Chaos, Solitons and Fractals 178 (2024) 114354

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

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

Forecasting coherence resonance in a stochastic Fitzhugh–Nagumo neuron model using reservoir computing

Alexander E. Hramov [a,b], Nikita Kulagin [a], Andrey V. Andreev [a,b], Alexander N. Pisarchik [c,a,*]

[a] Baltic Center for Neurotechnology and Artificial Intelligence, Immanuel Kant Baltic Federal University, Aleksandra Nevskogo Str., 14, Kaliningrad, 236041, Russia
[b] Engineering School of Information Technologies, Telecommunications and Control Systems, Ural Federal University, Mira Str., 194, Ekaterinburg, 620002, Russia
[c] Center for Biomedical Technology, Universidad Politécnica de Madrid, Campus Montegancedo, Pozuelo de Alarcón, 28223, Madrid, Spain

ARTICLE INFO

Keywords:
Coherence resonance
FitzHugh–Nagumo neuron
Stochastic system
Reservoir computer

ABSTRACT

We delve into the intriguing realm of reservoir computing to predict the intricate dynamics of a stochastic FitzHugh–Nagumo neuron model subjected to external noise. Through innovative reservoir design and training, we unveil the remarkable capacity of a reservoir computer to forecast the behavior of this stochastic system across a wide spectrum of noise intensity variations. Notably, our reservoir computer astutely replicates the intricate phenomenon of coherence resonance in the stochastic FitzHugh–Nagumo neuron, signifying the superior modeling capabilities of this approach. A detailed examination of the microscopic dynamics within the reservoir's hidden layer reveals the emergence of distinct neuronal clusters, each displaying unique behaviors. Certain neurons within the reservoir are adept at faithfully reproducing the dynamical traits of the neuron, particularly the spike generation mechanism. In contrast, the remaining neurons within the reservoir seem to emulate stochastic influences with remarkable precision, accurately capturing the moments of spike generation in the neuron under the sway of noise. This innovative reservoir design proves to be highly effective across a diverse range of noise control parameters, faithfully replicating the essential characteristics of the original stochastic FitzHugh–Nagumo neuron. These findings illuminate the potential of reservoir computing to model and predict the dynamics of complex stochastic systems, showcasing its adaptability and versatility in understanding and simulating natural phenomena.

1. Introduction

Complex nonlinear systems can exhibit very rich dynamics, encompassing chaos, structural formations, frequency entrainment, multistability, and intricate transitions when parameters are turned or external forcing is applied [1–3]. In most cases, the introduction of noise to such systems leads to heightened unpredictability in their dynamics. Nevertheless, there are instances where the noise interacts with the system's intrinsic order in a resonant manner. Under specific conditions, the motion reaches its peak coherence at a particular noise intensity level. This intriguing phenomenon, recognized as *coherence resonance*, has been observed in stochastic systems where regularity or coherence is maximized near critical points, such as a supercritical Hopf bifurcation [4] or a saddle–node bifurcation on the invariant circle [5], when noise intensity is judiciously varied.

Coherence resonance is a phenomenon that occurs across diverse natural and technical systems and complex networks (see, e.g., [6] and references therein). One prominent illustration of coherence resonance can be found in the FitzHugh–Nagumo (FHN) model, representing an excitable neuron [5]. In this model, resonant behavior emerges as a pronounced regularity in noise-induced oscillations when subjected to a specific intensity of external noise. This results in a resonant pattern in the regularity (or, in other words, coherence) of the neuron's spike sequence, with the spikes becoming notably more regular at the optimal noise intensity. Coherence resonance's manifestation hinges on the nature of the noise and is evident in the inter-event or inter-spike intervals within time series data, as well as in the characteristic correlation times that align with the resonance scale.

Coherent resonance has been the subject of investigation in a wide array of systems, ranging from mathematical models to experimental settings, encompassing neural networks [7,8], neurophysiological systems [9,10], and even the intricacies of the human brain [11]. Across these studies, the presence of noise, whether introduced externally or as an internal additive, has consistently influenced systems displaying resonant behavior in various coherence metrics. In many cases, such investigations can only be conducted through experiments,

* Corresponding author at: Center for Biomedical Technology, Universidad Politécnica de Madrid, Campus Montegancedo, Pozuelo de Alarcón, 28223, Madrid, Spain.
E-mail address: alexander.pisarchik@ctb.upm.es (A.N. Pisarchik).

https://doi.org/10.1016/j.chaos.2023.114354
Received 1 November 2023; Accepted 3 December 2023
Available online 15 December 2023

especially in the context of complex and less formally described systems like the human brain. However, recent years have witnessed the development of data-driven approaches, wherein models for poorly formalized processes and phenomena are constructed. These models generalize features using machine learning methods, drawing upon meticulously prepared and annotated datasets containing empirical, often experimental, data on the system under study.

One powerful tool within these model-independent machine learning techniques is the echo state network [12]. Also referred to as reservoir computers [13] or liquid state machines [14] in the literature, it has gained prominence for its efficacy in predicting and classifying dynamic systems. Its appeal lies in its training cost efficiency, architectural simplicity, and fixed reservoirs. In this context, reservoir computing plays a pivotal role in forecasting a diverse range of dynamical characteristics. A reservoir computer is a type of recurrent neural network (RNN) architecture designed for various machine learning and time-series prediction tasks. It is known for its simplicity, training efficiency, and strong predictive capabilities, particularly in dealing with complex dynamical systems, including predicting chaotic time series [15,16], estimating Lyapunov exponents [17], forecasting cluster synchronization [18], analyzing burst synchronization [19], observing spatiotemporal dynamics [20], characterizing macroscopic properties of complex adaptive networks [21], modeling basin of attraction [22], and elucidating stable/unstable manifolds, among others.

Despite the extensive research into reservoir computing, there is a notable gap when it comes to its application in predicting stochastic systems. The primary reason for this gap is the lack of efficient stochastic frameworks for reservoir computing, which currently hinder its ability to accurately predict stochastic dynamical systems. Specifically, when a reservoir computer is trained to forecast a stochastic system, it often encounters a swift accumulation of sliding prediction errors. In essence, establishing proper probabilistic models for prediction errors becomes imperative when dealing with stochastic dynamical systems [23–25]. This underscores the critical importance of researching and predicting the dynamics of systems influenced by noise, making it a pressing and significant challenge. Noteworthy recent contributions in this domain include the work by Grigoryeva and colleagues [26], where a time-delay reservoir computer, a type of echo state network, demonstrated robust performance in forecasting conditional covariances linked to multivariate discrete-time nonlinear stochastic processes of the VEC-GARCH variety. This model also excelled in predicting actual daily market realized volatilities based on intraday quotes, using daily log-return series of modest size as training input.

Another noteworthy advancement is presented in Fang et al.'s paper [27], where a data-driven framework fuses Reservoir Computing and Normalizing Flow to predict the long-term evolution of stochastic dynamical systems and replicate their behaviors. The authors validate the effectiveness of this framework through various simulations, encompassing systems like the stochastic Van der Pol oscillator, the simplified El Nino-Southern Oscillation model, and the stochastic Lorenz system. Furthermore, Liao and collaborators [28] propose a low-power-consumption physical reservoir computing model based on an overdamped bistable stochastic resonance system, offering an innovative avenue for efficient computation in stochastic environments.

