POLITECNICO DI TORINO
Repository ISTITUZIONALE

Leveraging phase model reduction and equilibrium propagation for neuromorphic design optimization of oscillatory neural networks

Original
Leveraging phase model reduction and equilibrium propagation for neuromorphic design optimization of oscillatory neural networks / Gemo, E., Bonnin, M., Corinto, F.. - In: NEUROMORPHIC COMPUTING AND ENGINEERING. - ISSN 2634-4386. - ELETTRONICO. - 6:(2026). [10.1088/2634-4386/ae66b3]

Availability:
This version is available at: 11583/3011450 since: 2026-05-27T08:22:22Z

Publisher:
IOP

Published
DOI:10.1088/2634-4386/ae66b3

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

20 November 2026

Neuromorphic
Computing and
Engineering

PURPOSE-LED
PUBLISHING™

PAPER • OPEN ACCESS

Leveraging phase model reduction and equilibrium propagation for neuromorphic design optimization of oscillatory neural networks

To cite this article: Emanuele Gemo *et al* 2026 *Neuromorph. Comput. Eng.* **6** 024016

View the article online for updates and enhancements.

You may also like

- Training and synchronizing oscillator networks with Equilibrium Propagation
Théophile Rageau and Julie Grollier
- Pattern recognition with simple oscillating circuits
R W Hölzel and K Krischer
- All-memristive neuromorphic computing with level-tuned neurons
Angeliki Pantazi, Stanisaw Woniak, Tomas Tuma et al.

This content was downloaded from IP address 109.54.178.228 on 27/05/2026 at 08:18

NEUROMORPHIC
Computing and Engineering

Check for updates

OPEN ACCESS

RECEIVED
6 November 2025

REVISED
17 April 2026

ACCEPTED FOR PUBLICATION
29 April 2026

PUBLISHED
22 May 2026

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

PAPER

Leveraging phase model reduction and equilibrium propagation for neuromorphic design optimization of oscillatory neural networks

Emanuele Gemo*, Michele Bonnin and Fernando Corinto

Department of Electronics and Telecommunications, Politecnico di Torino, Torino, Italy
* Author to whom any correspondence should be addressed.

E-mail: emanuele.gemo@polito.it

Keywords: oscillatory neural networks, amplitude-phase model reduction, phase-deviation model, energy-based learning, equilibrium propagation, associative memory, neuromorphic computing

Abstract

Although neuromorphic computing promises energy-efficient and biologically inspired machine learning architectures, the implementation of neural dynamics into practical hardware design remains a major challenge. This work presents a unified analytical framework that integrates equilibrium propagation (EP) with oscillatory neural networks (ONNs) through a phase-deviation formulation. By exploiting the synchronization properties of oscillatory systems, we establish a mathematical link between circuit dynamics, learning behavior, and the underlying energy landscape derived from the phase-deviation equations. This formulation reveals how ONNs naturally operate as an analog associative memory, where equilibrium points correspond to minima of the energy function governing collective phase dynamics. We derive the full phase-deviation formalism for weakly coupled oscillatory networks and adapt EP to their dynamics. Numerical simulations on pattern-recognition tasks are provided as a supporting example, illustrating how the proposed analytical framework can reproduce learning and retrieval behavior in a phase-based ONN setting. These results support the use of the proposed formalism as a principled framework for analyzing equilibrium structure, learning dynamics, and possible failure modes in oscillatory neuromorphic systems.

1. Introduction

Artificial intelligence (AI) has become a central topic in both academic research and industrial applications, with profound economic and social implications. AI-driven technologies are already widely accessible to the public and continue to attract growing interest across sectors. However, this rapid expansion presents a critical challenge: the substantial increase in energy consumption associated with AI systems. These systems rely on computationally intensive algorithms and architectures constrained by the von Neumann paradigm, both of which demand considerable resources. Data centers—the primary infrastructure supporting such high-performance workloads—already account for a significant share of global energy usage [1]. This reality underscores the urgent need for research into more energy-efficient algorithms and hardware architectures capable of offloading, accelerating, or otherwise optimizing core AI computations, or providing principled design guidance for such systems.

