


Physics-based approach to developing physical reservoir computers

Vahideh Shirmohammadli and Behraad Bahreyni ^{*}

Faculty of Applied Sciences, Simon Fraser University, British Columbia, V3T 0A3 Canada



(Received 11 February 2024; accepted 21 June 2024; published 11 July 2024)

Reservoir computing leverages the dynamic properties of a fixed, randomly connected neural network to facilitate simplified training and enhanced computational efficiency. Many forms of physical reservoir computers have been proposed. In this paper, we use a three-dimensional (3D)-printed reservoir computer as the design environment, develop analytic models to describe its performance, and validate the models through simulations. This approach offers practical insights for designing physical reservoirs with targeted computational capabilities and enables the assessment of the influence of reservoir parameters such as scale or material choice, on performance metrics, including speed and power consumption. Additionally, the proposed approach may be employed to optimally design physical reservoir computers to solve specific problems. This work contributes to the understanding of physical RC systems by providing a detailed analysis of the physical basis that connects computational performance with multidomain physical interactions at the device level. The methods and results from this work not only propel the development of future 3D-printed physical RC systems but also serves as a framework for evaluating and designing diverse physical RC models based on other approaches.

DOI: [10.1103/PhysRevResearch.6.033055](https://doi.org/10.1103/PhysRevResearch.6.033055)

I. INTRODUCTION

Reservoir computing has emerged as a transformative approach in artificial intelligence, characterized by leveraging the rich dynamics of a fixed, randomly connected neural network, termed the *reservoir*. Reservoir computing (RC) simplifies the training process, enables efficient computation, and provides insights into how information processing can be achieved in complex systems. The reliance of RC on a fixed reservoir has motivated the research on developing reservoir computers based on physical systems [1–3]. Over the past decade, physical reservoir computing has garnered considerable attention as an alternative to commonplace machine learning techniques. Diverging from conventional approaches that rely on digital systems, physical reservoir computing adopts a distinct methodology where the intrinsic characteristics of physical systems and their responses are utilized in lieu of complex digital computations [4–7]. At its essence, physical RC capitalizes on the inherent dynamics of these physical systems to nonlinearly transform input data into a high-dimensional feature space where the system dynamics provides a fading memory of recent events. Context is often determined through a weighted sum of the responses of the coupled nonlinear elements within the reservoir. Training is achieved using simple and robust algorithms, such as linear regression, to determine the weights for the readout layer [8]. In physical computing, specifically, training may be achieved

in ways similar to digital computers, where internal coupling between the computation nodes is adjusted based on training data [9]. Within the realm of physical RC, various physical systems, including electronic circuits [10], lasers [11], mechanical systems [12–14], or photonic systems [15–17], have been exploited to create a reservoir—a dynamic platform where complex calculations take place.

One significant advantage of physical RC lies in the integration of machine learning directly with sensors [18]. Two application cases are possible within this realm: (1) near-sensor computing, where the quantified information from a sensor is processed by a contextual processor that is integrated near the sensor, and (2) in-sensor processing, where the physical characteristics of sensors are used to carry out the computations needed for generating context. Within this framework, an integrated physical computer can provide several advantages over postprocessing of signals in digital domain, including reducing system complexity (and number of components), miniaturization, compatible manufacturing, and potentially lower power consumption and higher speed, among others [19]. However, the challenges will be the same as with other analog computers, such as a lack of flexibility (i.e., updating) for the core network (the output layer weights can be updated) and a potentially more challenging design process.

The research on physical RC has led to numerous investigations into both its theoretical foundations and practical applications [3, 20–22]. Physical RC has proven to be effective in tackling a wide range of problems, including speech recognition, image processing, and controlling nonlinear systems [23–25]. Its potential extends to the development of the next generation of edge processors, which could operate within sensing networks, enabling energy-efficient and real-time information processing capabilities [18]. Nonlinear behaviors in physical systems often arise from the properties of

^{*}Contact author: behraad_bahreyni@sfu.ca

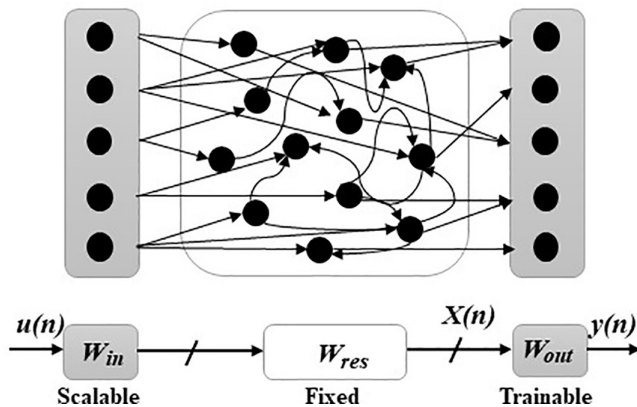


FIG. 1. A representation of echo state networks.

materials or devices, and the concept of fading memory can be linked to the relaxation time of these components within the system in response to external stimuli [26,27]. However, the dynamic behavior of devices can deviate from that of a single component when dealing with complex networks of connections that interconnect computational elements. While many physical RC systems use electrical connections for information exchange, there is also potential to employ other types of physical coupling, such as thermal coupling in integrated reservoirs that incorporate temperature-sensitive resistors (i.e., thermistors).

Despite the growing interest in physical RC systems, most of the presented work to date has focused on utilizing physical systems with desired responses in order to perform the operations without much attention to the underlying physics. However, the physics governing the operation of the reservoirs, and particularly, methods to design and optimize their operations, have remained understudied. This is partly due to the vast variations between the implementations of physical RC, but it is also because, like many new exploration areas, the focus has been on the demonstration of the utility of RC principles through different media.

This paper builds upon our earlier research involving three-dimensional (3D)-printed computers, where we reported the development of a physical RC based on common 3D printed medium [28]. In this work, we provide the physical basis required to bridge the gap between computational performance of the final system and the multidomain physical interactions. Results of this work can be readily applied to *design* physical reservoirs with a desired level of computational capability or help evaluate the effect of different reservoir parameters, such as its scale, on various performance parameters, such as speed or power consumption. Our approach presents the insight to not only develop future 3D-printed physical RC but also a roadmap for design and evaluation of other physical RC.

II. MATERIALS AND METHODS

Most of the digital implementations of RC are based on echo state networks (ESNs), a particular type of recurrent neural network (RNN), as shown in Fig. 1 [29,30]. ESNs utilize an input layer that scales temporal input and applies it to a fixed, randomly generated reservoir of nonlinear neurons

with a fading memory of the input history and fixed internal weights, represented as W_{res} , to perform computations. The reservoir state at each point of time is therefore determined by a time-dependent nonlinear mapping of the input signals. A trainable readout layer W_{out} , is trained and reconfigured for various temporal applications [31]. The operation of an RC is described through

$$\begin{aligned} \mathbf{x}(n) &= (1 - \alpha)\mathbf{x}(n - 1) + \alpha f[W_{in} \mathbf{u}(n) + W_{res}\mathbf{x}(n - 1)], \\ \mathbf{y}(n) &= W_{out} \mathbf{x}(n), \end{aligned} \quad (1)$$

where $\mathbf{x}(n)$ and $\mathbf{x}(n-1)$ are the reservoir states' vectors in the current and previous time steps, $\mathbf{u}(n)$ and $\mathbf{y}(n)$ represent the input and predicted output at time step n , f is the nonlinear mapping function, and α is the leaking rate. It is important to note that in digital or software implementations, the weight matrix that governs the behavior of the reservoir remains time independent, just like in the case of feedforward and recurrent neural networks. The fading memory requirement in digital hardware RC is implemented through cascaded logic gates, adders, and/or flip flops in a cyclic topology [32].

In physical implementation of RC, the weights between the neurons are determined by their coupling and energy exchange between them. Therefore, physical RC differs in its behavior from digital RC as the weight matrices for physical RC are often time dependent due to the dynamics of the physical systems. The weights may also exhibit nonlinearity due to the response of the materials or physical components in the reservoir. The time-dependent weight matrix allows the physical system to adapt dynamically to input signals, enhancing its computational power, which can be utilized to expand the capabilities of physical RC. However, proper modeling of such intricate issues remains a challenge that we try to address using our 3D printed RC as a tool.