Simultaneously, research exploring the application of reservoir computers in predicting the behavior of deterministic systems as control parameters are altered has attracted significant interest. In a noteworthy study, Fan and collegues [29] demonstrated the capability to anticipate the threshold coupling strengths at which synchronization modes occur. Similarly, Xiao et al. [30] successfully predicted amplitude death resulting from parameter drift. At the same time, Kim and co-authors [31] employed an alternative training scheme for recurrent neural networks, enabling the prediction of period doubling as systems transition towards chaos. In addition, in the study by Roy and team [22], an echo state network was trained on three time series with appropriate parameter values, creating a parameter-aware reservoir that exhibited remarkable accuracy in predicting dynamics across a range of parameters. Notably, the authors illustrated the reservoir computer's ability to infer dynamics with different parameters from training data, even when the training dynamics were relatively straightforward, such as pure periodic behavior.

Given the considerations above, pivotal questions arise: Can we expand the capacity of reservoirs to capture and generalize the dynamics of studied systems, potentially reducing it to a smaller number of parameters, perhaps even down to a single value? Is it conceivable to predict the behavior of a system across a wide spectrum of control parameters, especially when we include the characteristics of noise as one of these parameters? This intriguing avenue of exploration could significantly enhance our comprehension of how to forecast the behavior of stochastic systems amid variations in noise parameters. In light of these questions and to address the existing gap in the domain of predicting system behavior under the influence of varying control parameters, we employ a data-driven machine learning approach. Our objective is to predict the phenomenon of coherence resonance within the excitable FHN neuron, leveraging reservoir computing, while systematically adjusting the noise power.

2. Methods

2.1. FitzHugh–Nagumo neuron model

We employed a mathematical paradigmatic model of an excitable FHN neuron [32] as a representative model of an excitable system capable of exhibiting coherence resonance through the interplay of excitatory and restoring variables [33]. The governing equations for this stochastic system are as follows:

$$\dot{x} = x - \frac{x^3}{3} - y + RI, \qquad \dot{y} = 0.08(x - 0.8y + 0.7) + D\xi(t), \tag{1}$$

where x represents the excitatory variable, often referred to as the membrane potential, while y corresponds to the recovery variable, I denotes the magnitude of the stimulus current, with R set to 1. Additionally, $\xi(t)$ signifies the zero-mean white Gaussian noise with an autocorrelation function of $\langle \xi(t)\xi(t+\tau)\rangle = \delta(\tau)$, and D represents the noise amplitude. The I parameter, indicative of the stimulus current, plays a pivotal role in determining the equilibrium points and, consequently, the neuron's excitability threshold.

The system described by Eq. (1) operates in either an excitable regime (when $I < 0.3218$) or an oscillatory regime (when $I > 0.3218$), undergoing a Hopf bifurcation at $I \approx 0.3218$. For our study, we focus on the excitable regime, specifically in the pre-critical state where the system remains devoid of self-sustained oscillations. To do this, we set the stimulus current to $I = 0.3$, a value well within the excitable regime. In this configuration, the system Eq. (1) without any noise influence ($D = 0$) reaches a stationary state. However, as the noise intensity D increases, the system exhibits spike generation, with the characteristics of these spikes becoming increasingly dependent on the noise intensity.

For numerical solution, we employed the first-order Euler method with a time step of integration set at $\Delta t = 0.1$.

2.2. Reservoir computer architecture

The architecture of a reservoir computer typically consists of three primary components: Input Layer which receives the external input data or time series, Reservoir Layer which comprises a large number of recurrent neurons and randomly generated connections between them, and Output Layer responsible for producing predictions or classifications based on the information stored and processed in the reservoir. In our work, we employ a reservoir computer architecture known as the echo state network. This design utilizes an ensemble of interconnected nodes as its internal reservoir [13,34,35].

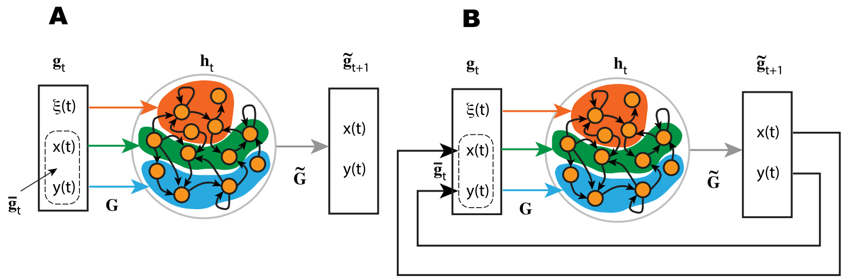

Fig. 1. Schematic presentation of the reservoir computer with fully separated inputs in the training mode (**A**) and predicting (testing) mode (**B**).

Fig. 1 illustrates the schematic structure of the reservoir computer model we considered. A distinctive aspect of our proposed reservoir configuration is the complete segregation of reservoir inputs among different neurons within the hidden inner layer. It is this unique feature that enables us to attain remarkable efficiency in prediction and modeling of the stochastic neuron. We refer to this specific type of echo state network as a reservoir computer with fully separated inputs.

The input vector, denoted as $\mathbf{g}(t)$, comprises three components, namely, $\mathbf{g}(t) = (D\xi(t), x(t), y(t))^T$, corresponding to the external noise exciting the neuron and the dynamical variables of the neuron model. This input vector influences the internal high-dimensional hidden state of the reservoir, represented as $\mathbf{h}_t$, with a dimension of $N_h = 3n$ (where $n \in \mathbb{N}^+$ is a member of the set of natural numbers excluding zero). The coupling of the input to the hidden state is mediated by the input-to-hidden matrix, $\mathbf{G}$, which is an $N_h \times 3$ matrix with elements taking the values

$$G_{i,j} = \begin{cases} \delta_{i,j}, & 1 + (j-1)n \le i \le jn, \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where $\delta_{i,j}$ are uniformly sampled from interval $[-\sigma, \sigma]$, index j can take values from the set $(1, 2, 3)$, i.e., the external noise corresponding to the first component of the input vector ($j = 1$) affects the first n neurons of the hidden layer of the reservoir with weights $\delta_{i,1}$, variable x ($j = 2$) is related to the next n neurons with weights $\delta_{i,2}$, and variable y ($j = 3$) corresponds to the remaining n neurons with weights $\delta_{i,3}$, as shown in Fig. 1A.

In addition to inputs originating from the input layer, each reservoir node i receives inputs from other reservoir nodes, with corresponding weights dictated by the reservoir's (hidden-to-hidden) adjacency matrix, denoted as $\mathbf{W}$, possessing dimensions $N_h \times N_h$. This hidden-to-hidden matrix $\mathbf{W}$ defines a random network characterized by the mean node degree $\langle k \rangle$. In network theory, the mean node degree $\langle k \rangle$ refers to the average number of links that a node possesses. To ensure network stability, the hidden-to-hidden matrix $\mathbf{W}$ is rescaled in such a way that the network's spectral radius is denoted as λ (the absolute value of the largest eigenvalue of the adjacency matrix).

Several key hyperparameters determine the configuration of the reservoir. These include σ, N_h, $\langle k \rangle$, and λ, and each must be optimally chosen when designing a specific reservoir computer. In our case, we established the following values for these hyperparameters: σ was fixed at a constant value of 1, $N_h = 1002$, $\langle k \rangle$ was optimized within the range of $[10, 20]$, and λ was explored within the interval of $[0.5, 0.9]$.

An important and defining characteristic of the reservoir computer is that once the matrices $\mathbf{G}$ and $\mathbf{W}$ are created, they remain unaltered throughout the process. This means that the input vector $\mathbf{g}$ enters the reservoir in a fixed manner, and the network's internal connections between neurons within the reservoir remain unchanged.