To this end, oscillatory neural networks (ONNs) have recently attracted renewed attention [2–4]. The defining feature of ONNs is the encoding of information in the phase relationships among interconnected oscillators, with computation emerging from their spontaneous or guided dynamics. The study of oscillatory systems builds upon a rich plethora of research dating back to the seminal works of Lorenz and Ueda in the 1960s, which revealed a wide range of nonlinear dynamical behaviors. Because physical oscillatory circuit elements are relatively simple to implement, theoretical developments have been readily transferred to hardware platforms, keeping this research area highly active to the present day [5].

© 2026 The Author(s). Published by IOP Publishing Ltd

Combined with the intrinsic energy efficiency of such oscillator-based circuits [6, 7], ONNs-based computational paradigms have emerged as particularly promising. Moreover, ONNs are particularly relevant in cognitive science [8–10] and neuromorphic engineering [11–15], due to the similarities between oscillator synchronization mechanisms and those observed in biological neural systems. Depending on the physical implementation and modeling assumptions, ONNs have been explored using discrete- or continuous-phase representations, as well as phase–amplitude formulations. Consequently, ONNs demonstrators have been proposed in many frameworks and for a variety of applications, including image processing [7, 16], vehicle routing [17], combinatorial optimization [18–21], pattern recognition [15, 22], and others.

Recent studies have further advanced the physical realizability and computational robustness of ONNs hardware implementations. Among these, Núñez *et al* [19] demonstrated that VO_2-based oscillators exploiting subharmonic injection locking can reliably discretize oscillator phases into two stable states, thereby enhancing resilience to variability and enabling associative-memory operations. Such architectures highlight the feasibility of nanoscale ONNs designs that encode binary phase information, paving the way toward scalable and energy-efficient neuromorphic hardware. However, discrete-phase encoding approaches inherently limit the richness of the accessible phase space. At the same time, despite the promising efficiency of ONN-based computing paradigms, a well-defined framework that unifies hardware considerations, processing constraints, and analytical modeling, is lacking. Delacour *et al* [21] have emphasized this gap, identifying analytical modeling as a needed link between neuromorphic theory and practical hardware realization. These considerations motivate further investigation into continuous-phase and analytically tractable ONNs models.

Another promising approach to minimizing both spatial and power demands in AI-oriented hardware platforms lies in the use of memristors. These elements can be reconfigured in a non-volatile manner, offering exceptionally high memory density and low write/erase energy requirements [23]. Moreover, memristors can be operated with minimal supporting circuitry, functioning as passive components during read operations. Consequently, they represent a compelling option for representing network parameters, such as synaptic weights, in emerging machine-learning and neuromorphic architectures. Importantly, the volatility of memristive elements can be engineered [24], providing designers with additional flexibility to implement biologically inspired synaptic dynamics. Furthermore, actively controlled memristors have been shown to replace the nonlinear diode in Chua's circuit [25], which evidence their suitability as compact nonlinear elements in oscillator-based dynamical models. In this context, memristive devices are naturally suited to serve as physically motivated models of adaptive coupling and nonlinearity within oscillator-based networks, while also offering a path toward compact implementations. In the present work, they are therefore considered primarily as analytically convenient and physically plausible representations of synaptic parameters, rather than as fully validated circuit components, allowing us to focus on the dynamical and learning principles that govern ONNs.

