberto Scionti and Fujita Masahiro

Received: 28 February 2024

Revised: 19 March 2024

Accepted: 20 March 2024

Published: 21 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern digital computers can solve virtually any computational problem. However, to accomplish a computational task of arbitrary complexity, they may require impractically large resources such as time and memory. To overcome this challenge, unconventional [1,2] and neuromorphic [3–10] approaches to computer engineering and data processing were proposed based on the idea that a computer should imitate the operation of a biological brain [11,12].

While neuromorphic computers may not be as universal as the traditional digital ones, they can solve certain practically important problems with high accuracy. Neuromorphic computers are also inherently scalable, parallel, and allow for collocation of data processing and memory [9]. Similarly to a biological brain, they operate only when input data are available and also mimic the randomness of the firing of biological neurons, thus saving energy and decreasing the overall cost of computations [13,14].

These characteristics make the neuromorphic computers ideally suitable for applications in approximate computing, another emergent approach to computations that benefits the fields of machine learning, multimedia processing, signal processing, and scientific computing by replacing high-accuracy resource- and energy-consuming computations by alternative solutions that produce practicable results while reducing energy and resource consumption [15–19]. Both approximate computing and neuromorphic computing also utilise errors as an opportunity for enhancing efficiency, imitating the ability of a biological brain to learn and improve from errors [9].

Reservoir computing (RC) is a resource-efficient neuromorphic information processing algorithm that is especially suitable for making forecasts of highly nonlinear and chaotic time series that underpin a number of essential natural and human-made phenomena, including the variation of climate, dynamics of Earth population, trends in financial markets, energy generation, and drug discovery [12,20–27]. A typical RC algorithm [20,21] employs

a randomly initialised task-dependent neural network (reservoir) that is connected to the input units through random connections. The dynamics of the reservoir are advanced in time using a nonlinear update equation, resulting in a set of neural activation states. Then, output weights are computed as the linear regression weights of the teacher outputs on the activation states. The so-trained RC system is then tasked to either solve a classification problem or make forecasts using a new set of input data.

A computational reservoir can also be created using a physical, either experimental or theoretical, nonlinear dynamical system that effectively recreates the dynamical properties of an update equation of the traditional RC algorithm [20]. Called the physical reservoir [11], this approach to computer engineering has been successfully validated using spintronic devices [28–30], electronic circuits [31,32], photonic systems [33,34], mechanical devices [35], and liquids [12,36–38].

Yet, similarly to the advantage of quantum computers over classical digital computers [39], quantum physical RC systems—neuromorphic computers with a reservoir operating according to the laws of quantum mechanics [11,40–47]—offer a number of advantages over classical, both algorithmic and physical, RC systems. In particular, quantum physical RC systems have demonstrated a superior capability of predicting complex dynamical systems with many degrees of freedom [45,47], as well as the ability to create a large number of densely connected artificial neurons using technically simple coupled quantum oscillators instead of sophisticated physically coupled qubits [44]. Moreover, it has been demonstrated that while the quantum noise is undesirable in conventional quantum computations [39], it can be exploited as a computational reservoir [42].

In this paper, we propose and theoretically validate a novel quantum RC architecture that exploits the dynamics of an atom trapped in a cavity. One prominent feature of the proposed system is a coherent driving of the atom at a certain driving rate with the possibility to observe transitions between quantum states and effective “freezing” of the quantum evolution of the system. Frequent observations of the atom eigenstates prevent the system from undergoing significant changes, a phenomenon known as the Zeno effect [48]. Conversely, less frequent probing of the system states enables the system to undergo Rabi oscillations [49,50]. Judiciously using these properties, we optimise the rate at which the atom is driven (i.e., we optimise the measurement rate of atomic eigenstates) to adjust the quantum dynamics of the reservoir to undertake diverse classification and prediction tasks. This approach opens up opportunities for controlling and stabilising quantum states during a computation, benefiting such essential applications as decoherence mitigation [51], quantum information processing [52], quantum error correction, and quantum state stabilisation [50,53,54].

The remaining discussion is structured as follows. In Section 2, we theoretically describe the dynamics of an atom–cavity system subjected to coherent detection. In Section 3, we introduce the quantum reservoir model. Then, in Section 4, we test the reservoir on a number of challenging benchmarking tasks, including a classification task, chaotic time series free-running forecast task and physical system prediction task. We also highlight the role of the model parameters on the accuracy of forecasts made by the reservoir, including the effect of the driving amplitude, the number of neurons in the reservoir, the measurement rate of the atomic state, and the length of the training datasets. Finally, in line with the envisioned applications of the quantum reservoir in the field of approximate computing, we demonstrate that the implementation of the measurement-controlled dynamics in our model significantly decreases the computational resources required by the RC systems to successfully undertake complex prediction tasks.

2. Atom–Cavity Interaction under Measurement Control

We consider a physical system that consists of a two-level atom (qubit) that is trapped inside a cavity. The qubit is represented by a mode σ , but the cavity is represented by a mode a . In the framework of this simplification, the model is entirely isolated from any

external influences, allowing only for measurement-controlled interactions of the qubit with a coherent field input and accounting for an inherent decay of the cavity.

We assume that the atom has two possible spin states, $|0\rangle = |\uparrow\rangle$ and $|1\rangle = |\downarrow\rangle$, that interact with a quantised electromagnetic field within a cavity. The governing Hamiltonian \hat{H}_i for this atom–cavity interaction is expressed as

$$\hat{H}_i = ga^\dagger a \sigma_- \sigma_+, \tag{1}$$

where g represents the strength of the atom-cavity coupling, a is the cavity annihilation operator, and σ_- and σ_+ are the lowering and raising operators for the atom, respectively. The interaction between the atom and the cavity leads to exchange of energy, giving rise to Rabi oscillations—oscillations of energy between the atom and the cavity. Yet, multiple reflections of the probe field from the mirrors that form the cavity effectively enhance the strength of the interaction between the electromagnetic field and the atom.

The cavity is coherently driven by an external signal that is incident on one of the cavity mirrors. Such a forcing is modelled by the Hamiltonian

$$\hat{H}_c = -i\beta(a^\dagger - a), \tag{2}$$

where β is the amplitude of the driving signal. The driving signal not only controls the dynamics of the atom–cavity system but also serves as the carrier of input information that is encoded using variations in the signal’s time-domain waveform.

Importantly, the state of the atom undergoes continuous monitoring by means of a coherent measurement process, which enables real-time observations and subsequent control of the quantum state of the atom. In the model, these processes are taken into account by the Hamiltonian

$$\hat{H}_z = g_z(\sigma_+ + \sigma_-), \tag{3}$$

where g_z denotes the amplitude of the coherent atomic driving. Given that the atom collapses into an eigenstate depending on the frequency of measurement, the choice of g_z plays a pivotal role in controlling the dynamics of the system since it determines the frequency at which the state of the atom is measured. Hence, by tuning the value of g_z , we can effectively control the dynamics of the quantum system.

Figure 1 illustrates the time evolution of the operator $\langle \sigma_+ \sigma_- \rangle$ in Equation (1). Depending on the driving amplitude g_z , the system either exhibits a freezing behaviour similar to that observed in the Zeno effect with frequent measurements (depicted by the solid orange line) or it undergoes oscillations (represented by the dashed blue line).

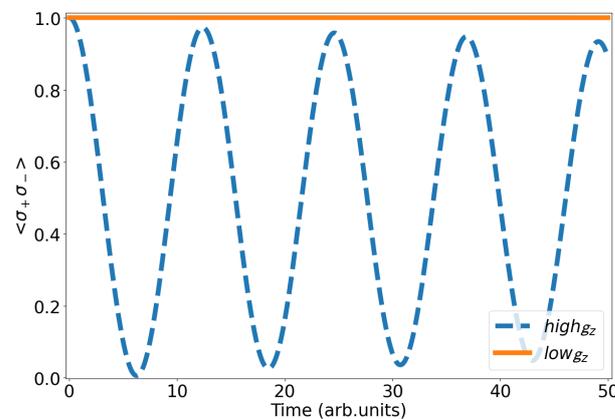


Figure 1. Time evolution of the operator $\langle \sigma_+ \sigma_- \rangle$ for high (infrequent measurement) and low (frequent measurement) values of g_z . Depending on the measurement rate, the system exhibits a Zeno effect with frequent measurements (the orange solid line) or it undergoes oscillations when the measurement rate is low (the blue dashed line).