Each reservoir node also possesses an output, which can be described by the following equation:

$$\mathbf{h}_t = \varphi\left(\mathbf{G}\mathbf{g}_t + \mathbf{W}\mathbf{h}_{t-1}\right). \tag{3}$$

In this equation, $\mathbf{h}_t$ represents the internal hidden state, allowing for the encoding of temporal dependencies based on past state history. The function $\varphi(\cdot)$ signifies the activation function of the hidden layer neurons, applied element-wise to the state vector $\mathbf{h}_t$. In this study, we employed hyperbolic tangent functions as the activation functions, denoted as $\varphi(\cdot) = \tanh(\cdot)$.

The objective of reservoir computing in this paper is to forecast the behavior of a nonlinear stochastic system Eq. (1), governed by noise, using the known signals of dynamical variables $x(t)$ and $y(t)$, along with the external noise $\xi(t)$. The input signals $x(t)$ and $y(t)$ and the desired output signals $\tilde{x}(t)$ and $\tilde{y}(t)$ in our work all stem from the same system. Subsequently, employing a readout hidden-to-output matrix denoted as $\tilde{\mathbf{G}}$, the reservoir computer's estimate at time $(t+1)$ is computed using the following equation:

$$\tilde{\mathbf{g}}_{t+1} = \tilde{\mathbf{G}}\hat{\mathbf{h}}_t, \tag{4}$$

Here, the augmented hidden state, represented as $\hat{\mathbf{h}}_t$, is an N_h-dimensional vector. Each component of $\hat{\mathbf{h}}_t$ is defined as follows:

$$\hat{h}_{i,t} = \begin{cases} h_{i,t}, & i \text{ is odd}, \\ h_{i,t}^2, & i \text{ is even}, \end{cases} \tag{5}$$

This augmentation enriches the dynamics by squaring the hidden state in half of the nodes [22]. The augmented output from each reservoir node is then directed to the output layer of the reservoir computer, where a linear operation involving specific weights is applied to generate the overall reservoir output.

Typically, when applying reservoir computing for predicting temporal sequences, the dimension of the output data vector matches that of the input data vector. However, in this particular scenario, we must deviate from the standard procedure to construct the reservoir computer. This is because predicting the random process $D\xi(t)$, which acts as an external independent influence on the system Eq. (1) under investigation, is not feasible. Consequently, the output vector $\tilde{\mathbf{g}}_t$ comprises only two components, namely $\tilde{\mathbf{g}}_t = (\tilde{x}(t), \tilde{y}(t))^T$, and the output layer $\tilde{\mathbf{G}}$ takes the form of a $(2 \times N_h)$ matrix with trainable weights. For convenience in subsequent analysis, we can introduce the target state of the neuron at time t as a vector denoted by $\bar{\mathbf{g}}_t = (x(t), y(t))^T$.

During the training process, the input signal $\mathbf{g}_t$, encompassing the driving noise $D\xi(t)$ and the FHN neuron signals $x(t)$ and $y(t)$, is fed into the input of the reservoir computer, and the corresponding output signal $\tilde{\mathbf{g}}_{t+1}$ is obtained. To train the output layer $\tilde{\mathbf{G}}$, the L_2-error between the target states $\bar{\mathbf{g}}_t$ and the estimated states $\tilde{\mathbf{g}}_t$ is minimized. This is achieved using ridge regression [36], which helps mitigate overfitting by penalizing excessive values of the fitting parameters. The aim is to determine an output matrix $\tilde{\mathbf{G}}$ that minimizes the following loss function:

$$L_2^{RC} = \sum_{t=1}^{T_{train}} \|\bar{\mathbf{g}}_t - \tilde{\mathbf{g}}_t\|^2 + \beta\|\tilde{\mathbf{G}}\|^2, \tag{6}$$

where the term $\beta\|\tilde{\mathbf{G}}\|^2$ is included to prevent overfitting, with β set to 10^{-8} as the L_2 regularization hyperparameter.

After the completion of training, we initiate the prediction phase, during which the reservoir computer attempts to autonomously predict signal dynamics, as illustrated in Fig. 1B. The prediction phase employs the same reservoir equations as described above [refer to Eqs. (3) and (4)], with the output layer weights $\tilde{\mathbf{G}}$ already determined. However, the input vector $\bar{\mathbf{g}}_{t+1}$ is now represented by the computed output vector $\tilde{\mathbf{g}}_{t+1}$ from the previous step, while external noise (noise from the first input $g_1(t) = D\xi(t)$) continues to drive the reservoir computer, maintaining continuous external noise influence.

The output vector $\tilde{\mathbf{g}}_{t+1}$ can be viewed as the macroscopic variables describing the integral dynamics of the trained reservoir computer under the influence of noise. Meanwhile, the internal state of the reservoir, $\mathbf{h}_t$, characterizes the internal microscopic dynamics of the reservoir [37]. This approach aligns in many aspects with the description of neuronal ensemble dynamics in the brain, whether at the level of macroscopic signals recorded using non-invasive techniques like electroencephalography (EEG) or magnetoencephalography (MEG), or at the microscopic level of individual neurons, which is examined through invasive electrode recordings [38].

2.3. Coherence resonance measure

To assess the regularity (coherence) of the spike train produced by the neuron or the reservoir computer, we employ the coefficient of variation. This measure is defined as the standard deviation σ_{ISI} of the interspike interval (ISI) normalized to the average ISI [5]:

$$\mathcal{R} = \sigma_{ISI}/\langle \Delta T\rangle, \qquad \sigma_{ISI} = \sqrt{\langle \Delta T_i^2\rangle - \langle \Delta T\rangle^2}, \tag{7}$$

where $\Delta T_i = T_{i+1} - T_i$ represents the duration of the ith ISI, $\langle \Delta T\rangle$ is the average ISI, and M denotes the number of spikes in the analyzed sequence ($m = 1, \dots, M$).

In the assessment of the spike sequence generated by the stochastic neuron, we utilize the variable $x(t)$ from Eq. (1) and denote the coefficient of variation as $\mathcal{R}$. When evaluating the spike sequence predicted by the reservoir computer, we employ the variable $\tilde{g}_{1t} = \tilde{x}(t)$ and refer to the result as $\mathcal{R}^{RC}$.

When the coefficient of variations $\mathcal{R}$ reaches its minimum value with respect to the noise amplitude D, we are dealing with coherence resonance, meaning that all ISIs ΔT_i are distributed in a narrower region towards the average value $\langle T\rangle$. The inverse value $\mathcal{R}^{-1}$ characterizes the regularity of the spike train generated by the neuron.

To estimate regularity measure of temporal dynamics of internal state of the reservoir $\mathbf{h}_t$ describes the microscopic dynamics of the reservoir computer we also use the coefficient of variations defined by Eq. (7) generalized to a neural network. In this case we evaluate the normalized standard deviation of the average ISI as follows

$$\mathcal{R}^{RC}_{IS} = \frac{\sqrt{\overline{\langle \Delta T_i^2\rangle} - \overline{\langle \Delta T\rangle}^2}}{\overline{\langle \Delta T\rangle}}, \tag{8}$$

where the over-line indicates the additional average over considered nodes of hidden layer of the reservoir computer.

3. Results

3.1. Macroscopic dynamics of reservoir computer versus stochastic FitzHugh–Nagumo neuron

We initiate our investigation by analyzing the macroscopic signal dynamics of the reservoir computer $\tilde{\mathbf{g}}_{t+1}$ as it predicts the behavior of the output variables $x(t)$ and $y(t)$ in the stochastic FHN neuron. In Fig. 2A, we illustrate the relationship between the coefficient of variation $\mathcal{R}$ given by Eq. (7) and noise intensity, which demonstrates resonant behavior at varying noise intensities for both the stochastic FHN neuron and the reservoir computer.