A growing body of computational and experimental studies explores the integration of memristive functionality within ONNs architectures [9, 14, 26, 27]. However, simulating such systems at the circuit level often necessitates the use of compact or approximate models, due to the polynomial growth in complexity arising from (i) the large number of circuit blocks, (ii) the interconnecting circuitry, and (iii) the extended simulated time required to reach ONN equilibrium states. A practical strategy to mitigate this challenge involves tracking the phase evolution of each oscillator circuit block through a reduced system of equations. Recently, several works [28–30] have demonstrated that phase-only oscillator models can be directly trained using local learning rules such as equilibrium propagation (EP), achieving competitive performance on pattern recognition tasks and standard benchmarks. These studies highlight the practical promise of phase-based learning in oscillatory networks, while typically assuming the validity of the reduced phase description throughout training.

In parallel, Rudner, Porod, and Csaba [31] recently demonstrated that modern machine learning techniques, such as backpropagation through time, can be applied directly to oscillator-based circuit models to train ONNs for classification and memory tasks. While this approach significantly extends the computational capabilities of ONNs, it relies on global gradient propagation and numerical differentiation across time, which pose substantial challenges for physical implementation in hardware. These limitations have motivated increasing interest in alternative learning paradigms that preserve local dynamics and physical interpretability. In particular, EP has emerged as a theoretically mature framework for training energy-based and dynamical systems using local updates, with well-established connections to recurrent backpropagation [32, 33].

In this context, reducing ONN dynamics to phase dynamics represents a natural and theoretically well-founded advancement. Phase reduction decomposes nonlinear oscillator behavior into coupled phase–amplitude equations that preserve essential synchronization dynamics while remaining amenable

to analytical treatment [34–37]. While this formulation offers stability and interpretability benefits and is widely used in control-theoretic and neuromorphic contexts, its validity relies on asymptotic assumptions such as weak coupling and stable limit cycles. Recent work on higher-order and non-asymptotic phase reduction shows that learning-induced changes in coupling or oscillator heterogeneity can introduce systematic corrections beyond first-order phase models [38–42]. Therefore, combining phase-reduced dynamics with local learning mechanisms (such as EP) yields a compact and analytically grounded framework whose applicability can be explicitly assessed.

The main goal of this work is to consolidate the mathematical foundation for using phase models in the design of ONNs circuitry through both analytical arguments and numerical results. We show that phase-reduced models can capture and predict the equilibrium structure of an ONN under the assumptions discussed in the paper. Building on this, we show how the phase-state function can be used to model an oscillatory circuit block, incorporating memristive elements as synaptic parameters, and to construct a small-scale fully connected numerical example. We further illustrate, through a pattern-recognition example, how this model can be coupled with the EP local learning rule. Finally, we show that this relatively simple framework provides an interpretable and analytically grounded platform for studying learning dynamics, equilibrium states, and failure modes in phase-based ONNs.

2. Materials and methods

2.1. Neuromorphic neural networks

Neuromorphic computing systems can generally be modeled as interconnected dynamical units, whose evolution depends on their internal state, on external inputs, and on adaptive parameters defining their coupling. Expanding on the formalism proposed in [43], the vectorial state-space representation of a neuromorphic system can be expressed, in its broadest form, as

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{1}$$

where $\boldsymbol{x}$ denotes the system's internal dynamics, $\boldsymbol{u}$ the input stimulus, and $\boldsymbol{w}$ the adaptive parameters (e.g, synaptic weights or coupling coefficients).

Within this framework, our study focuses on ONNs, in which each node behaves as a nonlinear oscillator capable of phase synchronization. The model consists of N non–identical, coupled dynamical systems, described by:

$$\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} = \boldsymbol{f}_i(\boldsymbol{x}_i) + \varepsilon \sum_{j \in N_i} \boldsymbol{c}_{ij}\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) \qquad i = 1, \ldots, N \tag{2}$$

where $\boldsymbol{x}_i : \mathbb{R}^+ \mapsto \mathbb{R}^n$, $i = 1, \ldots, N$ describes the state of the ith node, $\boldsymbol{f}_i : \mathbb{R}^n \mapsto \mathbb{R}^n$ describes its intrinsic dynamics, and $\boldsymbol{f}_i \in \mathcal{C}^1(\Omega \subseteq \mathbb{R}^n)$. The parameter $\varepsilon \geqslant 0$ measures the coupling strength, while $\boldsymbol{c}_{ij} : \mathbb{R}^n \times \mathbb{R}^n \mapsto \mathbb{R}^n$ is a Lipschitz-continuous function that describes the interaction between nodes i and j, running on all nodes N_i that are connected to the node i.