In particular, while the atom–cavity interaction provides a natural platform for quantum manipulation, controlling the driving amplitude of the atom enables us to tune the evolution of the system. Since, in our model, the parameter g_z defines the measurement rate of the atomic states, in the following, we will employ the term “measurement rate” to denote the driving amplitude of the atom.

System Dynamics

We analyse the dynamics of the proposed RC system using a stochastic master equation approach where the system is represented as a qubit and a cavity mode [55]. The master equation considered by us corresponds to the time-dependent Schrödinger equation that accounts for the atom–light interaction and the effect of frequent measurements on the spin states of the atom. The full Hamiltonian for the system is

$$\hat{H} = \hat{H}_i + \hat{H}_c + \hat{H}_z, \tag{4}$$

where \hat{H}_i , \hat{H}_c and \hat{H}_z are given by Equations (1), (2), and (3), respectively. The time evolution of the density matrix ρ is governed by the linear stochastic master equation, which accounts for the effects of decoherence and dissipation.

$$\dot{\rho} = -i[\hat{H}, \rho] + \hat{C}\rho\hat{C}^\dagger - \frac{1}{2}\hat{C}^\dagger\hat{C}\rho - \frac{1}{2}\rho\hat{C}^\dagger\hat{C}, \tag{5}$$

where \hat{H} is given by Equation (4), $\hat{C} = \sqrt{\kappa}a$ is the collapse operator associated with the cavity decay, and κ is the decay rate. By numerically solving Equation (5), we simulate measurement-driven dynamics of the atom–cavity system.

The basis of the quantum states within the cavity $|n, \sigma\rangle$ is generated by the tensor product of an atom with the two possible spin states and the n -dimensional space of a quantised field ($|n, \sigma\rangle = |n\rangle \otimes |\sigma\rangle$). The population of the Fock states is driven by the coherent amplitude (β) and is further manipulated by means of the measurement rate g_z . The occupation probabilities of these states are given by

$$P(n, \sigma) = \langle n\sigma | \rho | n\sigma \rangle, \tag{6}$$

where ρ is the density matrix whose time derivative is given by Equation (5). To optimise the operation of the reservoir, the cavity driving amplitude β is adjusted to initially populate the first few Fock states with a significant probability. In Figure 2, the occupation probabilities $P(n, \sigma)$ for the first six Fock states are depicted as a function of β . These probabilities exhibit a nonlinear response and, for values of β within the range [10, 15], there is a nonzero occupation probability for all Fock bases $|n\sigma\rangle \in |00\rangle \dots |21\rangle$.

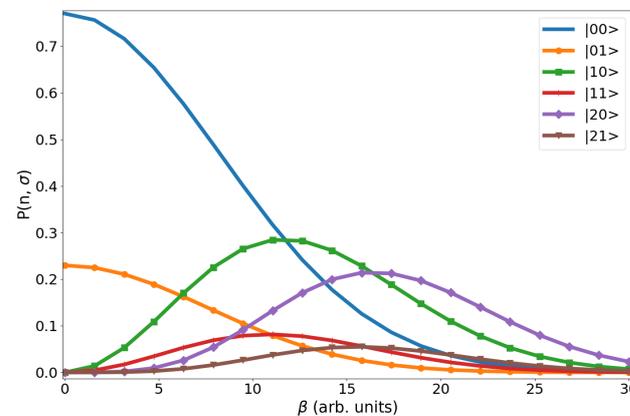


Figure 2. Occupation probabilities of Fock states $|n\sigma\rangle \in |00\rangle \dots |21\rangle$ as a function of cavity driving amplitude β .

3. Quantum Reservoir Model

The principle of operation of the quantum reservoir is illustrated in Figure 3. The neurons of the reservoir are defined by Fock states $|n, \sigma\rangle$ that form the basis for the quantum states within the cavity and constitute a computational space of $2n$ states. The input data points are converted into signals that modulate the driving amplitude $\beta(t)$ used to change the distribution of the Fock states population (effectively, as explained above, the reservoir nonlinearly transforms the input data into a high-dimensional state space). The input information is carried by an electromagnetic wave delivered through one mirror of the cavity, but the output signal is detected at the opposite cavity side. From the computational point of view, the output is determined by the expectation values $P(n, \sigma)$ of the occupancy of the basis states. These occupation probabilities are subjected to classification through a trainable fully connected layer (illustrated by the dashed black arrows in Figure 3) using a regression algorithm.

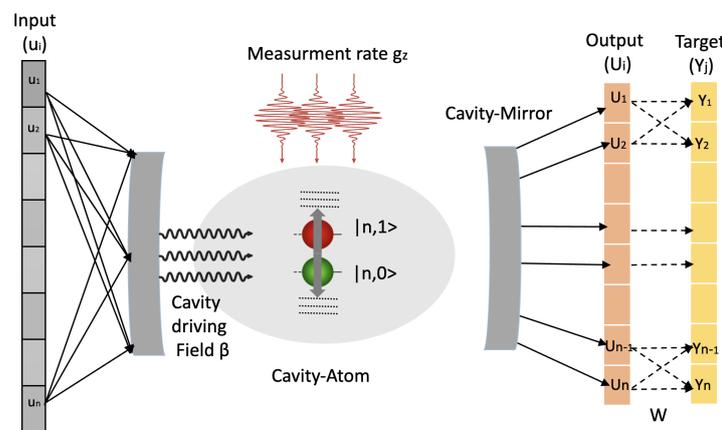


Figure 3. Sketch of the RC system with measurement-controlled quantum dynamics. The pivotal component of the reservoir is a cavity–atom system with continuously monitored states. The neural activations of the reservoir are given by the Fock states of the atom–cavity system. The reservoir is coherently driven using a signal defined by the discrete points of the input dataset. The classified readouts of the reservoir are processed by means of a linear regression technique.

Moreover, we tune the measurement rate to optimise the dynamics of the reservoir. That is, in our system, the measurement rate effectively corresponds to the leaking rate parameter that controls the dynamics of the traditional RC algorithm and that needs to be adjusted for every specific problem [20,21]. For instance, when the quantum reservoir is tasked with a problem that involves an extended plateau regime, its dynamics can be slowed down (“frozen”) using a suitable value of g_z . Conversely, the dynamics of the reservoir can be accelerated when the input of the RC system is defined by a rapidly evolving input time series.

The reservoir is trained using the input dataset $u = \{u_1, u_2, \dots, u_M\}$, with each data point corresponding to a discrete instant of time t_M . Additionally, each point of the input dataset is associated with a training dataset \tilde{Y}_{train} . As part of an iterative procedure, the input values u_i with $i = 1 \dots M$ are encoded as the driving amplitude β of the signal used to drive the dynamics of the reservoir (this computation corresponds to the solution of Equation (5) with the Hamiltonian given by Equation (4)). The values of the output layer U are obtained by measuring a set of observables $P(n, \sigma)$. A linear combination of these observables is then optimised to generate the target associated with each computational task.

Thus, the output of the reservoir can be represented as

$$\tilde{Y} = \mathbf{WF}(\mathbf{U}_{\text{train}}), \tag{7}$$

where $\mathbf{U}_{\text{train}}$ is the training dataset and \mathbf{F} is the function that encodes the transformation of the input into the outcome of measuring the states of the reservoir neurons. The weight

matrix \mathbf{W} is then optimised through a training process to align the neural network output $\tilde{\mathbf{Y}}$ with the target vector $\mathbf{Y}_{\text{train}}$. The weight matrix \mathbf{W} can be calculated as

$$\mathbf{W} = \tilde{\mathbf{Y}}_{\text{train}} \mathbf{F}^{\dagger}(\mathbf{U}_{\text{train}}), \quad (8)$$

where \mathbf{F}^{\dagger} is the Moore–Penrose pseudo inverse of the function \mathbf{F} . Then, at the stage of exploitation of the reservoir, the optimised weight matrix is applied to a test dataset \mathbf{U}_{test} as follows:

$$\mathbf{Y}_{\text{test}} = \mathbf{W}\mathbf{F}(\mathbf{U}_{\text{test}}). \quad (9)$$

In an idealised scenario, the prediction made by a trained reservoir should coincide with the target dataset (also called the ground truth) to graphical accuracy. However, in practice, the output of the RC system deviates from the target. In the literature on reservoir computing, the forecast-target deviation is often quantified using the root-mean-square error (RMSE) that can be calculated as

$$\text{RMSE} = \frac{1}{y_{\text{max}} - y_{\text{min}}} \sqrt{\frac{\sum_i^N (y_i - \tilde{y}_i)^2}{N}}, \quad (10)$$

where N is the total number of data points taken into account in the calculation of RMSE, y_i is the actual target value for the i th data point, and \tilde{y}_i is the value predicted by the RC system. The range of the target values is accounted for by the values of y_{max} and y_{min} . In addition, we evaluate the performance of the reservoir using the figure-of-merit called Accuracy, which is calculated as the ratio of the correct predictions to the total number of predictions made by the RC system:

$$\text{Accuracy} = \frac{\sum_i^{N_{\text{eqv}}} |y_i - \tilde{y}_i| < \epsilon}{N_{\text{eqv}}} \times 100, \quad (11)$$

The agreement percentage is calculated such that the difference between the correct predictions and the target is less than $\epsilon = 10^{-2}$.