As the noise intensity increases, there is a noticeable reduction in the coefficient of variation, indicating an enhancement in the regularity of the output spike sequence. This reaches a minimum at $D = D^* \approx 0.23$, corresponding to the most regular signal. Subsequently, with a further increase in noise intensity, we observe an increase in the coefficient of variation, implying a decrease in the coherence of the spike sequence.

To predict the dynamics of a stochastic neuron, we employed the technique outlined in Section 2.2 to train the reservoir computer. Specifically, we selected a single noise intensity value, denoted as D^0, and used signals $D^0\xi(t)$, $x(t)$, and $y(t)$ with a duration of T_{train} to generate the input signal $\mathbf{g}_t$ during the training phase of the reservoir computer, as depicted in Fig. 1A. For this specific noise intensity value, D^0, we determined the output weight matrix $\tilde{\mathbf{G}}^{D^0}$ that minimizes the loss function Eq. (6). Following the completion of training, we initiated the testing phase by using the first point of the testing signal $\mathbf{g}_t$ as an initial condition for the reservoir computer. We then compared the actual signal with the predicted signal over a time interval of T_η. The primary feature we compared was the coefficient of variation, estimated from the spike statistics over a designated time interval, for both the actual signal generated by the neuron model (specifically, the variable x) and the signal predicted by the reservoir (in this case, variable $\tilde{g}_1 = \tilde{x}$).

Subsequently, we selected a different value, denoted as $D \neq D^0$, and conducted calculations in the prediction phase using the same trained reservoir, which retained the same hidden-to-hidden matrix $\mathbf{W}$ and the output weight matrix $\tilde{\mathbf{G}}^{D^0}$ corresponding to the noise intensity D^0. In this case, a noise signal $g_1(t) = D\xi(t)$ was applied to the reservoir. While any pair of values (x_0, y_0) could serve as the initial condition, to ensure a fair comparison between the signal of the stochastic FHN neuron and the macroscopic signal of the reservoir, it was essential to select identical initial values (x_0, y_0) for both the FHN neuron and the reservoir, and to provide the same realization $D\xi(t)$ of the noise to both the neuron and the reservoir. This level of matching was not a prerequisite when predicting only the statistical characteristics of the output signal, such as the coefficient of variation $\mathcal{R}$.

We generated the output macroscopic signal over a time duration of $T_\eta = 500000$ and determined the coefficient of variation $\mathcal{R}^{RC}(D)$ of the predicted signal. Fig. 2A illustrates the dependencies of the coefficient of variation $\mathcal{R}^{RC}_{D^0_l}$ of the predicted signal $\bar{\mathbf{g}}$ as a function of noise intensity D in comparison to the true coefficient of variation $\mathcal{R}$ for the stochastic neuron. These results were obtained for three distinct values of noise intensity D^0_l ($l = 1, 2, 3$) for which the reservoir had been trained.

It is worth noting that as the noise intensity D increased, the spike frequency of the stochastic neuron also increased. Consequently, we adjusted the duration T_{train} of the input signal used for training the reservoir so that the number of spikes remained constant and approximately equal to 75. Fig. 2A clearly indicates that a reservoir trained for only one specific noise intensity value (which could be selected either on the left branch of the curve $\mathcal{R}(D)$ at $D = D^0_1 = 0.05$ ($T_{train} = 48500$), at the minimum $D^0_2 = D^* \approx 0.23$ ($T_{train} = 35000$), or on the right branch of the curve $D^0_3 = 1.0$ ($T_{train} = 30000$) effectively predicts the effect of coherence resonance. In other words, the trained reservoir can accurately model the statistical characteristics of the output signal at various noise levels $D \neq D^0$. This demonstrates that the reservoir, trained with a specific noise value $D = D^0$, successfully captures the dynamics of the neuron at different noise levels and accurately replicates the coherence resonance phenomenon within the reservoir computer.

Figs. 3 present the predicted time series $\tilde{g}1, t$ generated by the reservoir computer, which was trained using noise intensities $D^0_{1,2,3}$, alongside the target time series $x(t)$ of the stochastic neuron for three distinct fixed noise intensities, namely $D_{1,2,3}$. The gray boxes in the figures denote the values of D_i that correspond to D^0_i, which represent the noise intensities used for training the reservoir computer. The choice of initial values (x_0, y_0) was identical for both the stochastic neuron and the reservoir. It is evident that the reservoir computer accurately replicates the dynamics of the stochastic neuron at the same

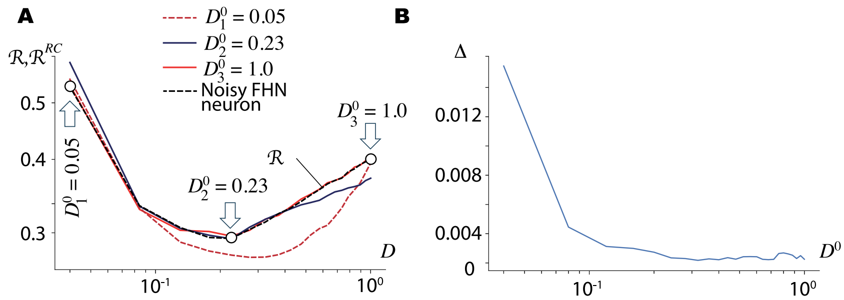

Fig. 2. (**A**) Coefficient of variations of the spike train versus noise intensity D for the stochastic FHN neuron model, $\mathcal{R}$, (black dashed line) and the reservoir computer, $\mathcal{R}^{RC}$, trained for three values of noise intensity $D_1^0 = 0.05$ (red dashed line), $D_1^0 = 0.23$ (solid black line), and $D_1^0 = 1.0$ (red solid line) [in a semi-logarithmic scale]. (**B**) Prediction performance Δ of the reservoir computer [see Eq. (9)] versus noise intensity D^0 at which the reservoir was trained. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

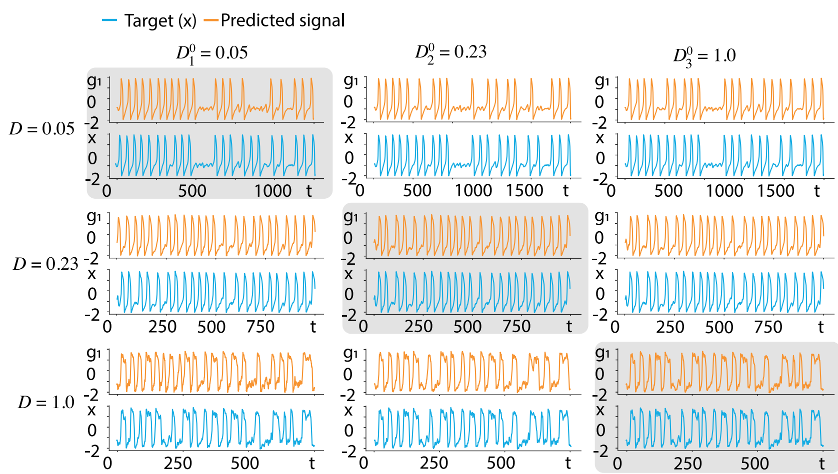

Fig. 3. Predicted time series $\tilde{g}_{1,t}$ of the reservoir computer trained with noise intensities $D_{1,2,3}^0$ and target time series $x(t)$ of the stochastic neuron Eq. (1) for three fixed values of noise intensity $D_{1,2,3}$. The gray boxes show the values of D_i corresponding to D_i^0 at which the reservoir computer was trained.

noise levels. Furthermore, the best-fit of the time series is observed at the highest noise intensity D_3.