In practical implementations, the nodes composing the network are homogeneous, and node-to-node differences mainly stem from parameter variations. Thus we shall assume

$$\boldsymbol{f}_i(\boldsymbol{x}_i) = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\mu}_i) \tag{3}$$

where $\boldsymbol{\mu}_i$, $i = 1, \ldots, N$ are vectors of parameters. For large numbers of oscillators, it is reasonable to assume that the parameters are normally distributed around a mean value $\boldsymbol{\mu}_0$. If the variance is small enough, we can write $\boldsymbol{\mu}_i = \boldsymbol{\mu}_0 + \varepsilon\delta\boldsymbol{\mu}_i$, and expanding equation (3) in Taylor series we obtain

$$\boldsymbol{f}_i(\boldsymbol{x}_i) = \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\mu}_0) + \varepsilon \frac{\partial \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\mu}_0)}{\partial \boldsymbol{\mu}_i} \delta\boldsymbol{\mu}_i + \mathcal{O}\left(\varepsilon^2\right) \tag{4}$$

where $\partial \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\mu}_0)/\partial \boldsymbol{\mu}_i$ is the jacobian matrix of partial derivatives. For simplicity of notation we shall write

$$\boldsymbol{f}_i(\boldsymbol{x}_i) \cong \boldsymbol{f}(\boldsymbol{x}_i) + \varepsilon\widetilde{\boldsymbol{f}}_i(\boldsymbol{x}_i) \tag{5}$$

where the term $\varepsilon\widetilde{\boldsymbol{f}}_i(\boldsymbol{x}_i) = \varepsilon\partial \boldsymbol{f}(\boldsymbol{x}_i, \boldsymbol{\mu}_0)/\partial \boldsymbol{\mu}_i\, \delta\boldsymbol{\mu}_i$ describe the deviation of the ith oscillator from an ideal one.

As a special case, we consider a structured network composed of $N = 2M$ oscillatory units, arranged in pairs of interacting subsystems, governed by

$$\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_i) + \varepsilon \left(\widetilde{\boldsymbol{f}}_i(\boldsymbol{x}_i) + \boldsymbol{d}_i(\boldsymbol{x}_i, \boldsymbol{x}_{i+M}) + \sum_{j \in \widetilde{N}_i} \widetilde{\boldsymbol{c}}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) \right) \tag{6}$$

$$\frac{\mathrm{d}\boldsymbol{x}_{i+M}}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_{i+M}) \tag{7}$$

for $i = 1, \ldots, M$, and with $\widetilde{N}_i \subseteq \{1, \ldots, M\}$. Without too much loss of generality, in equations (6) and (7) we have assumed that the oscillators $i + M$ are ideal. Equations (6) and (7) describe a pool of M driven oscillators coupled both to neighboring units (via $\widetilde{\boldsymbol{c}}_{ij}$) and to the corresponding driving oscillators (via $\boldsymbol{d}_i$). The driving subsystems $\boldsymbol{x}_{i+M}$ act as weakly coupled internal references, serving a function analogous to equilibrium or 'nudged' states in local learning paradigms (as a modeling construct). Although the driving nodes follow identical dynamics, their states may differ due to different initial conditions, i.e. $\boldsymbol{x}_i(t) \neq \boldsymbol{x}_{i+M}(t)$.