A. Physical model for electrothermal RC

Resistors made of materials with temperature-dependent resistivity are promising candidates to be used as neurons in physical RC. These devices exhibit nonlinearity when self-heated by passing a current through them. Moreover, nearby devices may exchange energy in electrical or thermal domains. We demonstrated two such implementations using off-the-shelf thermistors [33] and 3D printed RC [28]. For the 3D-printed RC, for instance, resistors made from conductive PLA (i.e., a composite of carbon black and polylactic acid) exhibit a highly nonlinear temperature-dependent resistivity [28]. This results in a nonlinear current-voltage response for such devices where their self-heating causes a change in their temperature, and thus, their resistance varies nonlinearly. As the temperature rises, the polymer matrix of the material softens, and the carbon particles may agglomerate or settle, increasing resistance. Moreover, since heat-dissipation is a relatively slow process, these devices will retain information in thermal domain and exhibit complex behavior such as hysteresis. Figure 2 presents a schematic of a 3D-printed resistor along with its nonlinear behavior observed during a current sweep. Each current value was injected for 10 s. This setup ensures that the thermistor heats up adequately within each current value, exhibiting nonlinear behavior and showcasing hysteresis. This phenomenon highlights the presence of

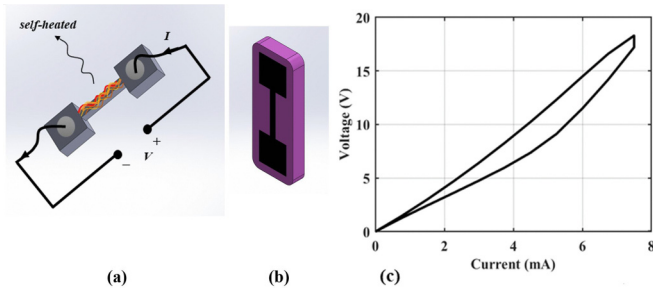


FIG. 2. (a) A schematic of a 3D-printed resistor using conductive PLA, where the nonlinearity comes from nonlinear, temperature-dependent resistivity in a neuron. (b) The 3D model of the resistor encapsulated inside the regular PLA. (c) The nonlinear behavior in the response of the 3D-printed resistor.

sufficient thermal memory within the system while behaving nonlinearly.

Such neurons can be 3D printed with the desired spatial distribution limited by the printing process, allowing for their coupling through heat exchange. Therefore, for 3D-printed reservoirs, the weight matrix can be derived by understanding the thermal coupling among the resistors. The temperature distribution in the medium encompassing the resistors changes in response to the self-heating of the resistors, as neurons heat up when current passes through them (*heat generation*) and heat is distributed within the surroundings (*heat transfer*). The amount of heat transfer and temperature distribution depends on the *boundary conditions* of the resistors within the reservoir.

B. Static thermal response of RC

An analytic model for the static response of a reservoir comprising four 3D-printed resistors is developed first. This

structure models resistors from electrically conductive PLA embedded in/on an electrically insulating substrate of regular PLA. Four such resistors, as illustrated in Fig. 3, are considered in this model that represent different potential combinations in a 3D printed reservoir. To simplify the analysis, it is assumed that each resistor takes the form of a long cylinder with a radius r_i . The primary objective here is to study the heat transfer within the medium surrounding these resistors. Furthermore, we assume that the temperature of the resistors does not change significantly across its cross section, so the average temperature for each resistor is equal to the temperature at its contact surface with the regular PLA, i.e., $T(r_i)$ (see the Supplemental Material [34]). Each resistor receives an input signal in the form of current density, J (in $\frac{A}{m^2}$), and the response of these neurons is quantified in terms of its temperature (in K). It is important to note that these neurons are not electrically connected. For the static analysis of heat distribution, the electrical conductivity of the neurons is considered to be insensitive to temperature, which does not affect the results. The focus is to gain insight into the thermal coupling between these neurons, including their strength and the factors that influence the strength of these connections.

The first step involves deriving the temperature distribution inside a regular PLA piece resulting from the heating of the resistors. This requires solving the heat transfer problem within the PLA to understand how heat spreads and impacts the overall temperature distribution. In the subsequent stage, the temperature at the location of each neuron (or resistor) is calculated, taking into consideration the influence of all the other neurons. Since we assume that the medium behaves linearly, we can apply the principle of superposition by calculating the temperature distribution from each heated resistor on its own and summing up the results after repeating the analysis for all resistors.

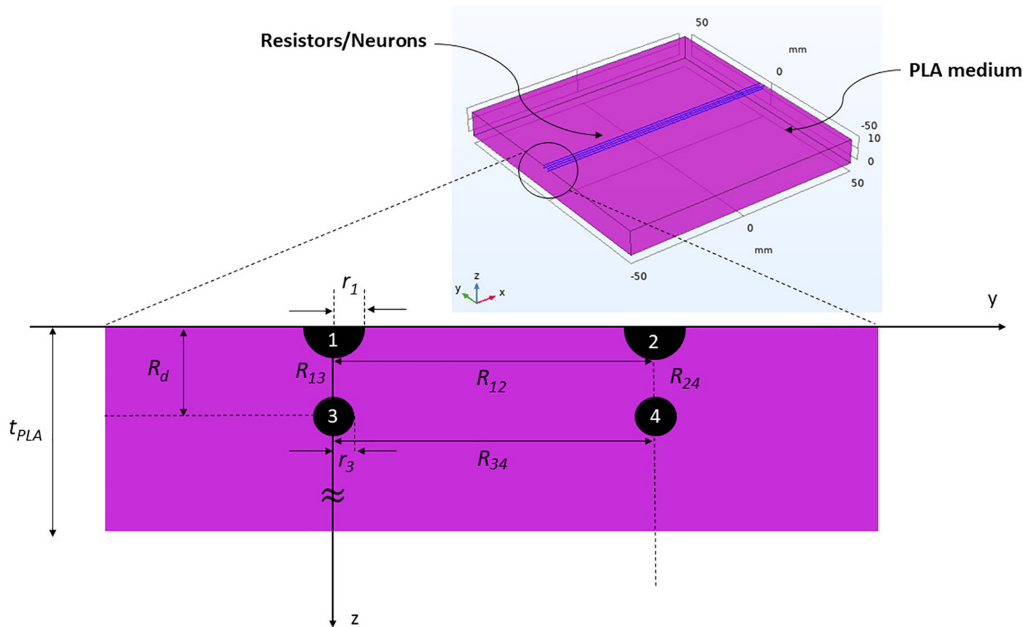


FIG. 3. The heat transfer problem with four resistors/neurons. Resistors are long cylinders with extremely small radiuses compared to medium thicknesses.

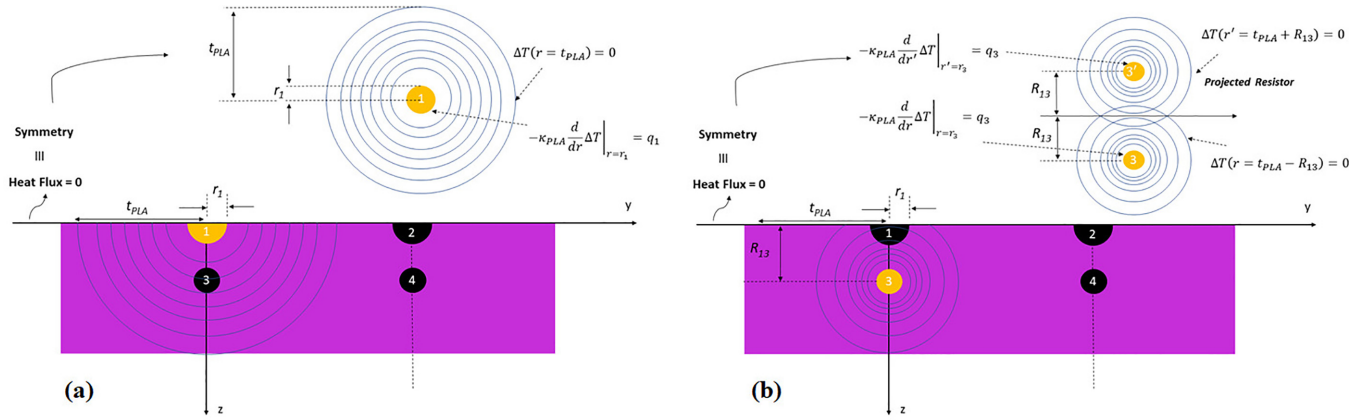


FIG. 4. (a) Simplification of heat transfer for resistor/neuron 1. It is assumed that the heat transfer ceases at the top surface of the PLA, where it contacts the surrounding air, allowing application of a thermal isolation boundary condition. Thermal isolation at a specific point can be caused due to the system's symmetry. Therefore, instead of solving a half-cylindrical problem, the general heat conduction problem for a fully cylindrical structure would result in the temperature distribution due to the self-heating of resistor 1. (b) Simplification of heat transfer for resistor/neuron 3. To accurately represent the thermal isolation boundary condition at the top surface in contact with the air, we can consider a projection of resistor 3 along the negative z axis.

To obtain the temperature distribution, the general heat conduction problem using a cylindrical coordinate system needs to be solved [35]:

$$\frac{1}{r} \frac{\partial}{\partial r} \left(\kappa_{\text{PLA}} r \frac{\partial T}{\partial r} \right) + \frac{1}{r^2} \frac{\partial}{\partial \varphi} \left(\kappa_{\text{PLA}} r \frac{\partial T}{\partial \varphi} \right) + \frac{\partial}{\partial z} \left(\kappa_{\text{PLA}} \frac{\partial T}{\partial z} \right) + \dot{e}_{\text{gen}} = \rho_m C_p \frac{\partial T}{\partial t}, \quad (2)$$

where κ_{PLA} , ρ_m , and C_p represent the thermal conductivity, mass density, and specific heat capacity of the PLA material, respectively. The term \dot{e}_{gen} signifies the heat generated within the PLA piece. As we are considering the static case, the term on the right side of the equations is zero.