As seen from Fig. 2A, the most precise correspondence in the behavior of the degree of spike regularity occurs when the reservoir is trained using signals with the maximum noise intensity, $D^0 = D_3$. In this case, the dependencies of the coefficient of variations of both the stochastic neuron and the reservoir computer are nearly identical. However, when the reservoir is trained at smaller noise levels (D_1 and D_2), some prediction deviations become apparent, with these deviations becoming more prominent as the noise intensity D^0 decreases. This phenomenon is illustrated in Fig. 2B, which depicts the prediction performance $\Delta\left(D^0\right)$ of the reservoir as a function of the noise intensity D^0 at which the reservoir was trained. For each chosen noise value D^0, we obtain the dependence $\mathcal{R}_{D^0}^{RC}(D)$. To assess the prediction performance from the perspective of the ISI statistics, we compute the dependence of

$$\Delta\left(D^0\right) = \left(\int \left(\mathcal{R}(D) - \mathcal{R}_{D^0}^{RC}(D)\right)^2 dD\right)^{1/2}, \tag{9}$$

integrated across all noise intensity values D. It is evident that the prediction error remains minimal and relatively constant across the entire range of noise variations for $D^0 > D^*$. However, when $D^0 < 0.1$, the prediction performance $\Delta\left(D^0\right)$ begins to decrease sharply. This decrease aligns with the slight deviations in the statistical characteristics of the spike sequences, such as the coefficient of variations, from those of the stochastic neuron, as seen in Fig. 2A.

These deviations may be attributed to the increase in the average ISI time, which could potentially degrade the quality of reservoir training. This is because longer training samples are required to maintain the same number of spikes used for training, thereby affecting the echo state network's ability to predict effectively. As a result, the network may face the issue of overfitting. Consequently, while the network continues to exhibit resonant behavior, it may fail to accurately predict statistical characteristics at specific noise values.

Our findings indicate that a reservoir, when subjected to a stochastic signal, can successfully predict the dynamics of a noisy neuron and the phenomenon of coherence resonance. Remarkably, this prediction can be achieved when the reservoir is exclusively trained using a single noise intensity value, denoted as D^0. When exposed to varying noise intensities (D) post-training, the pre-trained reservoir fully captures the dynamics of a stochastic neuron, including the effect of coherent resonance as noise intensity changes. Crucially, the prediction accuracy of the ISI statistics remains largely constant across the entire range of potential reference noise intensity values used for training. This consistency is only compromised when D^0 becomes very small (e.g., $D^0 < 0.1$), which we attribute to the low spike frequency that hampers the training process efficiency.

3.2. Coherence resonance in microscopic dynamics of reservoir computer

Now, let us delve into the microscopic dynamics of the trained reservoir computer as we vary the intensity of the noise influencing the system. As previously explained in Section 3.1, the most accurate alignment between the reservoir's dynamics and the dynamics of the stochastic neuron model, with respect to varying noise intensity, occurs when the reservoir is trained under high noise conditions (specifically, $D^0 = 1.0$). Consequently, this section exclusively focuses on the dynamics of the reservoir trained at $D^0 = D_3 = 1.0$.

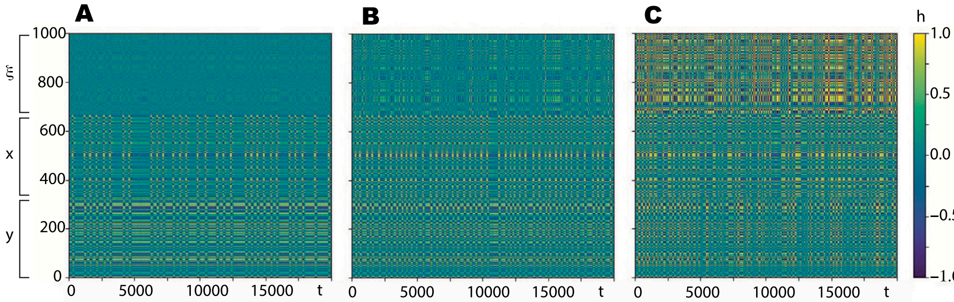

Fig. 4. Reservoir's internal hidden state dynamics for different values of noise intensity **(A)** $D_1 = 0.05$, **(B)** $D_2 = 0.23$, **(C)** $D_3 = 1.0$. According to Eq. (2) the first 334 neurons with $i = 1, \dots n$ corresponds to the hidden layer neurons affected by a noise signal ξ, the next 334 neurons with $i = n+1, \dots 2n$ — variable x, and the last 334 neurons with $i = 2n+1, \dots 3n$ — variable y, where $n = 334$. The scale on the left describes the amplitudes h_i of the states of the hidden layer neurons.

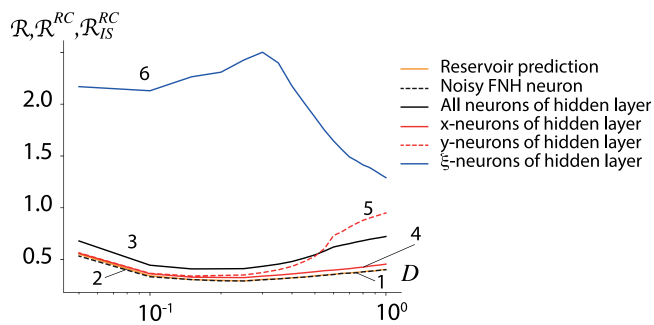

Fig. 5. Coefficient of variations of the spike train versus noise intensity D for the stochastic FHN neuron model, $\mathcal{R}$, (orange solid line 1); for the output macroscopic signal $\tilde{g}_{1,t}$ of reservoir computer, $\mathcal{R}^{RC}$, trained for noise intensity $D_3^0 = 1.0$ (black dashed line 2); for the whole hidden layer neural network of reservoir computer (black solid line 3); and only for the sub-networks of hidden layer neurons, $\mathcal{R}^{RC}_{IS}$, receiving only $x(t)$-variable (red solid line 4), $y(t)$-variable (red dashed line 5), and noise $D\xi(t)$ (blue solid line 6) inputs [in a semi-logarithmic scale]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 4 displays the typical internal hidden state dynamics of the reservoir for various noise intensity values, specifically $D_{1,2,3}$. The horizontal axis represents time, while the vertical axis corresponds to the ordinal number i of the hidden layer neurons. According to Eq. (2), the initial 334 neurons in the hidden layer receive input in the form of noise $D\xi(t)$, the following 334 artificial neurons receive the variable $x(t)$, and the last 334 neurons receive the variable $y(t)$. The amplitude h_i of hidden layer neuron activation is represented by the color scale, ranging from −1 to 1, a result of employing the hyperbolic tangent activation function φ described in Eq. (3).

Based on the activation dynamics of the artificial neurons in the hidden layer, as governed by Eq. (8), we computed the coefficient of variations $\mathcal{R}^{RC}_{IS}$ to analyze the hidden layer neural network's dynamics. The corresponding results are presented in Fig. 5. This figure illustrates the coefficients of variations $\mathcal{R}^{RC}_{IS}$ plotted against noise intensity D for the entire hidden layer neural network and separately for the sub-networks of neurons that exclusively receive input from the $x(t)$, $y(t)$, and noise $D\xi(t)$ variables. Additionally, it includes the coefficient of variations $\mathcal{R}^{RC}$ calculated from the reservoir's macroscopic dynamics (originating from Fig. 2A for ease of comparison). Fig. 5 clearly illustrates that the degree of coherence in the microscopic dynamics across the complete reservoir hidden layer network and the subnetworks of neurons linked to external inputs corresponding to the x and y variables generally follows the behavior observed in the stochastic neuron and the macroscopic reservoir dynamics, as discussed in the previous section.