This configuration unifies the general neuromorphic model of [43] with oscillatory-based formulations commonly used in hardware-motivated ONN computing models. It provides a tractable setting in which synchronization, energy minimization, and parameter adaptation can be jointly analyzed. In particular, the coupling structure $\boldsymbol{c}_{ij}$ can encode tunable interactions equivalent to synaptic weights. As shown by Rudner *et al* [31], oscillator networks can be optimized via circuit-level learning rules. The present work aims to extend the modeling approach by embedding local equilibrium-based adaptation directly into the dynamical model. This interpretation naturally links the oscillatory network model to local learning mechanisms such as EP [29, 30], while remaining consistent with standard phase-reduction approaches (e.g. Hongu and Iba [39]).

Hence, the present framework can be regarded as a specific instantiation of the general neuromorphic dynamical model, tailored to oscillatory computation. It establishes the mathematical basis for merging synchronization theory, memristive circuit models, and equilibrium-based learning within a unified dynamical formulation. The next section elaborates on this connection by deriving the reduced phase model that governs synchronization and adaptation in such networks.

2.2. Phase model reduction and the phase deviation equation

This section outlines a systematic procedure for deriving a reduced-order, simplified model that captures the essential network dynamics, making it suitable for applications in synchronization and neural network analysis.

Classical phase-reduction approaches based on Malkin's theorem [35–37], apply only to resonant oscillators, i.e. oscillators whose frequency ratios are rational numbers. More generally, phase-reduction techniques inspired by Kuramoto require the definition of an appropriate phase function (and possibly an amplitude function), which can be obtained analytically only for simple systems such as the Stuart–Landau oscillator [38–42, 44]. For more complex oscillators, these quantities must typically be computed numerically through phase response curves, which becomes increasingly demanding for high-dimensional systems. Local approximations of the phase gradient based on Floquet vectors provide only a local description and are not easily extended to problems requiring higher order approximations, such as stochastic oscillators, where higher-order derivatives arise from Itô's formula [34, 45]. By contrast, the systematic amplitude–phase decomposition introduced here is exact, can be applied to networks with arbitrary coupling strengths, and can be extended to noisy oscillators. A weak-coupling assumption is required only for the subsequent phase-reduced model, which naturally leads to a Kuramoto-type description.

Assuming that each node in the network behaves as an oscillator, the original system of equations can be reduced to a phase equation model–specifically, a set of ordinary differential equations (ODEs) governing the oscillator phases, analogous to the well-known Kuramoto model [44].

The phase model offers two main advantages. First, it significantly reduces the system's dimensionality, requiring only one state variable and one corresponding equation per node. Second, it greatly simplifies the study of synchronization manifolds: synchronous oscillations in the original system (2) correspond to equilibrium points in the phase deviation model. Likewise, the stability analysis becomes more tractable, as it no longer requires computing Floquet multipliers; instead, it suffices to evaluate the eigenvalues of the Jacobian matrix.

Assume that each unperturbed node ($\varepsilon = 0$) admits an isolated T-periodic solution $\boldsymbol{x}_s(t)$

$$\frac{\mathrm{d}\boldsymbol{x}_s(t)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}_s(t)) \tag{8}$$

$$\boldsymbol{x}_s(t) = \boldsymbol{x}_s(t+T), \tag{9}$$

representing a limit cycle in the oscillator's state space. We define the unit vector tangent to the limit cycle

$$\boldsymbol{u}_1(t) = \frac{\boldsymbol{f}(\boldsymbol{x}_s(t))}{||\boldsymbol{f}(\boldsymbol{x}_s(t))||}, \tag{10}$$

where $||\cdot||$ denotes the standard L_2 norm. Together with $\boldsymbol{u}_1(t)$, we consider other $n-1$ linearly independent vectors $\boldsymbol{u}_2(t), \ldots, \boldsymbol{u}_n(t)$, such that, for all t:

$$\mathrm{span}\{\boldsymbol{u}_1(t), \boldsymbol{u}_2(t), \ldots, \boldsymbol{u}_n(t)\} = \mathbb{R}^n.$$