4. Results and Discussion

4.1. Task Classification

To evaluate the accuracy of the predictions made by our RC system, we task it with solving a series of test problems. The first problem, which was used to assess the performance of several previously developed quantum RC systems [44,56], involves a binary categorisation of a synthetic waveform composed of randomly generated sinusoidal and square pulses (Figure 4a). In this task, the output of the reservoir is expected to be 0 (1) when an input point corresponds to the square (sinusoidal) portion of the test waveform. This task is designed to test the memory capacity and nonlinearity of the reservoir [44].

The performance of the RC system configured to have 8 and 16 neurons is evaluated in Figure 4b,c, respectively, where the solid green line denotes the target but the dashed red line corresponds to the output of the reservoir. We note that the target signal is plotted purely for the comparison of the reservoir output with the ground truth, i.e., the reservoir is not presented with the target data at the exploitation stage. We can see that the ability of the reservoir to classify the input improves as the number of neurons is increased. The reservoir with 16 neurons can complete the test task with Accuracy = 99.7%.

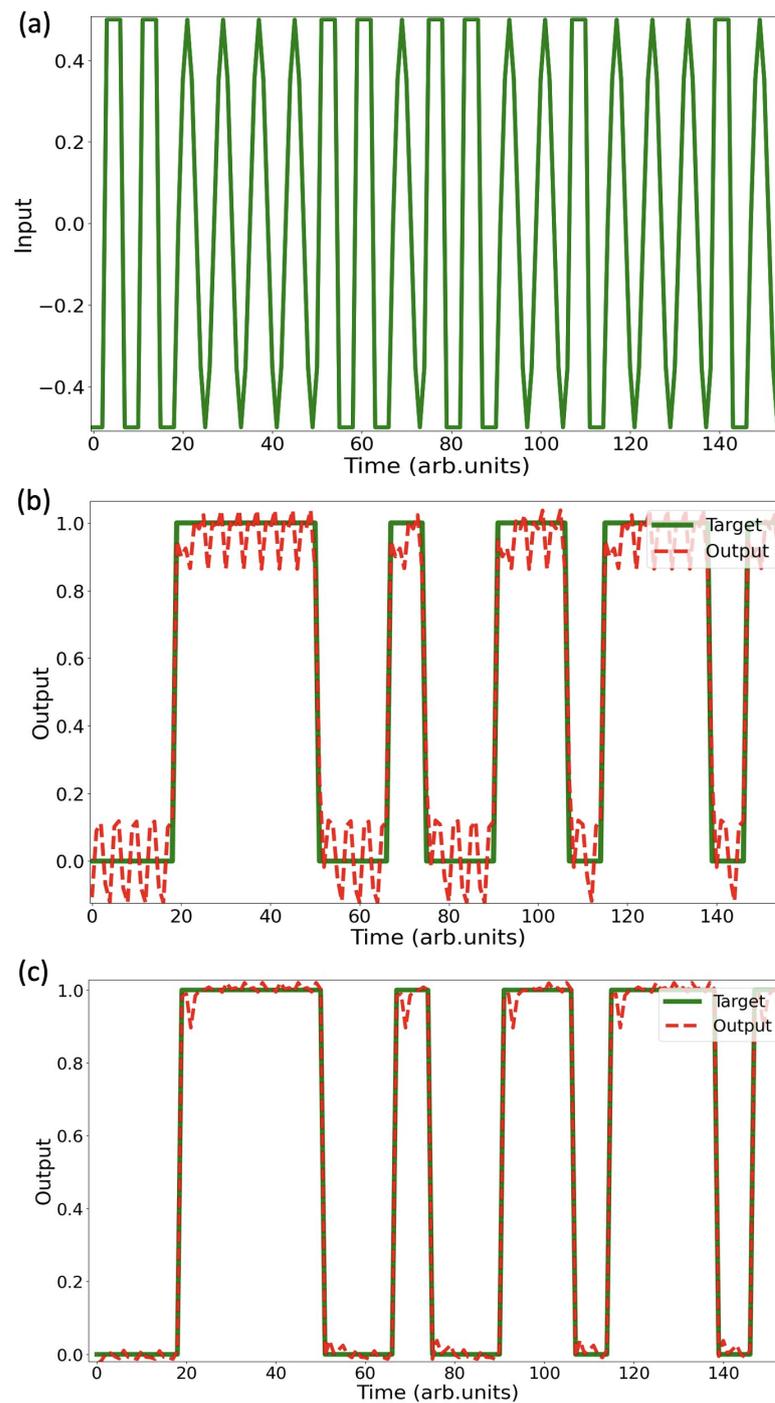


Figure 4. (a) Input data generated from a random array representing either a sinusoidal or square waveform. Results of the classification of the sinusoidal and square waveform by the reservoir with (b) 8 and (c) 16 neurons. The target is shown in solid green line and the reservoir prediction is denoted by the dashed red line.

To gain a deeper insight into the dependence of the reservoir performance on the number of neurons, Figure 5 plots the RMSE and Accuracy as a function of the number of neurons employed in the classification task. It is noteworthy that the reservoir with just 8 neurons demonstrates Accuracy $\approx 90\%$ and RMSE ≈ 0.1 . Yet, increasing the number of neurons to 16 results in a significant improvement, yielding Accuracy $\approx 99.7\%$ and RMSE $\approx 4 \times 10^{-3}$. Importantly, any further increase in the number of neurons beyond 16 results in just a marginal increase in these figures-of-merit, indicating that 16 neurons

is an optimal configuration that enables the reservoir to make feasible forecasts while minimising the resource consumption.

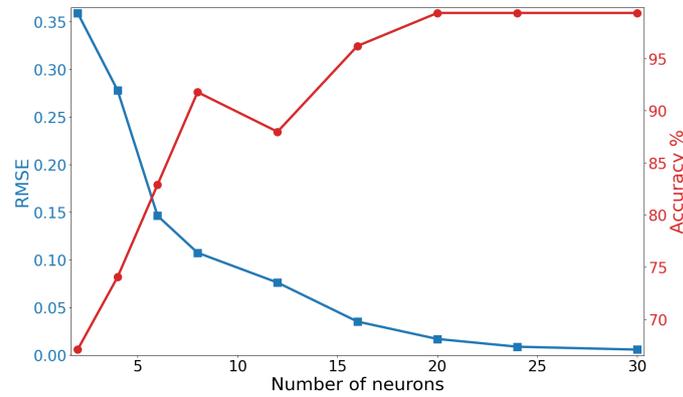


Figure 5. The root-mean-square error (RMSE) (the left y -axis and blue square markers) and Accuracy (the right y -axis and red circular markers) obtained for the sinusoidal-square waveform classification task as a function of the number of neurons in the reservoir.

We also used the quantum RC software that accompanies the discussion in Ref. [44] to undertake the same task as in Figure 4. To enable the correct benchmarking, the computational code from Ref. [44] was run on the same workstation computer equipped with the same version of Python programming language used to implement our reservoir model. We revealed that, for the same length of the training dataset, our reservoir accomplished the task in 9 s, compared with 50 s required by the reservoir developed in Ref. [44].