Simultaneously, the dynamics of artificial neurons within the hidden layer display a considerably higher level of irregularity, as indicated by the larger values of $\mathcal{R}^{RC}_{IS}$ computed from the dynamics of the hidden layer neuron ensemble. Among the sub-networks of artificial neurons that solely receive input from the $x(t)$ variable, the dynamics are the closest in quantitative agreement with the dynamics of the analyzed stochastic neuron. The sub-networks of neurons receiving input from the $y(t)$ variable show more noticeable deviations from the dynamics of the stochastic neuron but still exhibit the effect of coherent resonance. Conversely, the dynamics of neurons influenced by noise input do not show the coherent resonance effect at all.

Therefore, the architecture of a reservoir with input separation based on the original system variables, including noise, results in complex dynamics within the hidden layer network. Different artificial neurons exhibit varying behaviors as the noise intensity changes. However, the process of optimizing the output weights through the loss function Eq. (6)) and forming a hidden-to-output matrix $\tilde{\mathbf{G}}$ enables the predicted signal $\tilde{\mathbf{g}}$ to closely match the target signal of the stochastic neuron $x(t)$ and $y(t)$. This raises the question: how is this achieved?

To address this question, we can calculate the coefficient of variation $\mathcal{R}_i$ for each neuron in the hidden layer (as defined in Eq. (7)) and compare it to the amplitude of the coefficients in the hidden-to-output matrix $\tilde{\mathbf{G}}$. This matrix characterizes the extent to which a particular neuron influences the output values $\tilde{\mathbf{g}}$. To facilitate the comparison, we can plot these values for each hidden layer neuron in a coordinate system, where the coefficient of variation $\mathcal{R}_i$ is on one axis, and the corresponding element in the output layer matrix $\tilde{G}_{1,i}$ or $\tilde{G}_{2,i}$ for the variables $\tilde{g}_1 = x$ and $\tilde{g}_2 = y$, respectively, is on the other axis. The results can be visualized in Fig. 6, separately for the x and y variables. To differentiate between neurons receiving input from noise ξ, x, and y variables, we can use different colors for the points representing them.

Based on Figs. 6A and 6B, it is clear that two distinct clusters of neurons can be identified from the presented two-dimensional distribution of individual artificial neurons in the hidden layer. We will refer to these clusters as the "I cluster" and the "II cluster." The "I cluster" mainly consists of neurons that receive input from the variable $x(t)$. These neurons exhibit a high degree of coherence, with $\mathcal{R} < 0.5$, while they vary widely in the amplitude of the coefficients of the hidden-to-output matrix $\tilde{\mathbf{G}}$, featuring both large and small values.

On the other hand, the "II cluster" is characterized by a lower degree of regularity, with $\mathcal{R} > 0.5$ and a mean value of approximately $\mathcal{R} \approx 0.716$. This cluster has less distinct boundaries and includes a long tail of noisy neurons with $\mathcal{R} > 1.0$. The "II cluster" predominantly consists of hidden layer neurons that receive input from the noise variable ξ and the variable $y(t)$ of the stochastic neuron.

Let us now proceed with the process of selecting hidden-to-output connections to predict the signal of the stochastic neuron using only the artificial neurons from the first cluster or the second cluster. Mathematically, this involves the transformation of the hidden-to-output matrix $\tilde{\mathbf{G}}'$ to obtain a new matrix $\tilde{\mathbf{G}}'$, i.e. $\tilde{\mathbf{G}}'$

$$\tilde{G}'_{j,i} = \begin{cases} \tilde{G}_{j,i}, & \mathcal{R}_i < 0.5, \\ 0, & \mathcal{R}_i \geq 0.5 \end{cases} \tag{10}$$

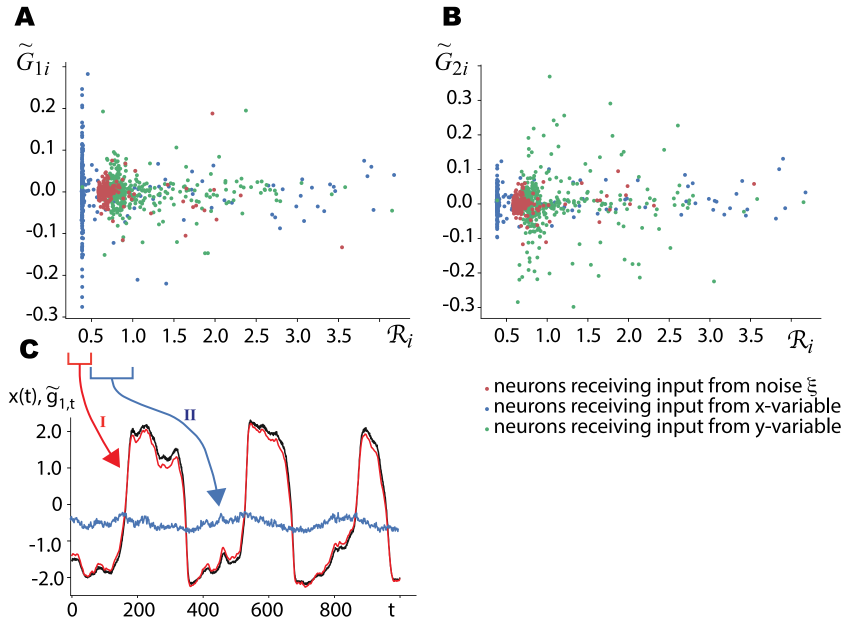

Fig. 6. (**A**) Representation of all neurons in the hidden layer of the reservoir computer (indexed as $i = 1, \dots, N_h$) in the spaces defined by the coefficient of variation $\mathcal{R}_i$ and the coefficient of the output layer matrix $\tilde{G}1, i$. (**B**) The same representation for $\mathcal{R}_i$ and $\tilde{G}_{2,i}$. The red points (with $i = 1, \dots, N_h/3$) correspond to neurons that receive input from the noise variable $D\xi$, the blue points ($i = N_h/3 + 1, \dots, 2N_h/3$) represent neurons receiving input from the x-variable, and the green points ($i = 2N_h/3 + 1, \dots, N_h$) depict neurons receiving input from the y-variable. (**C**) Comparison between the target signal of the $x(t)$-variable from the stochastic neuron (black line) and the predicted signals $\tilde{g}_{1,t}$ recovered from the neurons of the first I cluster (red line) and the second II cluster (blue line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

in the case of the first cluster, and

$$\tilde{G}'_{j,i} = \begin{cases} 0, & \mathcal{R}_i < 0.5 \lor \mathcal{R}_i > 1.0, \\ \tilde{G}_{j,i}, & 1.0 \geq \mathcal{R}_i \geq 0.5 \end{cases} \tag{11}$$

in the case of the second cluster. To recover the predicted signal $\tilde{\mathbf{g}}$ in Eq. (4), we use new matrices Eq. (10) or Eq. (11) depending on the cluster used: $\tilde{\mathbf{g}}_{t+1} = \tilde{\mathbf{G}}'\hat{\mathbf{h}}_t$.

Fig. 6C provides an illustration of the results obtained for the case when $D_3 = 1.0$. The figure shows the target signal of the stochastic neuron and the predicted signals recovered from the neurons of the first I cluster using Eq. (10) and the second II cluster using Eq. (11). It is evident that in the first case, the predicted signal $\tilde{g}_1$ closely matches the target signal x of the stochastic neuron, indicating that the remaining neurons of the second cluster do not significantly contribute to the original signal. This is further supported by the signal reconstructed from the neurons of the second cluster, which appears as an amplitude-modulated noise signal.

These results imply that the separation of input signals in the reservoir during its training and prediction, as depicted in Fig. 1, allows for an effective separation of the sub-network responsible for predicting spike generation and the sub-network that characterizes the noise influence on the neuron. It is crucial to segregate the hidden layer neurons into sub-networks that receive different signals. To demonstrate the significance of this separation, let us briefly consider a modified reservoir model that lacks such input signal separation, as described in Section 2.2.