It is worth noting that while heat is indeed generated within the resistors, the internal resistor volumes are excluded from these equations, and instead a specific heat flux boundary condition is applied at the interface between the resistor and the substrate. Mathematically, this boundary condition is expressed as $-\kappa_{\text{PLA}} \frac{dT}{dr} = q$, where q represents the heat flux at the point where the neuron contacts the PLA and the heat generated in the resistor is conducted away from the contact surface. The heat flux for each resistor with a radius of r_i is determined as $q_i = \rho_e J_i^2 r_i$, with ρ_e representing the electrical resistivity (see the Supplemental Material [34]). Additionally, it is assumed that the substrate under study has substantial dimensions compared to the resistors such that at a location far from the resistors, the temperature reaches the ambient temperature, denoted as T_a . This assumption allows for the application of a constant temperature boundary condition, specifically, $T(r = t_{\text{PLA}}) = T_a$. Furthermore, assuming negligible heat transfer to the ambient through the top surface of the substrate, a thermal isolation boundary condition for this surface may be used. This assumption simplifies the solution as the substrate may be assumed to be the symmetric copy of itself across the top surface. Therefore, instead of solving a half-cylindrical problem, we can address the general heat conduction problem for a fully cylindrical structure, as depicted

in Fig. 4(a). This simplifies the problem to

$$\frac{1}{r} \frac{d}{dr} \left(\kappa_{\text{PLA}} r \frac{dT}{dr} \right) = 0. \quad (3)$$

This equation needs to be solved for two separate cases of resistors that are printed on the surface and those that are embedded within the substrate. Solving this equation for resistor 1, considering the condition $-\kappa_{\text{PLA}} \frac{dT}{dr} \Big|_{r=r_1} = q_1$, yields the temperature distribution for the region within the PLA where $r_1 < r < t_{\text{PLA}}$ as given by

$$\Delta T_1(r) = \frac{\rho_e J_1^2 r_1^2}{\kappa_{\text{PLA}}} \ln \left(\frac{t_{\text{PLA}}}{r} \right). \quad (4)$$

Here, $\Delta T_i(r) = T_i(r) - T_a$, which represents the relative temperature of the resistors with respect to the ambient temperature. To obtain an approximate temperature of resistor #1, one can substitute $r = r_1$ into Eq. (4) and add the ambient temperature. It is important to highlight that the area through which heat transfers from resistor 3 and its transfer to the surrounding area differ from the previous case. For resistor 3, a thermal isolation boundary condition is applied at the top surface in contact with the air. This boundary condition is accurately represented by considering a projection of resistor 3 along the negative z axis, as depicted in Fig. 4(b). As a result, the heat flux boundary condition for resistor 3 and its projection is adjusted, $-\kappa_{\text{PLA}} \frac{dT}{dr} \Big|_{r=r_3} = \rho_e J_3^2 r_3$ and $-\kappa_{\text{PLA}} \frac{dT}{dr'} \Big|_{r'=r_3} = \rho_e J_3^2 r_3$. Solving Eq. (3) for the region where $r_3 < r, r' < t_{\text{PLA}} - R_d$ yields

$$\Delta T_3(r) = \frac{\rho_e J_3^2 r_3^2}{\kappa_{\text{PLA}}} \ln \left(\frac{t_{\text{PLA}}^2 - R_d^2}{r r'} \right), \quad (5)$$

where R_d represents the distance of resistor 3 from the top surface, and r and r' represent the coordinate system centered at the midpoint of resistor 3 and its projection, respectively.

Now, it is simpler to use the Cartesian coordinate system for the temperature distribution for the remaining resistors. We maintain the origin of the Cartesian coordinate system at the center of resistor 1. This leads to the following substitutions in the equations:

(i) By replacing $r = \sqrt{y^2 + z^2}$ in Eq. (4), the temperature distribution due to self-heating of resistor 1 for the region where $r_1 < \sqrt{y^2 + z^2} < t_{\text{PLA}}$ is determined as

$$\Delta T_1(y, z) = \frac{\rho_e J_1^2 r_1^2}{k_{\text{PLA}}} \ln\left(\frac{t_{\text{PLA}}}{\sqrt{y^2 + z^2}}\right). \quad (6)$$

(ii) By replacing $r = \sqrt{y^2 + (z - R_d)^2}$ and $r' = \sqrt{y^2 + (z + R_d)^2}$ in Eq. (5), the temperature distribution due to self-heating of resistor 3 for the region where $r_3 < \sqrt{y^2 + (z - R_d)^2} < t_{\text{PLA}} - R_d$ is determined as

$$\Delta T_3(y, z) = \frac{\rho_e J_3^2 r_3^2}{k_{\text{PLA}}} \ln\left(\frac{t_{\text{PLA}}^2 - R_d^2}{\sqrt{(y^2 + (z - R_d)^2)(y^2 + (z + R_d)^2)}}\right). \quad (7)$$

Now that the problem has been solved for two types of temperature distributions within the material—neurons located on the top layer and those buried at any depth from the top surface—the temperature distribution for the remaining resistors/neurons in the reservoir can be described as follows:

(iii) By replacing $y = y - R_{12}$ in Eq. (8), the temperature distribution due to the self-heating of resistor 2 for the region where $r_2 < \sqrt{(y - R_{12})^2 + z^2} < t_{\text{PLA}}$ is

$$\Delta T_2(y, z) = \frac{\rho_e J_2^2 r_2^2}{k_{\text{PLA}}} \ln\left(\frac{t_{\text{PLA}}}{\sqrt{(y - R_{12})^2 + z^2}}\right). \quad (8)$$

$$W_{\text{res}} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ W_{31} & W_{32} & W_{33} & W_{34} \\ W_{41} & W_{42} & W_{43} & W_{44} \end{bmatrix} = \frac{\rho_e}{k_{\text{PLA}}} = \begin{bmatrix} r_1^2 \ln\left(\frac{t_{\text{PLA}}}{r_1}\right) & r_2^2 \ln\left(\frac{t_{\text{PLA}}}{R_{12}}\right) & r_3^2 \ln\left(\frac{t_{\text{PLA}} - R_d^2}{R_{13}^2}\right) & r_4^2 \ln\left(\frac{t_{\text{PLA}} - R_d^2}{R_{14}^2}\right) \\ r_1^2 \ln\left(\frac{t_{\text{PLA}}}{R_{12}}\right) & r_2^2 \ln\left(\frac{t_{\text{PLA}}}{r_2}\right) & r_3^2 \ln\left(\frac{t_{\text{PLA}} - R_d^2}{R_{23}^2}\right) & r_4^2 \ln\left(\frac{t_{\text{PLA}} - R_d^2}{R_{24}^2}\right) \\ r_1^2 \ln\left(\frac{t_{\text{PLA}}}{R_{13}}\right) & r_2^2 \ln\left(\frac{t_{\text{PLA}}}{R_{23}}\right) & r_3^2 \ln\left(\frac{t_{\text{PLA}} - R_d^2}{2r_3 R_d}\right) & r_4^2 \ln\left(\frac{t_{\text{PLA}} - R_d^2}{R_{34} R_d}\right) \\ r_1^2 \ln\left(\frac{t_{\text{PLA}}}{R_{14}}\right) & r_2^2 \ln\left(\frac{t_{\text{PLA}}}{R_{24}}\right) & r_3^2 \ln\left(\frac{t_{\text{PLA}} - R_d^2}{R_{34} R_d}\right) & r_4^2 \ln\left(\frac{t_{\text{PLA}} - R_d^2}{2r_4 R_d}\right) \end{bmatrix} \quad (12)$$

The unit of the weight matrix is $\frac{\Omega \cdot K}{W} m^4$. The weight matrix is contingent upon the spacing between the resistors/neurons, represented as R_{ij} . As may be seen, the coupling between two resistors is strongly influenced by the inverse of the distance between them. Another important factor that affects the coupling between the neurons is the thermal conductivity of the substrate, k_{PLA} . Last, it is the resistivity of the resistors/neurons, ρ_e , that influences the weights. A temperature-independent resistivity yields a linear reservoir matrix, while nonlinear temperature-dependent resistivity gives rise to nonlinear interactions among the resistors/neurons (not modeled in the above). Furthermore, a key to have a rich reservoir (diverse weight elements) is to design distinct resistors/neurons with varying radii. Indeed, in a 3D-printed reservoir, the resolution of the printer serves as a key determinant of the achievable coupling weight resolution, as it dictates the physical dimensions governing thermal coupling. While this paper primarily focuses on the

(iv) By replacing $y = y - R_{34}$ in Eq. (9), the temperature distribution due to the self-heating of resistor 4 for the region where $r_4 < \sqrt{(y - R_{34})^2 + (z - R_d)^2} < t_{\text{PLA}} - R_d$ is given as

$$\Delta T_4(y, z) = \frac{\rho_e J_4^2 r_4^2}{k_{\text{PLA}}} \left[\ln\left(\frac{t_{\text{PLA}}^2 - R_d^2}{\sqrt{(y - R_{12})^2 + (z - R_d)^2}}\right) + \ln\left(\frac{t_{\text{PLA}}^2 - R_d^2}{\sqrt{(y - R_{34})^2 + (z + R_d)^2}}\right) \right]. \quad (9)$$

The principle of superposition then implies the overall temperature distribution as

$$\Delta T(y, z) = \sum_{i=1}^4 \Delta T_i(y, z). \quad (10)$$

Now, with the equation described above, the temperature of each resistor/neuron by incorporating their respective locations into the equation and considering all sources of heat generation results in the static model of the reservoir as

$$\begin{bmatrix} \Delta T_1 \\ \Delta T_2 \\ \Delta T_3 \\ \Delta T_4 \end{bmatrix} = W_{\text{res}}^{(4 \times 4)} \begin{bmatrix} J_1^2 \\ J_2^2 \\ J_3^2 \\ J_4^2 \end{bmatrix}. \quad (11)$$

Here ΔT is the vector describing the neuron responses and J^2 represents the input signal. W_{res} is the static weight matrix of the physical reservoir. Considering $r_i \ll R_{ij}$, where $i \neq j$, this weight matrix is given as

physical dimensions determining thermal coupling, it is essential to acknowledge the role of potential random electrical connections in enhancing the richness of the weight matrix. These connections introduce an additional degree of freedom in the design, allowing for greater flexibility in generating diverse weight matrices. Although this aspect has not been extensively discussed in this paper, it is an important consideration that can significantly influence the performance of the reservoir.