Figure 6 plots the value of RMSE as a function of the measurement rate. It reveals that an optimal performance of the reservoir can be achieved at $g_z = 5$ arb. units. To obtain a bigger picture, we computed RMSE for the larger values of the measurement rate, demonstrating that a higher rate does not result in better performance.

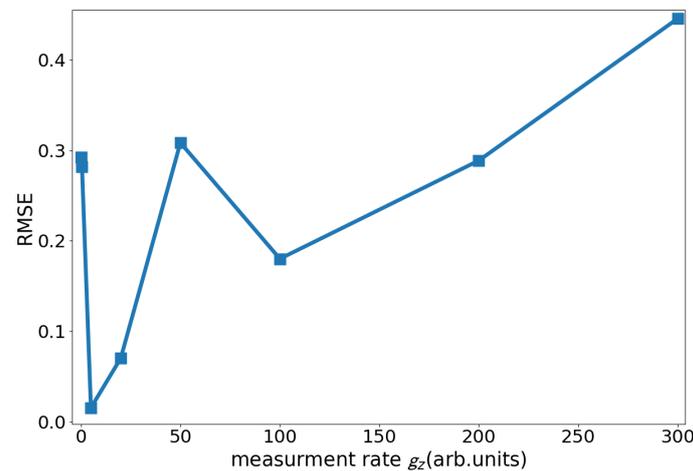


Figure 6. Performance of the reservoir as the function of the measurement rate. The optimal RMSE is achieved at the measurement rate of 5 arb. units.

4.2. Chaotic Times-Series Forecasting

The second test task used to evaluate the performance of our RC system consists of predicting a Mackay–Glass time series (MGTS) that is generated solving the delay differential equation [57]

$$\dot{x}_{MGTS}(t) = \beta_{MGTS} \frac{x_{MGTS}(t - \tau_{MGTS})}{1 + x_{MGTS}^q(t - \tau_{MGTS})} - \gamma_{MGTS} x_{MGTS}(t), \quad (12)$$

where overdot denotes differentiation with respect to time and $\tau_{MGTs} = 17$, $q = 10$, $\beta_{MGTs} = 0.2$, and $\gamma_{MGTs} = 0.1$ [21]. The generated time series is then split into two parts used at the training and testing stages, respectively.

In this test task, the reservoir operates in the generative mode, also known as the free-running forecast, where the output produced by a trained reservoir in the previous time step serves as an input at the next time step, i.e., $u_{n+1} = y_n$ [21] (in other words, the reservoir acts as a self-generator [58]). We stress that we deliberately choose to test the reservoir in the generative mode because, as shown in Refs. [12,19–21,59,60], demonstrations of the operation in the predictive mode are technically straightforward. While the operation in the generative mode is a more challenging task, the practical importance of generative reservoirs is typically much higher since they can be used to solve a wide range of problems concerned with the prediction of difficult-to-analyse processes such as the behaviour of financial markets and variations in climate [12].

Figure 7 demonstrates the result of the free-running forecast made by the quantum reservoir with 16 neurons. The forecast future evolution of MGTs is denoted by the red dashed line and it starts at the instant of time of 1000 arb. units. The green solid line corresponds to the ground truth. We can see that the RC system correctly reproduces the general pattern of MGTs, though it misses some minor features of the ground truth time series.

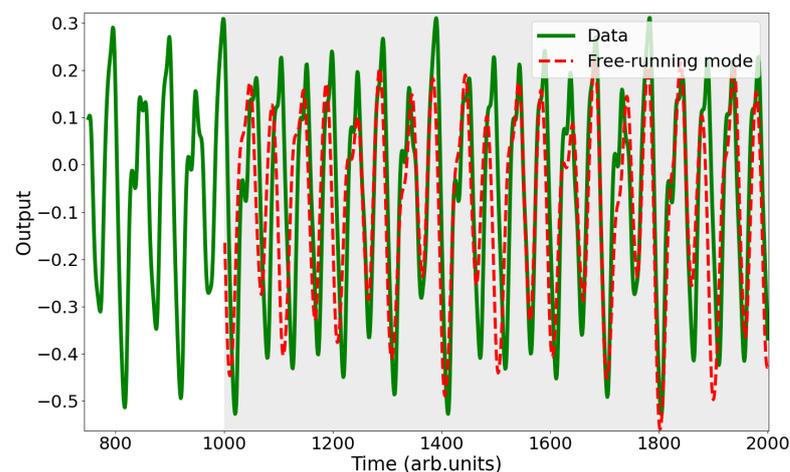


Figure 7. Generative mode operation exemplified by a free-running forecast of MGTs. In this figure, we compare the output of the reservoir (the dashed red line) with the target MGTs (the solid green line). The reservoir has 16 neurons and it was trained on several cycles of MGTs variations. Note that the reservoir was not presented with the ground truth MGTs data to make the forecast. The comparison with the ground truth is needed only to evaluate the accuracy of the forecast.

In Figure 8, we compare the forecast made by the quantum reservoir (the dashed line) with the prediction made by the traditional reservoir (the dashed-dotted line) described in Ref. [21]. Both reservoirs were trained on and tested against the same MGTs dataset (the solid line). It is noteworthy that the training portion of the MGTs dataset is 300 points long, which is the shortest dataset that can be used to train the traditional reservoir (i.e., below this threshold, the traditional reservoir fails to produce any meaningful output; however, as shown in Figure 9, the quantum reservoir can be trained using shorter datasets). For consistency with the preceding discussion, the quantum reservoir was configured to have 16 neurons. However, in the traditional reservoir, we had to use at least 35 neurons to produce a practicable result (i.e., the computer code of the traditional reservoir stops functioning properly if the number of neurons is less than 35). The other hyperparameters of the traditional RC system used in this comparison were the leaking rate $\alpha = 0.3$ and the ridge parameter $\lambda = 10^{-6}$. The spectral radius was computed according to the procedure outlined in Refs. [20,21].

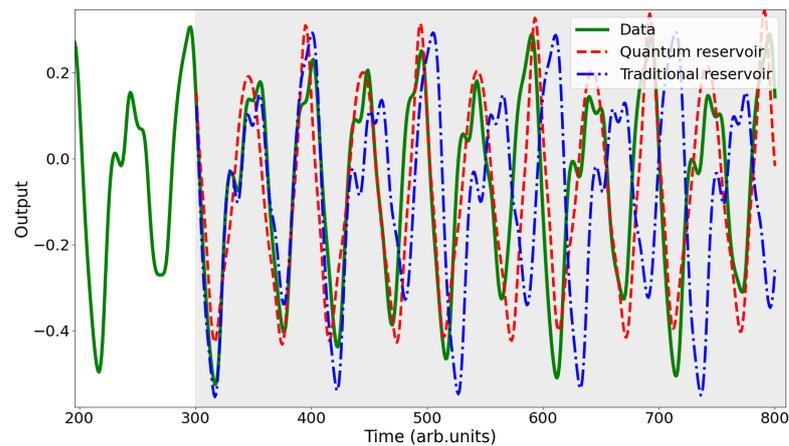


Figure 8. MGTS forecast of the quantum reservoir (the dashed line) compared with the forecast made by the traditional reservoir (the dashed-dotted line). The target MGTS is denoted by the solid line. Note that the quantum reservoir has 16 neurons but the traditional reservoir has 35 neurons, as discussed in the main text. Also, note the better accuracy of the long-term forecast made by the quantum reservoir.

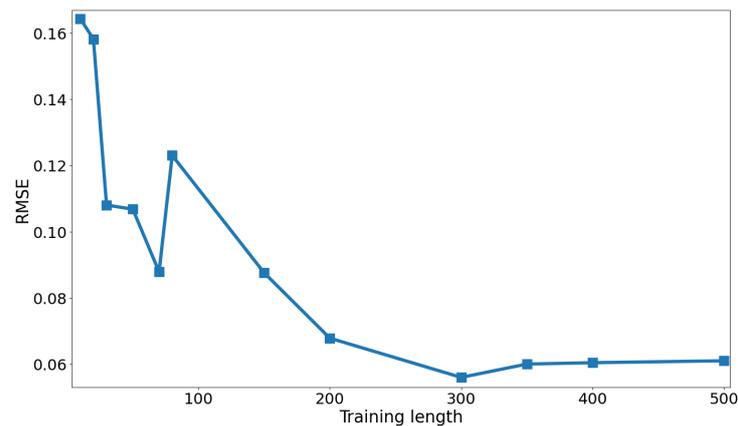


Figure 9. RMSE plotted as a function of the length of the training MGTS dataset.