Fig. 7A presents a reservoir computer in which the segregated artificial neurons of the hidden layer receive signals from the $x(t)$ and $y(t)$ variables of the stochastic neuron, similar to the previously considered reservoir configuration [see Eq. (2)]. However, in this configuration, the input associated with the noise signal ξ affects all neurons in the hidden layer. Consequently, the elements of the coupling input-to-hidden matrix $\mathbf{G}$ (also with dimension $N_h \times 3$, where $N_h = 2n$ and $n \in \mathbb{N}^+$) take the form: $G_{i,1} = \delta_i$, where $1 \leq i \leq N_h$, $G_{i,2} = \delta_i$ if $1 \leq i \leq N_h/2$, and $G_{i,2} = 0$ if $N_h/2 < i \leq N_h$, $G_{i,3} = 0$ if $1 \leq i \leq N_h/2$, and $G_{i,3} = \delta_i$ if $N_h/2 < i \leq N_h$. Here, the values δ_i are uniformly sampled from the interval $[-\sigma, \sigma]$. All other aspects of the reservoir design, its training method, and quantitative hyperparameters remain consistent with the previous case.

Let us now consider the outcomes of this modified reservoir model and discuss any differences in its performance compared to the earlier configuration.

Training this reservoir at different noise intensities D^0 allows us to reproduce and predict the dynamics of the stochastic FHN neuron only at the same noise value $D \approx D^0$. However, this reservoir design does not enable accurate predictions of the system's behavior at other noise values D. This limitation is illustrated in Fig. 7B, which shows the coefficients of variation $\mathcal{R}^{RC}$ at different noise values D for reservoirs trained at $D_1^0 = 0.08$, $D_2^0 = 0.23$, and $D_1^0 = 0.36$. The plots reveal that only at high noise intensity ($D^0 > 0.35$) can the reservoir qualitatively reproduce the coherent resonance dynamics observed in the stochastic FHN neuron.

Fig. 7C further illustrates the prediction performance Δ (as defined in Eq. (9)) as a function of the noise intensity D^0 at which the reservoir was trained, allowing for a direct comparison with Fig. 2B. In contrast to the previous case, this reservoir configuration only achieves a reliable reproduction of the stochastic FHN neuron's dynamics at $D^0 > 0.2$, and an acceptable level of accuracy for reproducing the stochastic resonance effect is only reached at noise intensity $D^0 > 0.4$.

This distinct behavior is influenced by the microscopic dynamics of the modified reservoir within its hidden layer. Fig. 7D displays the coefficient of variation $\mathcal{R}_i$ (defined in Eq. (7)) and the amplitude of the coefficients of the hidden-to-output matrix $\tilde{\mathbf{G}}$) for each ith neuron in the hidden layer, with respect to $\tilde{g}_1 = x$ and $\tilde{g}_2 = y$ variables, respectively (for comparison, see Fig. 6A, B). In this case, we observe that, unlike the reservoir with fully separated inputs, there is no clear structure or specialization in the distribution of artificial neurons in the hidden layer. This lack of specialization hampers the construction of an effective scheme for predicting the dynamics of a system of stochastic differential equations, specifically in predicting the behavior of coherence resonance.

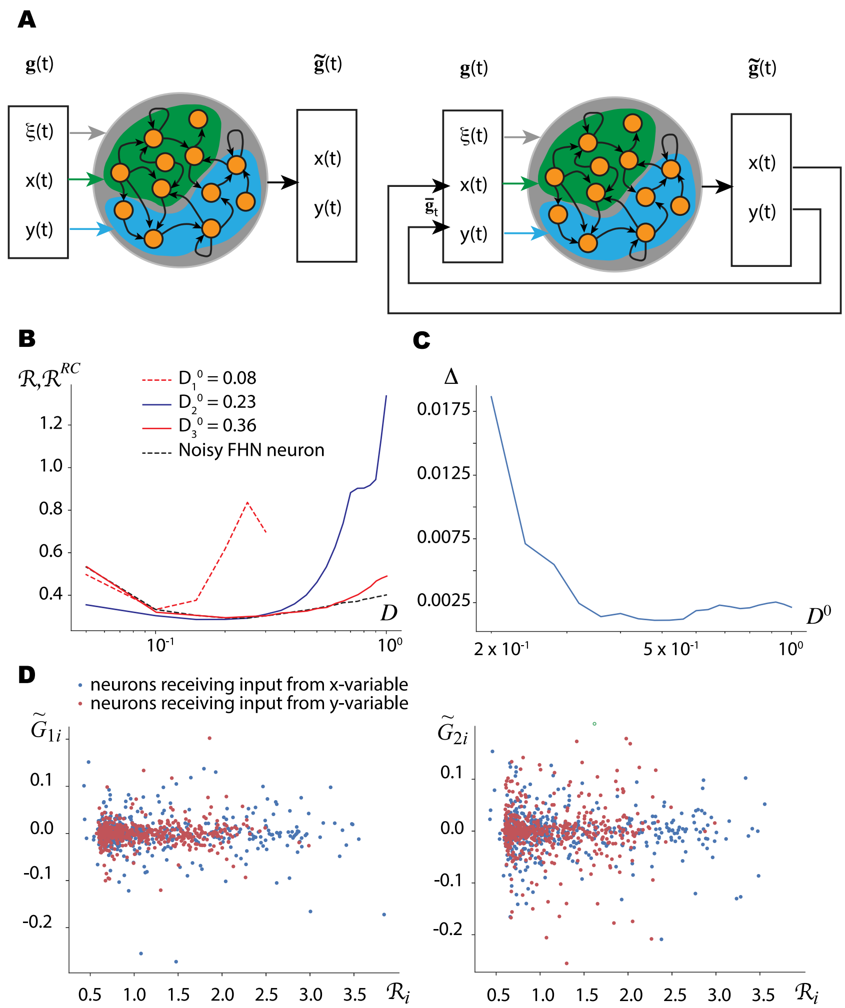

Fig. 7. (**A**) Schematic presentation of the reservoir computer with partially separated inputs both in training and testing modes. (**B**) Coefficient of variations of the spike train versus noise intensity D for the stochastic neuron, $\mathcal{R}$, (black dashed line) and the reservoir computer, $\mathcal{R}^{RC}$, trained for three values of noise intensity $D_1^0 = 0.08$ (red dashed line), $D_1^0 = 0.23$ (solid black line), and $D_1^0 = 0.36$ (red solid line) [in a semi-logarithmic scale]. (**C**) Prediction performance Δ of the reservoir computer with partially separated inputs versus the noise intensity D^0 at which the reservoir was trained. (**D**) Representation of all neurons of hidden layer of reservoir computer with partially separated inputs $[i = 1, \dots N_h]$ in the spaces: coefficient of variation $\mathcal{R}_i$ — coefficient of the output layer matrix $\tilde{G}_{1,i}$ or $\tilde{G}_{2,i}$. Blue points with $i = 1, \dots, N_h/2$ correspond to neurons receiving input from x-variable, and red points $[i = 2N_h + 1, \dots, N_h]$ — neurons receiving input from y-variable. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4. Conclusion

In this study, we have explored the potential of utilizing a reservoir computer to predict the dynamics of a stochastic FitzHugh–Nagumo neuron model under the influence of external noise. We demonstrated that a specially designed reservoir computer, trained with a single noise intensity value, can effectively predict the behavior of the stochastic system across a wide range of noise intensity variations. The reservoir computer accurately reproduces the phenomenon of coherence resonance in the stochastic FHN neuron, indicating the high-quality modeling capabilities of reservoir computing. This achievement was made possible by the unique reservoir design, which separates the artificial neurons in the hidden layer based on the information they receive from the input.