C. Dynamic thermal response of RC

Understanding how a reservoir reacts to time-series data is crucial. The transient heat transfer analysis helps model and analyze the dynamic behavior of a reservoir comprising thermally connected neurons. Self-heating resistors exhibit a transient response characterized by two thermal time constants: the internal time constant (τ_{int}) and the substrate time

constant (τ_{sub}). The internal time constant represents the duration it takes for the heat generated inside the resistors to stabilize inside the resistor volume. The substrate time constant signifies how fast the heat is conducted away from the resistors' contact surface into the substrate. The thermal time constants rely on the physical properties of the thermistor and substrate, including their dimensions, thermal mass, and thermal conductivity. In a reservoir with effectively large dimensions for the substrate compared to the resistors, the substrate time constant would dominate the response time and determine the transient behavior of the neurons and its design. Therefore, understanding factors that affect τ_{sub} are important. We introduce thermal diffusivity for the PLA as $\alpha_{\text{th}} = \frac{k_{\text{PLA}}}{\rho_m C_p}$. If we assume that temperature changes mostly in the radial direction (r), and we express temperature as a product of radial and temporal components, i.e., $T(r, t) = R(r) T(t)$, solving Eq. (2) by focusing on the time-dependent part gives us

$$\begin{aligned} T(t) &= T_f + (T_i - T_f)e^{-\alpha_{\text{th}}\lambda^2 t}, \\ R(r) &= B_1 \text{Bessel}J(0, \lambda r), \end{aligned} \quad (13)$$

where B_1 is an amplitude coefficient, $\tau = 1/\alpha_{\text{th}}\lambda^2$ is the thermal time constant, T_i and T_f are the resistor's initial and final temperatures, and $\text{Bessel}J(0, \lambda r)$ is the Bessel function of the first kind of order zero, a common solution to cylindrical heat conduction equations. To determine the value of λ , appropriate boundary conditions need to be applied and the problem solved numerically. A numerical solution for τ_{th} is given as [35]

$$\tau_{\text{th}} = \frac{L_{\text{ch}}^2}{2.4^2 \alpha_{\text{th}}}, \quad (14)$$

where L_{ch} represents the characteristic length of the substrate, defined as the distance from the resistor to the surface of the PLA piece with a constant temperature. For resistors in the top layer, L_{ch} equals t_{PLA} . For resistors/neurons deeper within the PLA, $R_{d,i}$, the characteristic length is $L_{\text{ch}} = \sqrt{(t_{\text{PLA}} - R_{d,i})^2 + (t_{\text{PLA}} + R_{d,i})^2} = \sqrt{t_{\text{PLA}}^2 + R_{d,i}^2}$.

This approach helps estimate thermal time constants, which approximate the reservoir's timescale. This timescale

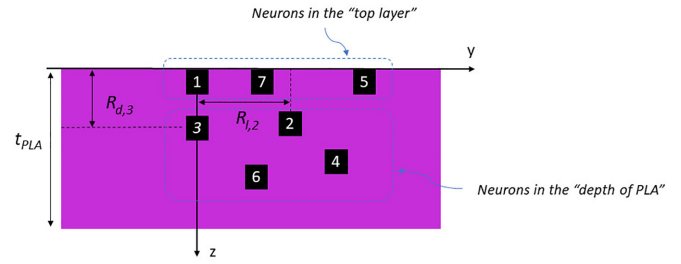


FIG. 5. A sample reservoir with a general, random configuration defining the neurons in the top layer and in-depth of PLA.

determines the length of events the reservoir can detect and, consequently, the frequency at which the reservoir output should be read [28]. Even though Eq. (13) describes the transient response of a reservoir with single resistor, the entire reservoir, comprising multiple interconnected elements, exhibits a distinct dynamical behavior. When all resistors generate heat in a linear medium, the dynamic response of the reservoir can be described as

$$T(t) = \sum_{j=1}^4 T_{f,j} + (T_{i,j} - T_{f,j})e^{-t/\tau_j}. \quad (15)$$

This equation accounts for the combined thermal effects across the different resistors and their respective time constants τ_j . Therefore, the dynamic temperature of the resistors evolves from an initial condition of $\Delta T_i(y, z, 0)$ as

$$\begin{bmatrix} \Delta T_1 \\ \Delta T_2 \\ \Delta T_3 \\ \Delta T_4 \end{bmatrix} = \begin{bmatrix} \Delta T_{i,1} e^{-t/\tau_1} \\ \Delta T_{i,2} e^{-t/\tau_2} \\ \Delta T_{i,3} e^{-t/\tau_3} \\ \Delta T_{i,4} e^{-t/\tau_4} \end{bmatrix} + W_{\text{res}}(t)^{(4 \times 4)} \begin{bmatrix} J_1^2 \\ J_2^2 \\ J_3^2 \\ J_4^2 \end{bmatrix}. \quad (16)$$

It is important to note that the initial condition is not considered constant, especially in the case of a time-series input signal. Equation (16) accounts for the extent to which the resistors are heated through the introduction of any input signal. This ensures that subsequent time steps in the time series are dependent on the temperatures reached in previous steps. The time-dependent weight matrix $W_{\text{res}}(t)$ is given by

$$W_{\text{res}}(t) = \begin{bmatrix} W_{11}(1 - e^{-t/\tau_1}) & W_{12}(1 - e^{-t/\tau_2}) & W_{13}(1 - e^{-t/\tau_3}) & W_{14}(1 - e^{-t/\tau_4}) \\ W_{21}(1 - e^{-t/\tau_1}) & W_{22}(1 - e^{-t/\tau_2}) & W_{23}(1 - e^{-t/\tau_3}) & W_{24}(1 - e^{-t/\tau_4}) \\ W_{31}(1 - e^{-t/\tau_1}) & W_{32}(1 - e^{-t/\tau_2}) & W_{33}(1 - e^{-t/\tau_3}) & W_{34}(1 - e^{-t/\tau_4}) \\ W_{41}(1 - e^{-t/\tau_1}) & W_{42}(1 - e^{-t/\tau_2}) & W_{43}(1 - e^{-t/\tau_3}) & W_{44}(1 - e^{-t/\tau_4}) \end{bmatrix}. \quad (17)$$

The value of τ_j depends on different neurons and input signal conditions. Importantly, as time (t) approaches infinity, $W_{\text{res}}(t)$ converges to the static weight matrix derived in the previous section, mathematically represented as

$$\lim_{t \rightarrow \infty} W_{\text{res}}(t) = W_{\text{res}}. \quad (18)$$

Despite the reservoir's continuous operation, it usually interacts with a digital system whose output is sampled at specific time intervals Δt . A discrete time representation of the weight matrix is

$$W_{\text{res}, \Delta t} = \begin{bmatrix} W_{11}(1 - e^{-\Delta t/\tau_1}) & W_{12}(1 - e^{-\Delta t/\tau_2}) & W_{13}(1 - e^{-\Delta t/\tau_3}) & W_{14}(1 - e^{-\Delta t/\tau_4}) \\ W_{21}(1 - e^{-\Delta t/\tau_1}) & W_{22}(1 - e^{-\Delta t/\tau_2}) & W_{23}(1 - e^{-\Delta t/\tau_3}) & W_{24}(1 - e^{-\Delta t/\tau_4}) \\ W_{31}(1 - e^{-\Delta t/\tau_1}) & W_{32}(1 - e^{-\Delta t/\tau_2}) & W_{33}(1 - e^{-\Delta t/\tau_3}) & W_{34}(1 - e^{-\Delta t/\tau_4}) \\ W_{41}(1 - e^{-\Delta t/\tau_1}) & W_{42}(1 - e^{-\Delta t/\tau_2}) & W_{43}(1 - e^{-\Delta t/\tau_3}) & W_{44}(1 - e^{-\Delta t/\tau_4}) \end{bmatrix}. \quad (19)$$

TABLE I. Specific dimensions used for the COMSOL Multiphysics simulations.