We reveal that the quantum reservoir tends to produce the MGTS with a slightly lower amplitude. However, this result should not be considered as an artefact since the same behaviour was observed in the normal operation of the previous theoretical [19] and experimental [60] works on physical RC systems. Hence, if required, the output of the quantum reservoir can be rescaled to achieve a better agreement with the ground truth. Most importantly, while we can see that during the first three pseudo-periods of MGTS the traditional reservoir outperforms the quantum one, it produces less accurate results in the long term. Indeed, we calculate RMSE, taking into account the 500 testing data points, and we observe $\text{RMSE} = 1.4 \times 10^{-2}$ for the quantum reservoir and $\text{RMSE} = 1.0 \times 10^{-1}$ for the traditional one.

In a previous work [60], we also demonstrated that the correct operation of a traditional RC algorithm is possible mostly using relatively long training datasets. In fact, the traditional RC system implemented following the procedure outlined in Refs. [20,21] requires approximately 1000 training MGTS data points. In Figure 9, we plot RMSE as a function of the length of the MGTS dataset used to train our RC system. We can see that a reasonable accuracy can be reached with a minimum of 300 training data points. Any further increase in the length of the training dataset does not result in a significant increase in the accuracy. This results mirrors our previous observation made in Ref. [60]: an RC system based on the physical principles can be trained using shorter datasets compared with the dataset length required to train a traditional RC system.

4.3. Damped Harmonic Oscillator Prediction

In this section, we demonstrate the ability of the proposed RC system to learn and predict real-life physical phenomena. As a test task, the RC system is presented with several periods of oscillation of a harmonic oscillator, a system that, when displaced from its equilibrium position, experiences a restoring force that is proportional to the displacement. In the presence of damping, depending on the friction coefficient, the system oscillates with a frequency that is lower than that in the undamped case, but the amplitude of oscillations decreases with time.

Despite its relative simplicity, this test task is nontrivial since it requires any RC system to adjust to constantly changing input conditions, also requiring strong output feedback needed to generate oscillations (for this reason, a conceptually similar test problem called frequency generator was used to demonstrate the operation of the traditional RC algorithm by its creators [12,61]). Moreover, undertaking this specific task presents a considerable challenge to the reservoir due to a two-parameter space nature of the problem: the RC system should be trained to accurately simulate the dynamics of both frequency and amplitude. In Figure 10, we plot the forecast made by the reservoir trained on a damped oscillator dataset. We observe a good agreement with the ground truth over the first few periods of oscillation. Then, we can see certain deviations of the free-running forecast from the natural dynamics of the oscillator. Despite this artefact, in the following section, we demonstrate that the ability of the RC system to predict a harmonic oscillator can find practical applications in the framework of the paradigm of approximate computing.

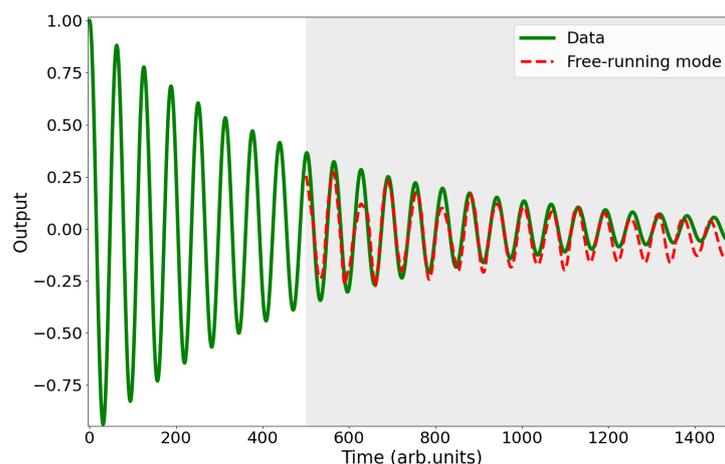


Figure 10. Output of the RC system trained to predict a damped harmonic oscillator. The free-running forecast made by the the RC system is denoted by the dashed red line. The solid green line denotes the ground truth.

5. Discussion

We demonstrated that the proposed quantum RC system can successfully undertake several challenging test tasks using a very small number of neurons compared with the traditional RC systems. Of course, software that implements the traditional RC algorithm can make a more accurate forecast using several thousands of neurons (see Ref. [21] and the computer code that accompanies it). However, such a computation will require a high-performance workstation computer that will need to be run for several hours or even days to find the optimal set of hyperparameters. We also established that our quantum RC system outperforms the other quantum-physical reservoirs in terms of computational resources needed to solve the standard test problems.

These characteristics make our RC system suitable for applications in approximate computing. Yet, our findings can also be used in embedded systems that typically have limited processing power and memory compared to general-purpose computers, often requiring low power consumption.

For example, while, admittedly, the forecast made by the RC system in Figure 10 is not ideal when compared with the ground truth, the demonstrated operation of the reservoir holds the potential to significantly facilitate certain computational and experimental research procedures, thereby fitting into the paradigm of approximate computing. Indeed, a quantum RC system trained using a damped harmonic oscillator may become a valuable tool for the investigation of individual quantum emitters that serve as a fundamental building block of many emerging devices [62,63]. From the computational point of view, the design of such devices requires the calculation of quality factors and decay rates of photonic resonators integrated with waveguiding structures, which is typically accomplished using complex numerical techniques, including the finite-difference time-domain (FDTD) method [64], or sophisticated analytical models [65].

The FDTD method is popular in the field of photonics since it enables one to accurately represent the geometry of optical resonators and waveguides while taking into account the optical properties of materials such as dispersion, nonlinearity, and absorption in a wide range of optical frequencies of interest. However, due to the time-domain nature of its algorithm, calculations of quality factors and decay rates of realistic photonic structures require one to run FDTD algorithm for a long time to reach a steady-state regime.

While several approaches intended to decrease the computational effort associated with the achievement of the steady-state regime have been proposed [66], a typical FDTD simulation requires at least several hours of CPU time of a high-performance workstation computer. Our RC system can be used to further decrease the calculation time: it can be trained on data obtained after a relatively short FDTD simulation (typically, it requires approximately 30 min to generate such data) and then used to predict the further temporal evolution. Since such a hybrid calculation will take about one hour in total, the user can save from two to eight hours of CPU time per simulation. Given that a typical computational research project involves several tens of independent simulation runs, the application of the RC system can potentially save up to 800 CPU hours. Of course, using this combined approach, one needs to take into account the imperfection of forecasts made by the RC system. On the other hand, even well-designed FDTD simulations are not free of numerical artefacts that can be of the same order of magnitude as the imperfection of the forecast. Yet, in some simulations, the FDTD method can suffer from late time instabilities that may not allow the user to run the software until the steady-state regime is reached [67]. This problem can also be resolved using the RC system.

Researchers conducting experimental work encounter similar problems during the analysis of raw data. For example, this is the case for research on nanodiamonds containing fluorescent nitrogen-vacancy (NV) centres that are essential for biomedical imaging and sensing [68]. The rates of radiative and nonradiative decay of the excited NV centre states are affected by surface proximity effects [69]. The analysis of these decay rates shapes the research in the field, relying on different models that produce the best fits of experimental data [68,69]. In many cases, the fitting procedure needs to be conducted manually. The RC system can be used to optimise the data fitting procedure, helping researchers predict the decay rates using noisy or incomplete experimental datasets. A similar RC-based fitting procedure can also be used to investigate the oscillation of many other physical systems, including magnetic gas sensors [70] and nonlinearly oscillating gas bubbles in liquids [71].

In many practical applications, including the aforementioned areas of biomedical imaging and sensing, data processing needs to be conducted using portable and miniaturised systems. Usually, such systems have limited computational resources and are designed to consume low power. The quantum reservoir proposed in this paper can be used to optimise the performance of such systems. The large computational power is required to perform operations with matrices whose size depends on the number of neurons in the reservoir. Since our quantum RC system uses a small number of neurons, the matrices associated with its algorithm are also small. In our recent work [60], we demonstrated that software that implements an RC system that requires small matrices can be run on inexpensive microcontrollers. We also showed that microcontroller-based RC systems can

be integrated with various sensors and actuators, thereby enabling one to implement an approximate computing scheme in a field experiment and conduct calculations on board of an unmanned aerial vehicle (UAV) [72]. An on-board RC system can also help an UAV to recognise other drones [73].