Given the matrix $\boldsymbol{U}(t) \in \mathbb{R}^{n,n}$, with $\boldsymbol{U}(t) = [\boldsymbol{u}_1(t), \ldots, \boldsymbol{u}_n(t)]$, let

$$\boldsymbol{V}(t) = \boldsymbol{U}^{-1}(t) = \begin{bmatrix} \boldsymbol{v}_1^T(t) \\ \vdots \\ \boldsymbol{v}_n^T(t) \end{bmatrix}. \tag{11}$$

Thus

$$\mathrm{span}\{\boldsymbol{v}_1(t), \ldots, \boldsymbol{v}_n(t)\} = \mathbb{R}^n,$$

also holds for all t, together with a bi-orthogonality condition

$$\boldsymbol{v}_i^T(t)\,\boldsymbol{u}_j(t) = \boldsymbol{u}_i^T(t)\,\boldsymbol{v}_j(t) = \delta_{ij}, \tag{12}$$

where δ_{ij} is the Kronecker delta. Finally, we introduce the matrices $\boldsymbol{Y}(t) = [\boldsymbol{u}_2(t), \ldots, \boldsymbol{u}_n(t)]$, and $\boldsymbol{Z}(t) = [\boldsymbol{v}_2(t), \ldots, \boldsymbol{v}_n(t)]$.

Because the vectors $\{\boldsymbol{u}_1(t), \boldsymbol{u}_2(t), \ldots, \boldsymbol{u}_n(t)\}$ and $\{\boldsymbol{v}_1(t), \boldsymbol{v}_2(t), \ldots, \boldsymbol{v}_n(t)\}$ are basis for $\mathbb{R}^n$, they can be used to decompose any vector. In particular, the coupling function can be decomposed into two contributions. The first contribution is projected along the direction spanned by $\boldsymbol{u}_1(t)$. At any time instant, this component is tangent to the limit cycle and, for small enough values of ε, it induces a shift along the limit cycle, without affecting the amplitude. The shift is measured by a function $\theta_i(t)$, that is interpreted as a phase, mapping the limit cycle onto the interval $[0, 2\pi]$. The second component is projected into the linear space spanned by the vectors $\{\boldsymbol{v}_2(t), \ldots, \boldsymbol{v}_n(t)\}$. This component is transversal to the limit cycle. Therefore, it modifies the amplitude of the limit cycle, inducing an amplitude deviation measured by the function $\boldsymbol{R}_i(t)$, but it leaves the phase unaffected as a consequence of the bi-orthogonality condition.

Consequently, the state of each node in the ONN can be described as a shifted version of the unperturbed limit cycle $\boldsymbol{x}_s(\theta_i(t))$, plus a deviation from the limit cycle $\boldsymbol{R}_i(t)$, measured along the direction spanned by the vectors $\{\boldsymbol{v}_2(\theta_i(t)), \ldots, \boldsymbol{v}_n(\theta_i(t))\}$, evaluated at the 'shifted' time θ_i. The first step to obtain the phase reduced model of the ONN (2), is the derivation of a ODEs system for the phase and the amplitude deviation functions.

Theorem 1 (phase-amplitude deviation equations). *Consider system (2), where each node admits a T-periodic limit cycle $\boldsymbol{x}_s(t)$ for $\varepsilon = 0$, together with the sets $\{\boldsymbol{u}_1(t), \ldots, \boldsymbol{u}_n(t)\}$, $\{\boldsymbol{v}_1(t), \ldots, \boldsymbol{v}_n(t)\}$ and the matrices $\boldsymbol{Y}(t)$, and $\boldsymbol{Z}(t)$. Consider the coordinate transformation*

$$\boldsymbol{x}_i(t) = \boldsymbol{h}(\theta_i(t), \boldsymbol{R}_i(t)) = \boldsymbol{x}_s(\theta_i(t)) + \boldsymbol{Y}(\theta_i(t))\,\boldsymbol{R}_i(t), \tag{13}$$

where $\theta_i(t)$ and $\boldsymbol{R}_i(t)$ are the phase and amplitude deviation, respectively. Then, in a neighborhood of the limit cycle $\boldsymbol{x}_s(t)$, the phase $\theta_i(t)$ and the amplitude deviation $\boldsymbol{R}_i(t)$ satisfy:

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}t} = 1 + a_\theta(\theta_i, \boldsymbol{R}_i) + \varepsilon \left(\delta a_\theta(\theta_i, \boldsymbol{R}_i) + \sum_{j \in N_i} C_{ij}(\theta_i, \boldsymbol{R}_i, \theta_j, \boldsymbol{R}_j) \right) \tag{14}$$

$$\frac{\mathrm{d}\boldsymbol{R}_i}{\mathrm{d}t} = \boldsymbol{L}(\theta_i)\,\boldsymbol{R}_i + \boldsymbol{a}_{\boldsymbol{R}}(\theta_i,\boldsymbol{R}_i) + \varepsilon\left(\delta\boldsymbol{a}_{\boldsymbol{R}}(\theta_i,\boldsymbol{R}_i) + \sum_{j\in N_i}\boldsymbol{D}_{ij}\left(\theta_i,\boldsymbol{R}_i,\theta_j,\boldsymbol{R}_j\right)\right), \tag{15}$$

where (explicit dependence on t in θ_i and $\boldsymbol{R}_i$ is omitted for simplicity):

$$K(\theta_i,\boldsymbol{R}_i) = \boldsymbol{v}_1^T(\theta_i)\left(\boldsymbol{f}(\boldsymbol{x}_s(\theta_i)) + \frac{\partial\boldsymbol{Y}(\theta_i)}{\partial\theta_i}\boldsymbol{R}_i\right), \tag{16a}$$

$$a_\theta(\theta_i,\boldsymbol{R}_i) = K^{-1}(\theta_i,\boldsymbol{R}_i)\,\boldsymbol{v}_1^T(\theta_i)\left(\boldsymbol{f}(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i) - \boldsymbol{f}(\boldsymbol{x}_s(\theta_i)) - \frac{\partial\boldsymbol{Y}(\theta_i)}{\partial\theta_i}\boldsymbol{R}_i\right), \tag{16b}$$

$$\delta a_\theta(\theta_i,\boldsymbol{R}_i) = K^{-1}(\theta_i,\boldsymbol{R}_i)\,\boldsymbol{v}_1^T(\theta_i)\,\widetilde{\boldsymbol{f}}_i(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i), \tag{16c}$$

$$C_{ij}\left(\theta_i,\boldsymbol{R}_i,\theta_j,\boldsymbol{R}_j\right) = K^{-1}(\theta_i,\boldsymbol{R}_i)\,\boldsymbol{v}_1^T(\theta_i)\,\boldsymbol{c}_{ij}\left(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i,\boldsymbol{x}_s\left(\theta_j\right)+\boldsymbol{Y}\left(\theta_j\right)\boldsymbol{R}_j\right), \tag{16d}$$

$$\boldsymbol{L}(\theta_i) = -\boldsymbol{Z}^T(\theta_i)\,\frac{\partial\boldsymbol{Y}(\theta_i)}{\partial\theta_i}, \tag{16e}$$

$$\boldsymbol{a}_{\boldsymbol{R}}(\theta_i,\boldsymbol{R}_i) = -\boldsymbol{Z}^T(\theta_i)\left(\frac{\partial\boldsymbol{Y}(\theta_i)}{\partial\theta_i}\boldsymbol{R}_i\,a_\theta(\theta_i,\boldsymbol{R}_i) - \boldsymbol{f}(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i)\right), \tag{16f}$$