Parameter	Definition	Value
r_1 and r_2	radius of resistors/neurons in the top layer	$5\sqrt{2}$ (μm)
r_3 and r_4	radius of resistors/neurons in the second layer	5 (μm)
t_{PLA}	thickness of the 3D-printed piece PLA	10 (mm)
W_{PLA}	width of the 3D-printed piece PLA	100 (mm)
L_{PLA}	length of the 3D-printed piece and the resistors/neurons	100 (mm)
R_{12} and R_{34}	shortest spacing between two neighboring resistors/neurons in one layer	2 (mm)
R_{13} and R_{24}	shortest spacing between two neighboring resistors/neurons in two different layers	2 (mm)

after each time step Δt . When Δt is significantly smaller than the thermal time constants ($\Delta t \ll \tau_j$), the weight matrix approaches zero, indicating that the input at the current time step has little influence on the temperature distribution within the reservoir. Consequently, the reservoir retains the previous response, albeit slightly attenuated, and interneuron interactions become negligible. Conversely, when Δt is much larger than the thermal time constants ($\Delta t \gg \tau_j$), the input at the current time step essentially determines the temperature distribution in the system, overriding past events. This analysis highlights the importance of time response considerations with regard to the suitability of a physical reservoir computer to solving problems. For a given physical reservoir, the choice of sampling frequency ($f_s = \frac{1}{\Delta t}$) plays a crucial role in determining its performance.

D. Generalization of the model

The first step in developing a general model is to bridge the gap between theoretical assumptions and real-world conditions. An adjustment is made to the existing model to account for the actual dimensions of the 3D-printed resistors. These resistors have rectangular cross sections, which differ from the cylindrical cross sections assumed in the earlier model (see the Supplemental Material [34]).

A systematic approach involving superposition can be employed to create a comprehensive model accommodating N resistors/neurons within a reservoir. This method enables the derivation of a generalized model capable of addressing diverse configurations and scenarios. Specifically, this model considers a range of neurons within the reservoir, each contributing to the overall temperature distribution. For resistors/neurons situated in the top layer, the temperature distribution in the region where $r_i < \sqrt{(y - R_{l,i})^2 + z^2} < t_{\text{PLA}}$ can be expressed as follows:

$$\Delta T_i(y, z) = \frac{\rho_e J_i^2 r_i^2}{k_{\text{PLA}}} \ln \left(\frac{t_{\text{PLA}}}{\sqrt{(y - R_{l,i})^2 + z^2}} \right). \quad (20)$$

In this equation, $R_{l,i}$ signifies the lateral distance of the neuron from a coordinate system's origin. For resistors/neurons located deeper within the PLA piece, at a distance $R_{d,i}$ from the top surface, the temperature distribution in the region where $r_i < \sqrt{(y - R_{l,i})^2 + (z - R_{d,i})^2} < t_{\text{PLA}} - R_{d,i}$ is determined as

$$\Delta T_i(y, z) = \frac{\rho_e J_i^2 r_i^2}{k_{\text{PLA}}} \left[\ln \left(\frac{\sqrt{t_{\text{PLA}}^2 - R_{d,i}^2}}{\sqrt{(y - R_{l,i})^2 + (z - R_{d,i})^2}} \right) + \ln \left(\frac{\sqrt{t_{\text{PLA}}^2 - R_{d,i}^2}}{\sqrt{(y - R_{l,i})^2 + (z + R_{d,i})^2}} \right) \right]. \quad (21)$$

These definitions are depicted in Fig. 5 for a sample reservoir configuration. Their spatial arrangement and thermal interactions then determine the temperature distribution within the reservoir.

III. RESULTS AND DISCUSSION

Evaluation of the static response. This analysis focused on simulating a reservoir consisting of four resistors/neurons, as illustrated in Fig. 3, with specific dimensions outlined in Table I. For reference, we used specific material properties compiled in Table II. To carry out these simulations, we utilized COMSOL Multiphysics® 5.5 software. Simulating the

TABLE II. Material characteristics.

Material	PLA	Conductive PLA
T_m (K)	60–65	60–70
T_g (K)	180–220	170
ρ_m (kg/m ³)	1210–1240	1220
C_p (J/kg K)	1180–1210	1180
κ (W/m K)	0.12–0.15	0.18–0.2
ϵ_r	2.88–3.48	
ρ_e (Ω cm)		30–115

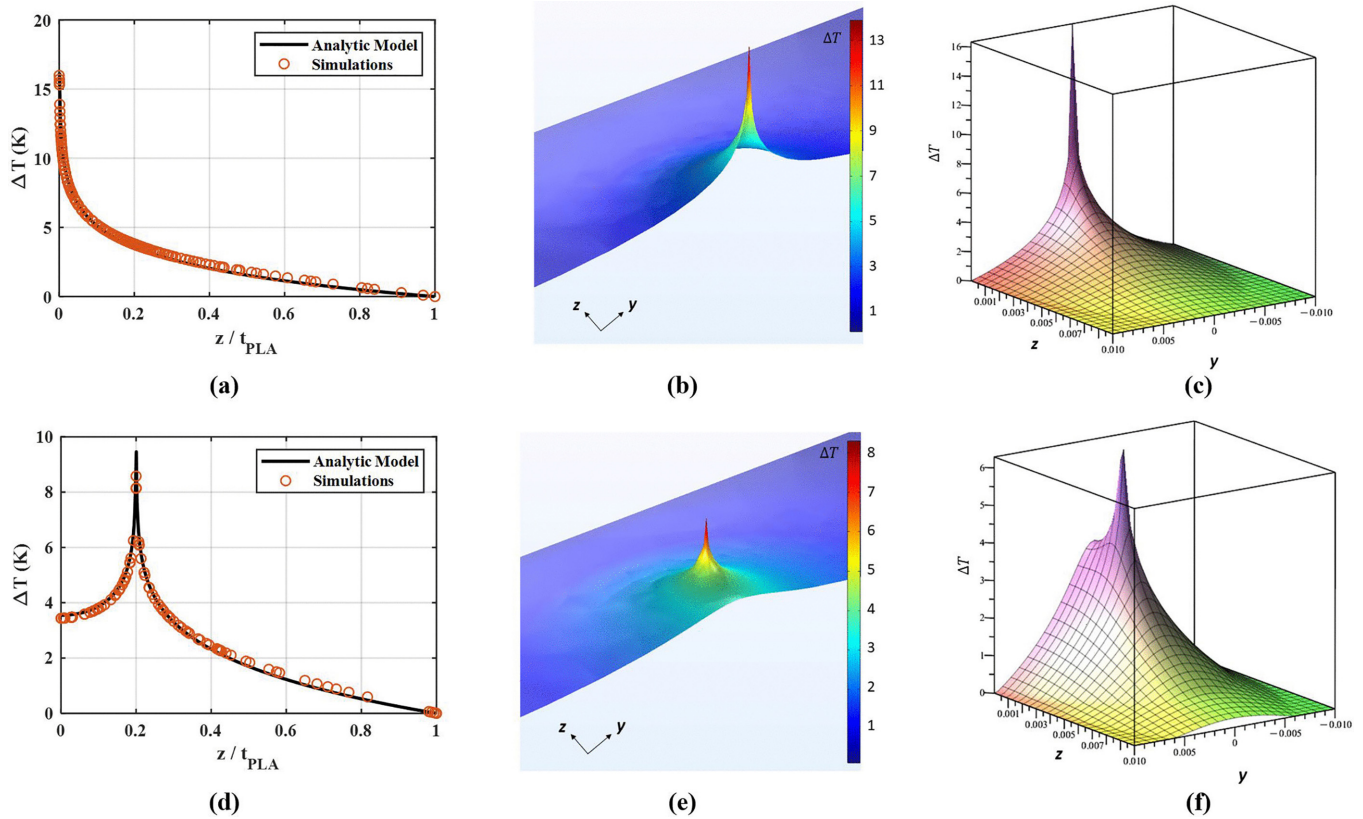


FIG. 6. Comparing analytic model and simulations for resistors 1 and 3. ΔT of (a) resistor 1 and (d) resistor 3 in z direction when $y = 0$. Simulated temperature distribution for resistor 1 (b) compared to its analytic model (c). Simulated temperature distribution for resistor 3 (e) compared to its analytical model (f).

phenomenon of self-heating in a resistor requires the integration of multiple physics modules including electric currents from the *AC/DC module*, heat transfer in solids from the *heat transfer module*, and a coupled physics module known as *electromagnetic heating*. Current densities (J_i) were applied through the resistors/neurons, with each having a value of $J_i = 0.127 \left(\frac{\mu A}{\mu m^2} \right)$ for $i = 1, 2, 3, 4$. Figure 6 compares the results of numerical simulations against the analytic model derived earlier, illustrating good agreement between the two models. To further validate the analytical model, the simulation results for the temperature distribution within the reservoir were also studied. This comparison revealed a consistent temperature distribution in both the y and z directions, further affirming our analytical model’s reliability.

Evaluation of the dynamic response. Reservoir dynamics when all resistors are self-heated is illustrated in Fig. 7. To study how systems react over time, we use MATLAB to process simulation data. This involves using a combination of two exponentials in our computations, which helps us automatically calculate thermal time constants. It is worth noting that resistors within the same layer tend to exhibit similar time constants, but there can be noticeable differences in dynamics between neurons located in distinct layers.