6. Future Work

Apart from the aforementioned prospective applications of the proposed RC system, there exist several potential research directions that warrant a special discussion due to their potential to advance the physical and technical aspects of quantum reservoir computing.

Firstly, quantum mechanics is rich in intriguing physical effects that await exploration in the context of computational science and machine learning. For example, while in this paper we theoretically proposed a particular physical model that can be used to implement a measurement-controlled reservoir, we point out the existence of other physical systems and processes that can be used to create a similar computational reservoir operating in a wide range of frequencies, spanning from microwaves to the optical regime [74]. Broadly speaking, the ideas expressed in this paper will be of interest to researchers working at an interface of electronics [32], magnetism-based RC systems [29,75], and quantum-mechanical systems that exhibit complex dynamical behaviour [76–78].

Secondly, software based on the proposed quantum RC system can be employed in ultra-low-power artificial intelligence and machine learning systems [79]. We already successfully tested our software on several mobile systems, including Raspberry Pi, and we plan to further optimise it for operation on the Arduino platform [60].

Finally, we anticipate the application of quantum RC systems in research projects that aim to validate the quantum mind theory that suggests that human behaviour can be modelled using the laws of quantum mechanics [80–83]. For example, artificial deep neural networks built using some principles of quantum physics have been employed to model the human perception of optical illusions (see Refs. [84–86] and references therein). In particular, in Ref. [86], the deep neural network describes the human perception using a synthetic quasi-random time-dependent signal. Subsequently, since the RC algorithm is especially suitable for forecasting complex time series [20], the quantum reservoir can be used to project the output of the deep neural network into the future instances of time, thus dramatically reducing the computational resources needed to train and exploit the more computationally demanding deep neural network.

7. Conclusions

We proposed and theoretically validated a computational reservoir system that operates using the dynamics of a probed atom in a cavity and relies on the control of the quantum measurement rate. Benchmarking the performance of the quantum reservoir using several challenging test problems, we demonstrated that feasible forecasts can be made using just a few quantum neurons compared with at least several tens of traditional artificial neurons required for the operation of a traditional reservoir computing system. We also showed that our quantum reservoir produces accurate results even when it is trained on relatively short training datasets.

While the performance of the traditional reservoir can be improved, until a certain point, by increasing the number of neurons and optimising the set of classical reservoir hyperparameters, including the leaking rate and spectral radius, this procedure will require the use of an expensive and difficult-to-access high-performance computer. Subsequently, given that the quantum reservoir can produce usable results with a small number of neurons, it is plausible that it may be used to solve many practical problems in the framework of the paradigm of approximate computing.

Yet, compared with the other quantum and classical reservoir systems, software that implements our reservoir can be run on ordinary desktop and laptop computers equipped with modest computational resources. This quality makes it possible to utilise our reservoir

in embedded and portable systems, including various type of sensors and actuators that can be integrated with autonomous vehicles and robots.

Author Contributions: Conceptualization, writing, editing, and discussion: A.H.A. and I.S.M.; methodology, software, and validation: A.H.A.; supervision: I.S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The code used in this study is available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RC	Reservoir Computing
CPU	Central processing unit
FDTD	Finite-difference time-domain
MGTS	Mackay–Glass time series
NV	Nitrogen-vacancy
RC	Reservoir computing
RMSE	Root-mean-square error
UAV	Unmanned aerial vehicle

References

1. Adamatzky, A. (Ed.) *Advances in Unconventional Computing. Volume 2: Prototypes, Models and Algorithms*; Springer: Berlin, Germany, 2017.
2. Adamatzky, A. A brief history of liquid computers. *Philos. Trans. R. Soc. B* **2019**, *374*, 20180372. [[CrossRef](#)]
3. Shastri, B.J.; Tait, A.N.; de Lima, T.F.; Pernice, W.H.P.; Bhaskaran, H.; Wright, C.D.; Prucnal, P.R. Photonics for artificial intelligence and neuromorphic computing. *Nat. Photon.* **2020**, *15*, 102–114. [[CrossRef](#)]
4. Marcucci, G.; Pierangeli, D.; Conti, C. Theory of neuromorphic computing by waves: Machine learning by rogue waves, dispersive shocks, and solitons. *Phys. Rev. Lett.* **2020**, *125*, 093901. [[CrossRef](#)] [[PubMed](#)]
5. Marković, D.; Mizrahi, A.; Querlioz, D.; Grollier, J. Physics for neuromorphic computing. *Nat. Rev. Phys.* **2020**, *2*, 499–510. [[CrossRef](#)]
6. Suárez, L.E.; Richards, B.A.; Lajoie, G.; Misic, B. Learning function from structure in neuromorphic networks. *Nat. Mach. Intell.* **2021**, *3*, 771–786. [[CrossRef](#)]
7. Rao, A.; Plank, P.; Wild, A.; Maass, W. A long short-term memory for AI applications in spike-based neuromorphic hardware. *Nat. Mach. Intell.* **2022**, *4*, 467–479. [[CrossRef](#)]
8. Sarkar, T.; Lieberth, K.; Pavlou, A.; Frank, T.; Mailaender, V.; McCulloch, I.; Blom, P.W.M.; Torricelli, F.; Gkoupidenis, P. An organic artificial spiking neuron for in situ neuromorphic sensing and biointerfacing. *Nat. Electron.* **2022**, *5*, 774–783. [[CrossRef](#)]
9. Schuman, C.D.; Kulkarni, S.R.; Parsa, M.; Mitchell, J.P.; Date, P.; Kay, B. Opportunities for neuromorphic computing algorithms and applications. *Nat. Comput. Sci.* **2022**, *2*, 10–19. [[CrossRef](#)] [[PubMed](#)]
10. Krauhausen, I.; Coen, C.T.; Spolaor, S.; Gkoupidenis, P.; van de Burgt, Y. Brain-inspired organic electronics: Merging neuromorphic computing and bioelectronics using conductive polymers. *Adv. Funct. Mater.* **2023**, 2307729. [[CrossRef](#)]
11. Nakajima, K.; Fisher, I. *Reservoir Computing*; Springer: Berlin/Heidelberg, Germany, 2021.
12. Maksymov, I.S. Analogue and physical reservoir computing using water waves: Applications in power engineering and beyond. *Energies* **2023**, *16*, 5366. [[CrossRef](#)]
13. Maass, W.; Natschläger, T.; Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **2002**, *14*, 2531–2560. [[CrossRef](#)]
14. Jaeger, H.; Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **2004**, *304*, 78–80. [[CrossRef](#)]
15. Mittal, S. A Survey of Techniques for Approximate Computing. *ACM Comput. Surv.* **2016**, *48*, 1–33. [[CrossRef](#)]
16. Liu, W.; Lombardi, F.; Schulte, M. Approximate Computing: From Circuits to Applications. *Proc. IEEE* **2020**, *108*, 2103–2107. [[CrossRef](#)]
17. Henkel, J.; Li, H.; Raghunathan, A.; Tahoori, M.B.; Venkataramani, S.; Yang, X.; Zervakis, G. Approximate Computing and the Efficient Machine Learning Expedition. In Proceedings of the 2022 IEEE/ACM International Conference on Computer Aided Design (ICCAD), San Diego, CA, USA, 30 October–3 November 2022; pp. 1–9.