Our analysis of the microscopic dynamics within the hidden layer of the reservoir revealed the formation of distinct clusters of neurons, each exhibiting different dynamics. Some neurons are responsible for replicating the dynamical properties of the neuron, primarily the spike generation mechanism, while the remaining neurons in the reservoir can be viewed as simulating stochastic influences that accurately reproduce the moments of spike generation in the neuron under the influence of noise. This configuration proves to be effective over a broad range of noise control parameters for replicating the essential characteristics of the original stochastic FHN neuron.

Future work may involve fine-tuning the parameters of the reservoir computer, including various hyperparameters of the computational framework, and conducting theoretical analyses to optimize connections within and between reservoir layers. This approach can be extended to various stochastic systems and could find applications in predicting the dynamics of neuronal networks under stochastic influences. The framework we have developed provides a practical means to leverage reservoir computing for the analysis, classification, and prediction of stochastic processes, which are prevalent in the natural world alongside deterministic processes.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Data availability

No data was used for the research described in the article.

Acknowledgments

This work was supported by Ministry of Education and Science of Russian Federation in the framework of PRIORITY-2030 Program.

References

[1] Ott E. Chaos in dynamical systems. Second ed.. Cambridge: Cambridge University Press; 2002.
[2] Boccaletti S, Pisarchik AN, del Genio CI, Amann A. Synchronization: from coupled systems to complex networks. Cambridge: Cambridge University Press; 2018.
[3] Pisarchik AN, Hramov AE. Multistability in physical and living systems: characterization and applications. Cham: Springer Nature; 2022.
[4] Balenzuela P, Rué P, Boccaletti S, Garcia-Ojalvo J. Collective stochastic coherence and synchronizability in weighted scale-free networks. New J Phys 2014;16(1):013036.
[5] Pikovsky AS, Kurths J. Coherence resonance in a noise-driven excitable system. Phys Rev Lett 1997;78(5):775.
[6] Pisarchik AN, Hramov AE. Coherence resonance in neural networks: Theory and experiments. Phys Rep 2023;1000:1–57.
[7] Baspinar E, Schülen L, Olmi S, Zakharova A. Coherence resonance in neuronal populations: Mean-field versus network model. Phys Rev E 2021;103(3):032308.
[8] Andreev AV, Makarov VV, Runnova AE, Pisarchik AN, Hramov AE. Coherence resonance in stimulated neuronal network. Chaos Solitons Fractals 2018;106:80–5.
[9] Collins JJ, Priplata AA, Gravelle DC, Niemi J, Harry J, Lipsitz LA. Noise-enhanced human sensorimotor function. IEEE Eng Med Biol Mag 2003;22(2):76–83.
[10] Priplata A, Niemi J, Salen M, Harry J, Lipsitz LA, Collins J. Noise-enhanced human balance control. Phys Rev Lett 2002;89(23):238101.
[11] Pisarchik AN, Maksimenko VA, Andreev AV, Frolov NS, Makarov VV, Zhuravlev MO, et al. Coherent resonance in the distributed cortical network during sensory information processing. Sci Rep 2019;9(1):18325.
[12] Jaeger H. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. In: Bonn, germany: german national research center for information technology GMD technical report, vol. 148, no. 34. Bonn; 2001, p. 13.
[13] Nakajima K, Fischer I. Reservoir computing. Springer; 2021.
[14] Natschläger T, Maass W. Spiking neurons and the induction of finite state machines. Theor Comput Sci 2002;287(1):251–65.
[15] Carroll TL. Using reservoir computers to distinguish chaotic signals. Phys Rev E 2018;98(5):052209.
[16] Shahi S, Fenton FH, Cherry EM. Prediction of chaotic time series using recurrent neural networks and reservoir computing techniques: A comparative study. Mach Learn Appl 2022;8:100300.
[17] Pathak J, Lu Z, Hunt BR, Girvan M, Ott E. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. Chaos 2017;27(12).
[18] Saha S, Mishra A, Ghosh S, Dana SK, Hens C. Predicting bursting in a complete graph of mixed population through reservoir computing. Phys Rev Res 2020;2(3):033338.
[19] Roy M, Senapati A, Poria S, Mishra A, Hens C. Role of assortativity in predicting burst synchronization using echo state network. Phys Rev E 2022;105(6):064205.
[20] Zimmermann RS, Parlitz U. Observing spatio-temporal dynamics of excitable media using reservoir computing. Chaos 2018;28(4).
[21] Andreev AV, Badarin AA, Maximenko VA, Hramov AE. Forecasting macroscopic dynamics in adaptive Kuramoto network using reservoir computing. Chaos 2022;32(10).
[22] Roy M, Mandal S, Hens C, Prasad A, Kuznetsov N, Dev Shrimali M. Model-free prediction of multistability using echo state network. Chaos 2022;32(10).
[23] Harlim J, Jiang SW, Liang S, Yang H. Machine learning for prediction with missing dynamics. J Comput Phys 2021;428:109922.
[24] Gottwald GA, Reich S. Supervised learning from noisy observations: Combining machine-learning techniques with data assimilation. Physica D 2021;423:132911.
[25] Levine M, Stuart A. A framework for machine learning of model error in dynamical systems. Commun Am Math Soc 2022;2(07):283–344.
[26] Grigoryeva L, Henriques J, Larger L, Ortega J-P. Stochastic nonlinear time series forecasting using time-delay reservoir computers: Performance and universality. Neural Netw 2014;55:59–71.
[27] Fang C, Lu Y, Gao T, Duan J. Reservoir computing with error correction: Long-term behaviors of stochastic dynamical systems. 2023, arXiv preprint arXiv:2305.00669.
[28] Liao Z, Wang Z, Yamahara H, Tabata H. Low-power-consumption physical reservoir computing model based on overdamped bistable stochastic resonance system. Neurocomputing 2022;468:137–47.
[29] Fan H, Kong L-W, Lai Y-C, Wang X. Anticipating synchronization with machine learning. Phys Rev Res 2021;3(2):023237.
[30] Xiao R, Kong L-W, Sun Z-K, Lai Y-C. Predicting amplitude death with machine learning. Phys Rev E 2021;104(1):014205.
[31] Kim JZ, Lu Z, Nozari E, Pappas GJ, Bassett DS. Teaching recurrent neural networks to infer global temporal structure from local examples. Nat Mach Intell 2021;3(4):316–23.
[32] Izhikevich EM. Dynamical systems in neuroscience. MIT Press; 2007.
[33] Muratov CB, Vanden-Eijnden E, Weinan E. Self-induced stochastic resonance in excitable systems. Physica D 2005;210(3–4):227–40.
[34] Lukoševičius M, Jaeger H, Schrauwen B. Reservoir computing trends. KI-Künstliche Intell 2012;26:365–71.
[35] Vlachas PR, Pathak J, Hunt BR, Sapsis TP, Girvan M, Ott E, et al. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. Neural Netw 2020;126:191–217.
[36] McDonald GC. Ridge regression. Wiley Interdiscip Rev Comput Stat 2009;1(1):93–100.
[37] Maslennikov OV, Gao C, Nekorkin VI. Internal dynamics of recurrent neural networks trained to generate complex spatiotemporal patterns. Chaos 2023;33(9).
[38] Maksimenko VA, Lüttjohann A, Makarov VV, Goremyko MV, Koronovskii AA, Nedaivozov V, et al. Macroscopic and microscopic spectral properties of brain networks during local and global synchronization. Phys Rev E 2017;96(1):012316.