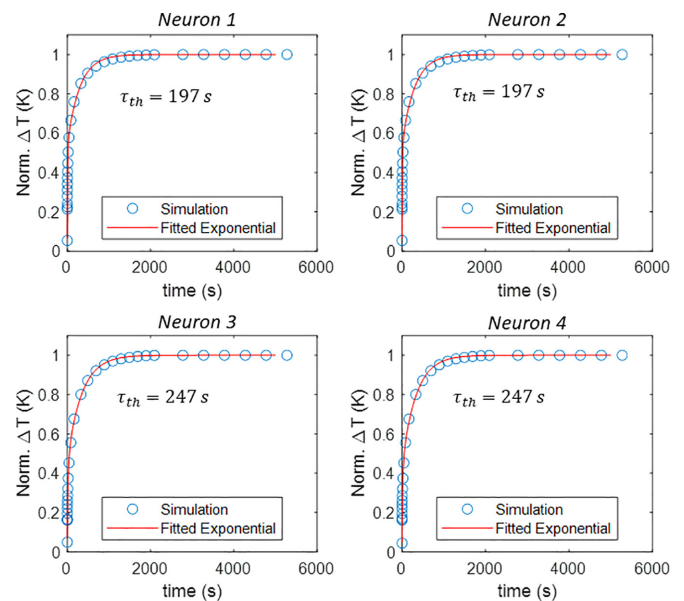


FIG. 7. Dynamics of the reservoir under different conditions when all resistors are self-heated together.

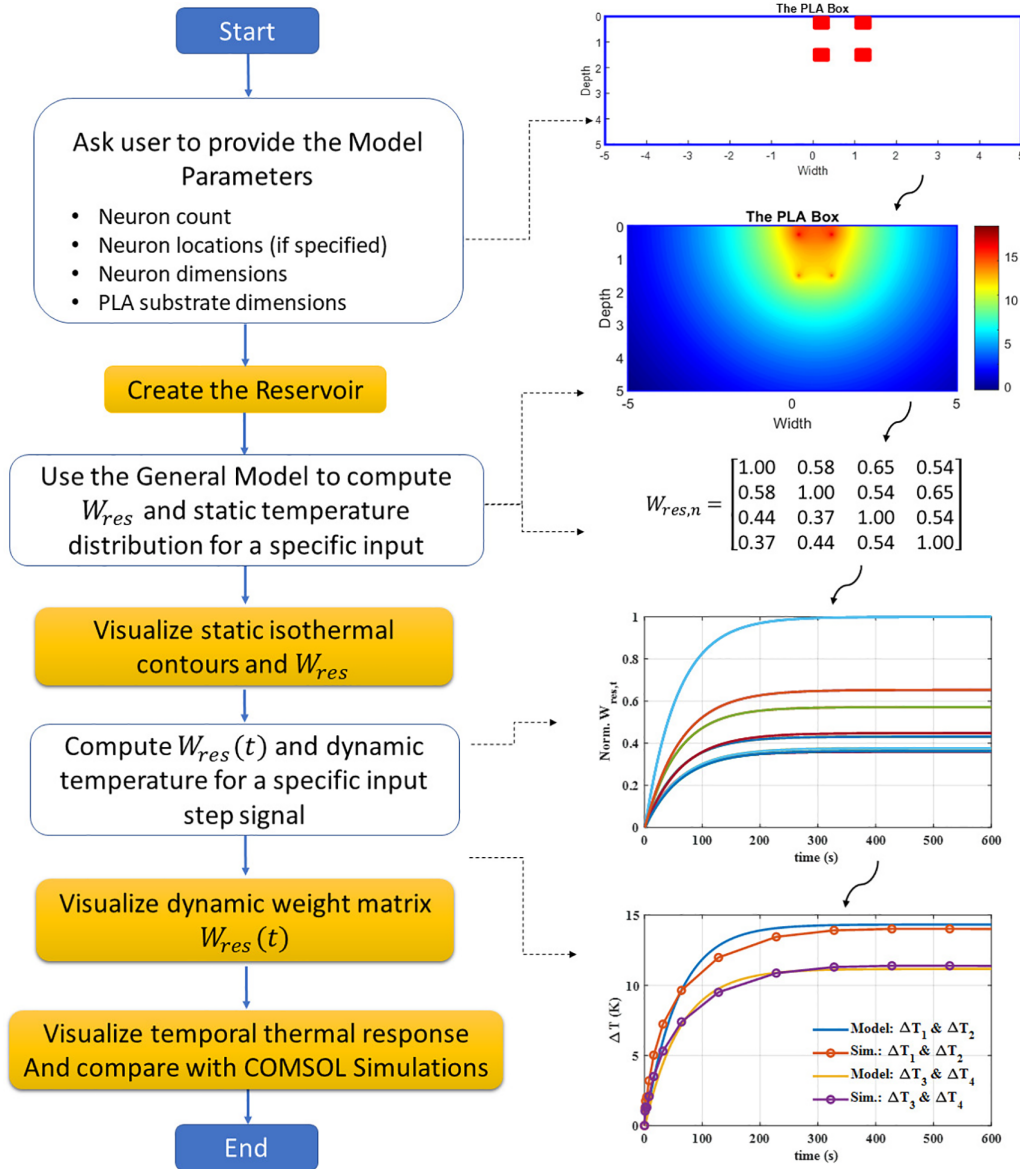


FIG. 8. Flowchart of the MATLAB script implementing a sample reservoir using the developed general model in the previous sections along with its step-by-step outcomes. The reservoir is configured first, isothermal contours are visualized, both static and dynamic weight matrices are derived based on a specified input, and a comparison between the temperature across each resistor/neuron within the physical reservoir to the corresponding temperatures obtained from COMSOL Multiphysics simulations are made.

Moreover, we expected a τ_{th} value of 225 s for a resistor on the top layer and 234 s for a resistor situated 2 mm below the top surface in a 10-mm-thick PLA. However, a deviation from the calculated values for the thermal time constants was observed. This deviation may be attributed to the fact that, although the dynamics of an individual neuron/resistor are defined by Eq. (13), the entire reservoir exhibits a dynamical behavior similar to predictions of Eq. (15).

IV. MODEL UTILITY IN DEVELOPING PHYSICAL RC

Based on the model developed in the above, a MATLAB script was developed to design and evaluate the performance of various 3D-printed RCs. The code asks the user for parameters necessary for constructing and developing 3D-printed physical reservoirs. These parameters include the number of neurons, neuron locations (can be specified or will be randomly assigned), neuron dimensions, and PLA reservoir dimensions. With these inputs, the script configures a

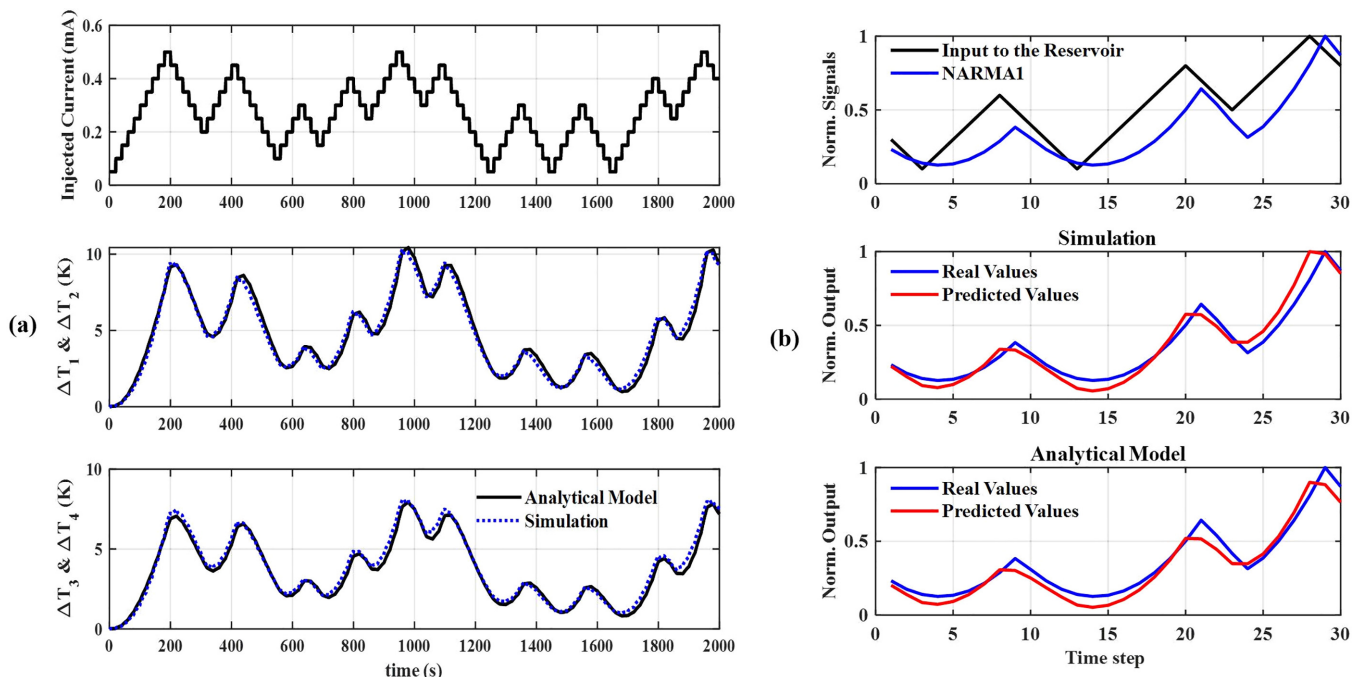


FIG. 9. (a) Comparison of the reservoir built in MATLAB environment using the developed general model and the simulated reservoir in COMSOL Multiphysics in response to a time series signal in the input. (b) Performance of the reservoir built in MATLAB environment using the developed analytical model with the reservoir simulated in COMSOL Multiphysics in solving a nonlinear problem.

reservoir. Drawing from insights derived from the general model, the script employs superposition to generate isothermal contours, which visually represent the temperature distribution within the reservoir. The script also generates both static and dynamic weight matrices based on input signals. This enables an evaluation of how the reservoir processes information over time, offering insights into its computational behavior. To validate our theoretical model and ensure its practical utility, the MATLAB script conducts a crucial verification step. It compares temperature data across each resistor/neuron within the developed physical reservoir using the described model above in MATLAB with corresponding data obtained from COMSOL Multiphysics simulations. This verification process strengthens the theoretical foundation of our model and validates its real-world applicability. The entire process is shown in Fig. 8.