18. Ullah, S.; Kumar, A., Introduction. In *Approximate Arithmetic Circuit Architectures for FPGA-Based Systems*; Springer International Publishing: Cham, Germany, 2023; pp. 1–26.
19. Maksymov, I.S.; Pototsky, A.; Suslov, S.A. Neural echo state network using oscillations of gas bubbles in water. *Phys. Rev. E* **2021**, *105*, 044206. [[CrossRef](#)] [[PubMed](#)]
20. Lukoševičius, M.; Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **2009**, *3*, 127–149. [[CrossRef](#)]
21. Lukoševičius, M. A Practical Guide to Applying Echo State Networks. In *Neural Networks: Tricks of the Trade, Reloaded*; Montavon, G., Orr, G.B., Müller, K.R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 659–686.
22. Bala, A.; Ismail, I.; Ibrahim, R.; Sait, S.M. Applications of metaheuristics in reservoir computing techniques: A Review. *IEEE Access* **2018**, *6*, 58012–58029. [[CrossRef](#)]
23. Tanaka, G.; Yamane, T.; Héroux, J.B.; Nakane, R.; Kanazawa, N.; Takeda, S.; Numata, H.; Nakano, D.; Hirose, A. Recent advances in physical reservoir computing: A review. *Neural Newt.* **2019**, *115*, 100–123. [[CrossRef](#)] [[PubMed](#)]
24. Nakajima, K. Physical reservoir computing—An introductory perspective. *Jpn. J. Appl. Phys.* **2020**, *59*, 060501. [[CrossRef](#)]
25. Cucchi, M.; Abreu, S.; Ciccone, G.; Brunner, D.; Kleemann, H. Hands-on reservoir computing: A tutorial for practical implementation. *Neuromorph. Comput. Eng.* **2022**, *2*, 032002. [[CrossRef](#)]
26. Damicelli, F.; Hilgetag, C.C.; Goulas, A. Brain connectivity meets reservoir computing. *PLoS Comput. Biol.* **2022**, *18*, e1010639. [[CrossRef](#)] [[PubMed](#)]
27. Zhang, H.; Vargas, D.V. A survey on reservoir computing and its interdisciplinary applications beyond traditional machine learning. *IEEE Access* **2023**, *11*, 81033–81070. [[CrossRef](#)]
28. Riou, M.; Torrejon, J.; Garitainė, B.; Araujo, F.A.; Bortolotti, P.; Cros, V.; Tsunegi, S.; Yakushiji, K.; Fukushima, A.; Kubota, H.; et al. Temporal pattern recognition with delayed-feedback spin-torque nano-oscillators. *Phys. Rep. Appl.* **2019**, *12*, 024049. [[CrossRef](#)]
29. Watt, S.; Kostylev, M. Reservoir computing using a spin-wave delay-line active-ring resonator based on yttrium-iron-garnet film. *Phys. Rev. Appl.* **2020**, *13*, 034057. [[CrossRef](#)]
30. Allwood, D.A.; Ellis, M.O.A.; Griffin, D.; Hayward, T.J.; Manneschi, L.; Musameh, M.F.K.; O’Keefe, S.; Stepney, S.; Swindells, C.; Trefzer, M.A.; et al. A perspective on physical reservoir computing with nanomagnetic devices. *Appl. Phys. Lett.* **2023**, *122*, 040501. [[CrossRef](#)]
31. Cao, J.; Zhang, X.; Cheng, H.; Qiu, J.; Liu, X.; Wang, M.; Liu, Q. Emerging dynamic memristors for neuromorphic reservoir computing. *Nanoscale* **2022**, *14*, 289–298. [[CrossRef](#)]
32. Liang, X.; Tang, J.; Zhong, Y.; Gao, B.; Qian, H.; Wu, H. Physical reservoir computing with emerging electronics. *Nat. Electron.* **2024**. [[CrossRef](#)]
33. Sorokina, M. Multidimensional fiber echo state network analogue. *J. Phys. Photonics* **2020**, *2*, 044006. [[CrossRef](#)]
34. Rafayelyan, M.; Dong, J.; Tan, Y.; Krzakala, F.; Gigan, S. Large-scale optical reservoir computing for spatiotemporal chaotic systems prediction. *Phys. Rev. X* **2020**, *10*, 041037. [[CrossRef](#)]
35. Coulombe, J.C.; York, M.C.A.; Sylvestre, J. Computing with networks of nonlinear mechanical oscillators. *PLoS ONE* **2017**, *12*, e0178663. [[CrossRef](#)]
36. Kheirabadi, N.R.; Chiolerio, A.; Szaciłowski, K.; Adamatzky, A. Neuromorphic liquids, colloids, and gels: A review. *ChemPhysChem* **2023**, *24*, e202200390. [[CrossRef](#)]
37. Gao, C.; Gaur, P.; Rubin, S.; Fainman, Y. Thin liquid film as an optical nonlinear-nonlocal medium and memory element in integrated optofluidic reservoir computer. *Adv. Photonics* **2022**, *4*, 046005. [[CrossRef](#)]
38. Marcucci, G.; Caramazza, P.; Shrivastava, S. A new paradigm of reservoir computing exploiting hydrodynamics. *Phys. Fluids* **2023**, *35*, 071703. [[CrossRef](#)]
39. Nielsen, M.; Chuang, I. *Quantum Computation and Quantum Information*; Oxford University Press: New York, NY, USA, 2002.
40. Mujal, P.; Martínez-Peña, R.; Nokkala, J.; García-Beni, J.; Giorgi, G.L.; Soriano, M.C.; Zambrini, R. Opportunities in quantum reservoir computing and extreme learning machines. *Adv. Quantum Technol.* **2021**, *4*, 2100027. [[CrossRef](#)]
41. Govia, L.C.G.; Ribeill, G.J.; Rowlands, G.E.; Krovi, H.K.; Ohki, T.A. Quantum reservoir computing with a single nonlinear oscillator. *Phys. Rev. Res.* **2021**, *3*, 013077. [[CrossRef](#)]
42. Suzuki, Y.; Gao, Q.; Pradel, K.C.; Yasuoka, K.; Yamamoto, N. Natural quantum reservoir computing for temporal information processing. *Sci. Rep.* **2022**, *12*, 1353. [[CrossRef](#)] [[PubMed](#)]
43. Govia, L.C.G.; Ribeill, G.J.; Rowlands, G.E.; Ohki, T.A. Nonlinear input transformations are ubiquitous in quantum reservoir computing. *Neuromorph. Comput. Eng.* **2022**, *2*, 014008. [[CrossRef](#)]
44. Dudas, J.; Carles, B.; Plouet, E.; Mizrahi, F.A.; Grollier, J.; Marković, D. Quantum reservoir computing implementation on coherently coupled quantum oscillators. *NPJ Quantum Inf.* **2023**, *9*, 64. [[CrossRef](#)]
45. Götting, N.; Lohof, F.; Gies, C. Exploring quantumness in quantum reservoir computing. *Phys. Rev. A* **2023**, *108*, 052427. [[CrossRef](#)]
46. Llodrà, G.; Charalambous, C.; Giorgi, G.L.; Zambrini, R. Benchmarking the role of particle statistics in quantum reservoir computing. *Adv. Quantum Technol.* **2023**, *6*, 2200100. [[CrossRef](#)]
47. Čindrak, S.; Donvil, B.; Lüdge, K.; Jaurigue, L. Enhancing the performance of quantum reservoir computing and solving the time-complexity problem by artificial memory restriction. *Phys. Rev. Res.* **2024**, *6*, 013051. [[CrossRef](#)]
48. Harrington, P.M.; Monroe, J.T.; Murch, K.W. Quantum Zeno Effects from Measurement Controlled Qubit-Bath Interactions. *Phys. Rev. Lett.* **2017**, *118*, 240401. [[CrossRef](#)] [[PubMed](#)]