We introduced a time series as an input to the model and simulations, allowing us to observe the dynamic response of the reservoir system in real time. Temperature distribution across the reservoir and individual neurons was monitored, providing a comprehensive view of the system's behavior over time. The results, as demonstrated in Fig. 9(a), demonstrated a high level of agreement between the simulation and theoretical model, affirming the model's ability to capture the reservoir computing system's dynamics in real-world scenarios.

Finally, we successfully addressed a nonlinear problem with memory constraints, known as nonlinear autoregressive moving average (NARMA), using the outlined reservoir. NARMA is a discrete-time temporal task with an n th-order time lag, given by [36]

$$y(t) = \alpha y(t-1) + \beta y(t-1) \sum_{i=1}^n y(t-i) + \gamma u(t-n) u(t-1) + \delta, \quad (22)$$

where $\alpha = 0.3$, $\beta = 0.05$, $\gamma = 1.5$, and $\delta = 0.1$ [36]. Performance assessments were conducted, showcasing the model's comparability with reservoirs constructed in both the MATLAB environment and simulations, as illustrated in Fig. 9(b). This alignment underscores the practical applicability and relevance of our theoretical model in real-world RC applications.

Importance of sampling frequency. This study delves into the utility of Eq. (16), which highlights the reliance of the physical reservoir's performance on the sampling frequency f_s . In Fig. 10(a), the developed model predicts the temperature across each resistor under varying f_s , while applying the same time-series input current to the reservoir. Noticeably, reducing f_s provides the resistors with more time to heat up, resulting in higher temperatures. This underscores the significance of sampling frequency in time-series data analysis. Fig. 10(b) presents the NARMA1 task under different f_s . A

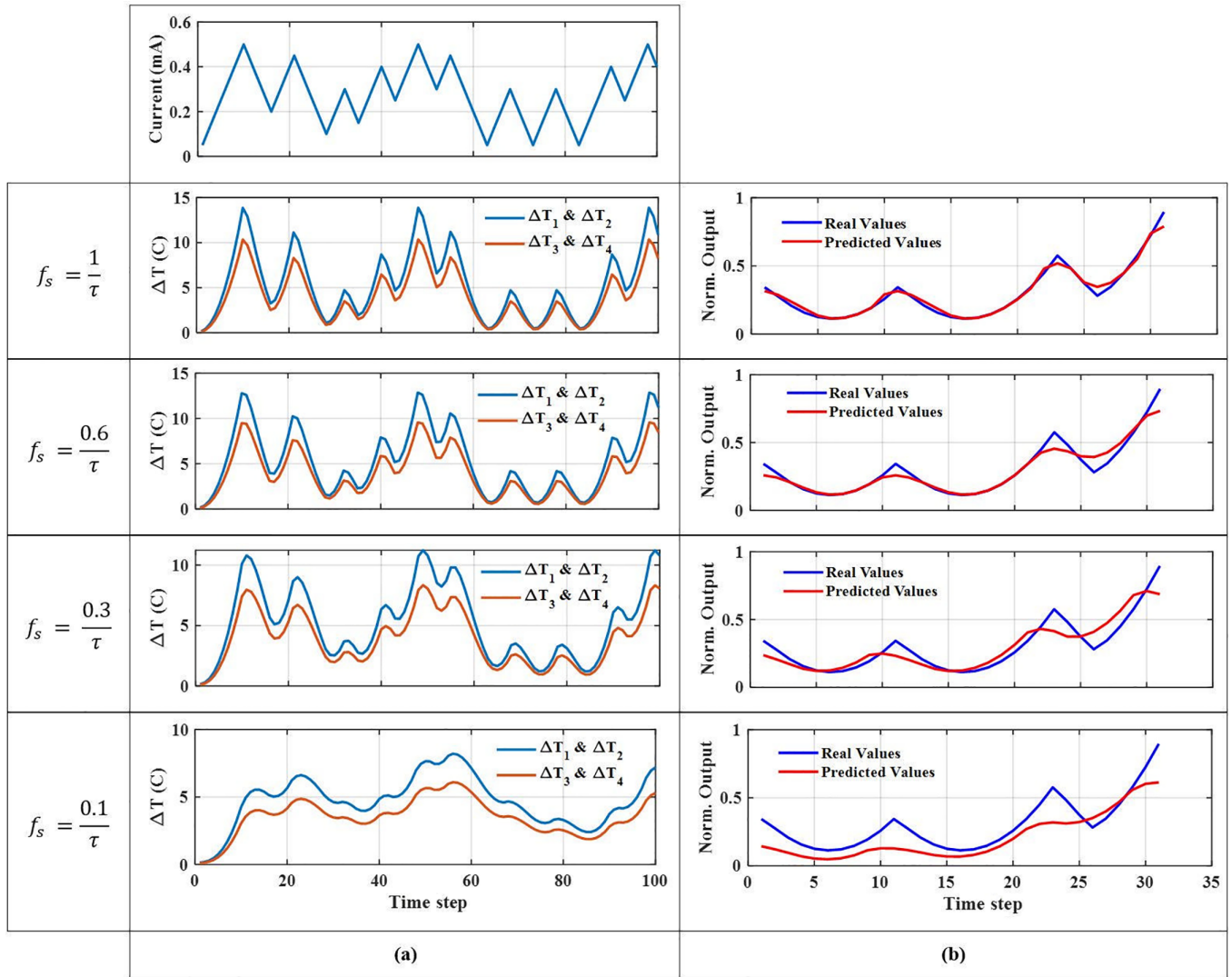


FIG. 10. Impact of the sampling frequency on (a) temperature across the neurons and (b) NARMA1 performance. This also underscores the model’s proficiency in capturing the time-dependent behavior of thermal reservoirs.

higher frequency captures less nonlinearity, leading to poorer performance in predicting target values. However, at lower frequencies, the resistors experience greater nonlinearity, excelling in tracking the target signal. In addition, a memory capacity task was conducted to evaluate the system’s ability to retain information over time. Short-term memory capacity in the RC systems is crucial for applications requiring recollection of past events. This is based on the system’s capability to recall past information using linear combinations of its internal states. Short-term memory capacity (MC) is evaluated by computing the t_d -delay memory capacity, $MC(t_d)$, as introduced and derived in [27]:

$$MC(t_d) = \frac{\text{cov}^2[\mathbf{u}(t - t_d), \mathbf{y}_{t_d}(t)]}{\sigma^2[\mathbf{u}(t)] \sigma^2[\mathbf{y}_{t_d}(t)]}, \quad (23)$$

where $\mathbf{u}(t - t_d)$ is a delayed input and $\mathbf{y}_{t_d} = \mathbf{u}(t - t_d)$ is its reconstruction at the output of the RC system. cov represents

the covariance of the two time series under study, and σ_2 is the variance of the time-series signals, either the input or the output of the linear readout layer. The overall short-term memory capacity is then approximated as

$$MC = \sum_{\tau=1}^{\infty} MC(t_d). \quad (24)$$

The results in Fig. 11 demonstrate a direct correlation between sampling frequency and memory capacity: increasing the sampling frequency enhances the system’s memory capacity, allowing it to retain more information from past observations. Conversely, decreasing the sampling frequency reduces the system’s memory capacity, limiting its ability to preserve historical data.

Power consumption and speed. During our experiments, we 3D printed a reservoir with 18 neurons (~324 weight values)

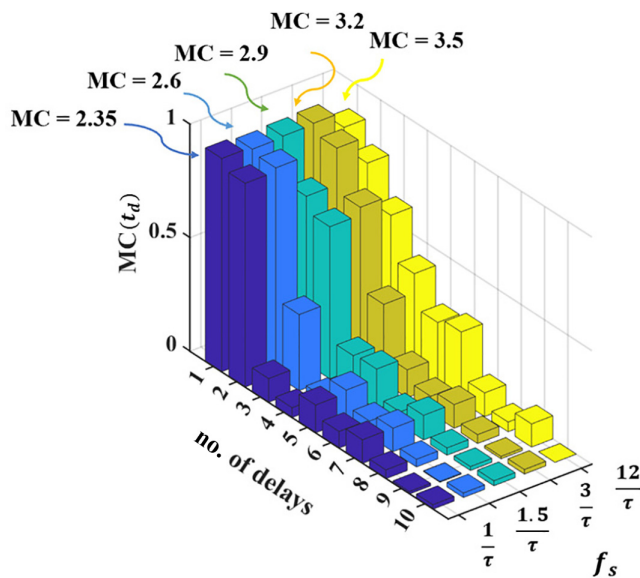


FIG. 11. Memory capacity of the physical reservoir under different sampling frequencies. Note that this MC is for four neurons solely. By increasing the number of recurrently connected neurons inside the reservoir, the MC increases.

leveraging a thermal time constant of ~ 60 s and average power consumption of 7 mW/neuron (i.e., 0.4 mW/weight) [28]. Since the reservoir is a passive network, it remains inactive until an input signal is present. The average power consumed by the network is measured by the amount of power drawn from the input sources. Subsequently, the power consumption per neuron is calculated by dividing the total

power by the number of neurons in the network. We employed a common 3D printer to fabricate a processor with promising computing capabilities processing a sample every 10 s. Exploring advanced 3D printing technologies like microstereolithography (micro-SLA), with minimum feature sizes in micrometers, holds promise. By reducing each dimension by a factor of 10 (from mm to sub-mm range), power consumption could decrease by 3 orders of magnitude to 7 μ W/neuron (i.e., 0.4 μ W/weight), and thermal time constant decreases by 2 orders of magnitude to ~ 600 ms, offering a processing speed of ~ 10 samples/s. Transitioning from a 3D-printed reservoir to silicon offers a remarkable leap, due to smaller feature sizes and better thermal properties reducing the thermal time constant by at least 5 orders of magnitude, resulting in a maximum timescale of ~ 0.6 ms and a minimum processing speed of 10 000 samples/s. An average power consumption of 7 nW/neuron (i.e., 0.4 nW/weight) is achievable by dimension reduction from a range of mm to tens of μ m using readily available microfabrication technologies, which can be translated to 0.4 pJ/weight for silicon reservoirs. This shift underscores the pivotal advantages of silicon reservoirs compared to a standard feedforward neural network running on the most efficient supercomputer on the Green500 list consuming 15 pJ/weight [37].

ACKNOWLEDGMENTS

Funding for this research was provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) through Grant No. RGPIN-2020-06348. The authors acknowledge the support from CMC Microsystems that provided access to software and measurement tools.

- [1] H. Jaeger, The ‘echo state’ approach to analysing and training recurrent neural networks, in *German National Research Center for Information Technology*, Technical Report GMD Report 148 (2001), pp. 1–47.
- [2] L. Melandri, Introduction to reservoir computing methods, Ph.D. thesis, Universita di Bologna, 2014.
- [3] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, Recent advances in physical reservoir computing: A review, *Neural Netw.* **115**, 100 (2019).
- [4] G. Van Der Sande, D. Brunner, and M. C. Soriano, Advances in photonic reservoir computing, *Nanophotonics* **6**, 561 (2017).
- [5] F. Duport, A. Smerieri, A. Akrou, M. Haelterman, and S. Massar, Fully analogue photonic reservoir computer, *Sci. Rep.* **6**, 22381 (2016).
- [6] B. Barazani, G. Dion, J. F. Morissette, L. Beaudoin, and J. Sylvestre, Microfabricated neuroaccelerometer: Integrating sensing and reservoir computing in MEMS, *J. Microelectromechanical Syst.* **29**, 338 (2020).
- [7] Q. Xia and J. J. Yang, Memristive crossbar arrays for brain-inspired computing, *Nat. Mater.* **18**, 309 (2019).
- [8] F. Caravelli and J. Carbajal, Memristors for the curious outsiders, *Technologies* **6**, 118 (2018).
- [9] H. C. Card, C. R. Schneider, and R. S. Schneider, Learning capacitive weights in analog CMOS neural networks, *J. VLSI Signal Process.* **8**, 209 (1994).
- [10] L. Appeltant, M. C. Soriano, G. V. Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, Information processing using a single dynamical node as complex system, *Nat. Commun.* **2**, 468 (2011).
- [11] S. Barbay, R. Kuszelewicz, and A. M. Yacomotti, Excitability in a semiconductor laser with saturable absorber, *Opt. Lett.* **36**, 4476 (2011).
- [12] G. Dion, S. Mejaouri, and J. Sylvestre, Reservoir computing with a single delay-coupled non-linear mechanical oscillator, *J. Appl. Phys.* **124**, 152132 (2018).
- [13] J. C. Coulombe, M. C. A. York, and J. Sylvestre, Computing with networks of nonlinear mechanical oscillators, *PLoS One* **12**, e0178663 (2017).
- [14] P. Bhowad and S. Li, Physical reservoir computing with origami and its application to robotic crawling, *Sci. Rep.* **11**, 13002 (2021).
- [15] K. Vandoorne, P. Mechet, T. V. Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, Experimental demonstration of reservoir computing on a silicon photonics chip, *Nat. Commun.* **5**, 3541 (2014).
- [16] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing, *Opt. Express* **20**, 3241 (2012).

- [17] M. N. Ashner, U. Paudel, M. Luengo-Kovac, J. Pilawa, T. J. Shaw, and G. C. Valley, Photonic reservoir computer with all-optical reservoir, in *Proceedings of the SPIE 11703, AI and Optical Data Sciences II, 117030L* (SPIE OPTO, Bellingham, Washington USA, 2021), pp. 1–12.
- [18] F. Zhou and Y. Chai, Near-sensor and in-sensor computing, *Nat. Electron.* **3**, 664 (2020).
- [19] M. Yan, C. Huang, P. Bienstman, P. Tino, W. Lin, and J. Sun, Emerging opportunities and challenges for the future of reservoir computing, *Nat. Commun.* **15**, 2056 (2024).
- [20] K. Nakajima, Physical reservoir computing—an introductory perspective, *Jpn. J. Appl. Phys.* **59**, 060501 (2020).
- [21] J. K. Han, S. Y. Yun, J. M. Yu, and Y. K. Choi, Leaky FinFET for reservoir computing with temporal signal processing, *ACS Appl. Mater. Interfaces* **15**, 26960 (2023).
- [22] D. A. Allwood *et al.*, A perspective on physical reservoir computing with nanomagnetic devices, *Appl. Phys. Lett.* **122**, 040501 (2023).
- [23] Q. Ren and G. Yao, An opportunistic packet forwarding for energy-harvesting wireless sensor networks with dynamic and heterogeneous duty cycle, *IEEE Access* **10**, 121109 (2022).
- [24] A. Jalalvand, K. Demuynck, W. De Neve, and J. P. Martens, On the application of reservoir computing networks for noisy image recognition, *Neurocomputing* **277**, 237 (2018).
- [25] N. Schaetti, M. Salomon, and R. Couturier, Echo state networks-based reservoir computing for MNIST handwritten digits recognition, in *International Conference on Computational Science and Engineering, Paris, France* (IEEE, New York City, U.S., 2016), pp. 484–491, HAL Id: hal-02131170.
- [26] M. Dale, S. Stepney, J. F. Miller, and M. Trefzer, Reservoir computing in materio: A computational framework for in materio computing, in *Proceedings of the International Joint Conference on Neural Networks* (IEEE, Anchorage, AK, USA, 2017), pp. 2178–2185.
- [27] H. Jaeger, Short term memory in echo state networks, GMD-Report, 152. GMD - German National Research Institute for Computer Science, 2002.
- [28] V. Shirmohammadli and B. Bahreyni, A 3D-printed computer, *Adv. Intell. Syst.* **5**, 2300015 (2023).
- [29] H. Jaeger, Echo state network, *Scholarpedia* **2**, 2330 (2007).
- [30] A. Rodan and P. Tiño, Minimum complexity echo state network, *IEEE Trans. Neural Networks* **22**, 131 (2011).
- [31] M. Lukoševičius, A practical guide to applying echo state networks, in *Neural Networks: Tricks of the Trade*, 2nd ed., edited by G. Montavon, G. B. Orr, and K.-R. Müller, Lecture Notes in Computer Science (Springer, Berlin, Heidelberg, 2012), Vol. 7700, pp. 659–686.
- [32] M. L. Alomar Barceló, Methodologies for hardware implementation of reservoir computing systems, Doctoral thesis, Universitat de les Illes Balears, Palma, Spain, 2017.
- [33] V. Shirmohammadli and B. Bahreyni, A neuromorphic electrothermal processor for near-sensor computing, *Adv. Mater. Technol.* **7**, 2200361 (2022).
- [34] See Supplemental Material at <http://link.aps.org/supplemental/10.1103/PhysRevResearch.6.033055> for detailed derivations of the heat transfer model for resistors/neurons, adaptations for practical realistic dimensions, and analysis.
- [35] Y. A. Cengel, *Heat Transfer: A Practical Approach*, 2nd ed. (McGraw-Hill, New York, 2004).
- [36] F. Stelzer, A. Röhm, K. Lüdge, and S. Yanchuk, Performance boost of time-delay reservoir computing by non-resonant clock cycle, *Neural Netw.* **124**, 158 (2020).
- [37] Green500, <https://www.top500.org/lists/green500/2023/06/>.