49. Raimond, J.M.; Facchi, P.; Peucedercf, B.; Pascazio, S.; Sayrin, C.; Dotsenko, I.; Gleyzes, S.; Brune, M.; Haroche, S. Quantum Zeno dynamics of a field in a cavity. *Phys. Rev. A* **2012**, *86*, 032120. [[CrossRef](#)]
50. Lewalle, P.; Martin, L.S.; Flurin, E.; Zhang, S.; Blumenthal, E.; Hacohen-Gourgy, S.; Burgarth, D.; Whaley, K.B. A Multi-Qubit Quantum Gate Using the Zeno Effect. *Quantum* **2023**, *7*, 1100. [[CrossRef](#)]
51. Kondo, Y.; Matsuzaki, Y.; Matsushima, K.; Filgueiras, J.G. Using the quantum Zeno effect for suppression of decoherence. *New J. Phys.* **2016**, *18*, 013033. [[CrossRef](#)]
52. Alex Monras, O.R.I. Quantum Information Processing with Quantum Zeno Many-Body Dynamics. *arXiv* **2009**, arXiv:0801.1959.
53. Paz-Silva, G.A.; Rezakhani, A.T.; Dominy, J.M.; Lidar, D.A. Zeno Effect for Quantum Computation and Control. *Phys. Rev. Lett.* **2012**, *108*, 080501. [[CrossRef](#)]
54. Burgarth, D.K.; Facchi, P.; Giovannetti, V.; Nakazato, H.; Yuasa, S.P.K. Exponential rise of dynamical complexity in quantum computing through projections. *Nat. Commun.* **2014**, *5*, 5173. [[CrossRef](#)]
55. Nielsen, A.E.B.; Mølmer, K. Stochastic master equation for a probed system in a cavity. *Phys. Rev. A* **2008**, *77*, 052111. [[CrossRef](#)]
56. Riou, M.; Araujo, F.A.; Torrejon, J.; Tsunegi, S.; Khalsa, G.; Querlioz, D.; Bortolotti, P.; Cros, V.; Yakushiji, K.; Fukushima, A.; et al. Neuromorphic computing through time-multiplexing with a spin-torque nano-oscillator. In Proceedings of the 2017 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, USA, 2–6 December 2017; pp. 36.3.1–36.3.4. [[CrossRef](#)]
57. Mackey, M.C.; Glass, L. Oscillation and chaos in physiological control systems. *Science* **1977**, *197*, 287–289. [[CrossRef](#)]
58. Shougat, M.R.E.U.; Perkins, E. The van der Pol physical reservoir computer. *Neuromorph. Comput. Eng.* **2023**, *3*, 024004. [[CrossRef](#)]
59. Maksymov, I.S.; Pototsky, A. Reservoir computing based on solitary-like waves dynamics of liquid film flows: A proof of concept. *EPL* **2023**, *142*, 43001. [[CrossRef](#)]
60. Maksymov, I.S. Physical reservoir computing enabled by solitary waves and biologically inspired nonlinear transformation of input data. *Dynamics* **2024**, *4*, 119–134. [[CrossRef](#)]
61. Jaeger, H. Echo state network. *Scholarpedia* **2007**, *2*, 2330. [[CrossRef](#)]
62. Mochalin, V.N.; Shenderova, O.; Ho, D.; Gogotsi, Y. The properties and applications of nanodiamonds. *Nat. Nanotech.* **2012**, *7*, 11–23. [[CrossRef](#)] [[PubMed](#)]
63. Basso, L.; Cazzanelli, M.; Orlandi, M.; Miotello, A. Nanodiamonds: Synthesis and application in sensing, catalysis, and the possible connection with some processes occurring in space. *Appl. Sci.* **2020**, *10*, 4094. [[CrossRef](#)]
64. Taflove, A.; Hagness, S.C. *Computational Electrodynamics: The Finite-Difference Time-Domain Method*; Artech House: Boca Raton, FL, USA, 2005.
65. Yang, J.; Perrin, M.; Lalanne, P. Analytical formalism for the interaction of two-level quantum systems with metal nanoresonators. *Phys. Rev. X* **2015**, *5*, 021008. [[CrossRef](#)]
66. Guo, W.H.; Li, W.J.; Huang, Y.Z. Computation of resonant frequencies and quality factors of cavities by FDTD technique and Pade approximation. *IEEE Microw. Wirel. Compon. Lett.* **2001**, *11*, 223–225.
67. Douvalis, V.; Hao, Y.; Parini, C. Reduction of late time instabilities of the finite difference time domain method in curvilinear coordinates. In Proceedings of the Fourth International Conference on Computation in Electromagnetics, CEM 2002 (Ref. No. 2002/063), Bournemouth, UK, 8–11 April 2002. :20020197 [[CrossRef](#)]
68. Reineck, P.; Trindade, L.F.; Havlik, J.; Stursa, J.; Heffernan, A.; Elbourne, A.; Orth, A.; Capelli, M.; Cigler, P.; Simpson, D.A.; et al. Not all fluorescent nanodiamonds are created equal: A comparative study. *Part. Part. Syst. Charact.* **2019**, *36*, 1900009. [[CrossRef](#)]
69. Reineck, P.; Lau, D.W.M.; Wilson, E.R.; Fox, K.; Field, M.R.; Deeleepojananan, C.; Mochalin, V.N.; Gibson, B.C. Effect of surface chemistry on the fluorescence of detonation nanodiamonds. *ACS Nano* **2017**, *11*, 10924–10934. [[CrossRef](#)]
70. Maksymov, I.S.; Kostylev, M. Magneto-electronic hydrogen gas sensors: A critical review. *Chemosensors* **2022**, *10*, 49. [[CrossRef](#)]
71. Maksymov, I.S.; Nguyen, B.Q.H.; Suslov, S.A. Biomechanical sensing using gas bubbles oscillations in liquids and adjacent technologies: Theory and practical applications. *Biosensors* **2022**, *12*, 624. [[CrossRef](#)] [[PubMed](#)]
72. Poroykov, A.Y.; Surkov, D.A.; Ilina, N.S.; Lebedev, S.V.; Ul'yanov, D.V.; Lapitsky, K.M.; Shmatko, E.V. Development of the flight laboratory for research of aerodynamic surfaces deformation. *J. Phys. Conf. Ser.* **2020**, *1636*, 012029. [[CrossRef](#)]
73. Henderson, A.; Yakopcic, C.; Harbour, S.; Taha, T.M. Detection and Classification of Drones Through Acoustic Features Using a Spike-Based Reservoir Computer for Low Power Applications. In Proceedings of the 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), Portsmouth, VA, USA, 18–22 September 2022; pp. 1–7. [[CrossRef](#)]
74. Vysotskii, V.I.; Vysotskyy, M.V. Fundamental prerequisites for realization of the quantum Zeno effect in the microwave and optical ranges. *Eur. Phys. J. D* **2022**, *76*, 158. [[CrossRef](#)]
75. Vidamour, I.T.; Swindells, C.; Venkat, G.; Manneschi, L.; Fry, P.W.; Welbourne, A.; Rowan-Robinson, R.M.; Backes, D.; Maccherozzi, F.; Dhesi, S.S.; et al. Reconfigurable reservoir computing in a magnetic metamaterial. *Commun. Phys.* **2023**, *6*, 230. [[CrossRef](#)]
76. Bar, D. The Zeno effect for spins. *Phys. A* **1999**, *267*, 434–442. [[CrossRef](#)]
77. Kominis, I.K. Quantum Zeno effect explains magnetic-sensitive radical-ion-pair reactions. *Phys. Rev. E* **2009**, *80*, 056115. [[CrossRef](#)]
78. Kumari, K.; Rajpoot, G.; Joshi, S.; Jain, S.R. Qubit control using quantum Zeno effect: Action principle approach. *Ann. Phys.* **2023**, *450*, 169222. [[CrossRef](#)]
79. Schizas, N.; Karras, A.; Karras, C.; Sioutas, S. TinyML for ultra-low power AI and large scale IoT deployments: A systematic review. *Future Internet* **2022**, *14*, 363. [[CrossRef](#)]
80. Khrennikov, A. Quantum-like brain: “Interference of minds”. *Biosystems* **2006**, *84*, 225–241. [[CrossRef](#)]

81. Atmanspacher, H.; Filk, T. A proposed test of temporal nonlocality in bistable perception. *J. Math. Psychol.* **2010**, *54*, 314–321. [[CrossRef](#)]
82. Busemeyer, J.R.; Bruza, P.D. *Quantum Models of Cognition and Decision*; Oxford University Press: New York, NY, USA, 2012.
83. Aerts, D.; Beltran, L. A Planck radiation and quantization scheme for human cognition and language. *Front. Psychol.* **2022**, *13*, 850725. [[CrossRef](#)] [[PubMed](#)]
84. Moreira, C.; Tiwari, P.; Pandey, H.M.; Bruza, P.; Wichert, A. Quantum-like influence diagrams for decision-making. *Neural Netw.* **2020**, *132*, 190–210. [[CrossRef](#)] [[PubMed](#)]
85. Martínez-Martínez, I.; Sánchez-Burillo, E. Quantum stochastic walks on networks for decision-making. *Sci. Rep.* **2016**, *6*, 23812. [[CrossRef](#)] [[PubMed](#)]
86. Maksymov, I.S. Quantum-inspired neural network model of optical illusions. *Algorithms* **2024**, *17*, 30. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.