$$\delta\boldsymbol{a}_{\boldsymbol{R}}(\theta_i,\boldsymbol{R}_i) = -\boldsymbol{Z}^T(\theta_i)\left(\frac{\partial\boldsymbol{Y}(\theta_i)}{\partial\theta_i}\boldsymbol{R}_i\,a_\theta(\theta_i,\boldsymbol{R}_i) - \widetilde{\boldsymbol{f}}_i(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i)\right) \tag{16g}$$

$$\begin{aligned}\boldsymbol{D}_{ij}\left(\theta_i,\boldsymbol{R}_i,\theta_j,\boldsymbol{R}_j\right) = &-\boldsymbol{Z}^T(\theta_i)\,\frac{\partial\boldsymbol{Y}(\theta_i)}{\partial\theta_i}\boldsymbol{R}_i\,C_{ij}\left(\theta_i,\boldsymbol{R}_i,\theta_j,\boldsymbol{R}_j\right)\\ &+\boldsymbol{Z}^T(\theta_i)\,\boldsymbol{c}_{ij}\left(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i,\boldsymbol{x}_s\left(\theta_j\right)+\boldsymbol{Y}\left(\theta_j\right)\boldsymbol{R}_j\right).\end{aligned} \tag{16h}$$

Proof. First, we show that the transformation (13) is invertible in a neighborhood of the limit cycle. The Jacobian matrix of the coordinate transformation is

$$D\boldsymbol{h}(\theta_i,\boldsymbol{R}_i) = \begin{bmatrix}\dfrac{\partial\boldsymbol{h}}{\partial\theta_i} & \dfrac{\partial\boldsymbol{h}}{\partial\boldsymbol{R}_i}\end{bmatrix} = \begin{bmatrix}\dfrac{\partial\boldsymbol{x}_s(\theta_i)}{\partial\theta_i} + \dfrac{\partial\boldsymbol{Y}(\theta_i)}{\partial\theta_i}\boldsymbol{R}_i & \boldsymbol{Y}(\theta_i)\end{bmatrix}. \tag{17}$$

On the limit cycle $\boldsymbol{R}_i = 0$ and then

$$D\boldsymbol{h}(\theta_i,\boldsymbol{R}_i)\big|_{\boldsymbol{R}_i=0} = \begin{bmatrix}\dfrac{\partial\boldsymbol{x}_s(\theta_i)}{\partial\theta_i} & \boldsymbol{Y}(\theta_i)\end{bmatrix} = [\,|\boldsymbol{f}(\boldsymbol{x}_s(\theta_i))\,|\,\boldsymbol{u}_1(\theta_i),\boldsymbol{u}_2(\theta_i),\ldots,\boldsymbol{u}_n(\theta_i)\,]. \tag{18}$$

Because $\{\boldsymbol{u}_1(t),\ldots,\boldsymbol{u}_n(t)\}$ are linearly independent, it follows that the Jacobian is regular. Then, by the inverse function theorem, there exists a neighborhood of $\boldsymbol{R}_i = 0$ where $\boldsymbol{h}$ is invertible. Moreover, if $\boldsymbol{h}$ is of class $\mathcal{C}^k$, then its inverse is also of class $\mathcal{C}^k$.

For deriving the governing equations for the phases and the amplitudes, taking the derivative of (13) we have:

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{x}_i}{\mathrm{d}t} &= \frac{\partial\boldsymbol{h}(\theta_i,\boldsymbol{R}_i)}{\partial\theta_i}\frac{\mathrm{d}\theta_i}{\mathrm{d}t} + \frac{\partial\boldsymbol{h}(\theta_i,\boldsymbol{R}_i)}{\partial\boldsymbol{R}_i}\frac{\mathrm{d}\boldsymbol{R}_i}{\mathrm{d}t} = \left(\frac{\partial\boldsymbol{x}_s(\theta_i)}{\partial\theta_i} + \frac{\partial\boldsymbol{Y}(\theta_i)}{\partial\theta_i}\boldsymbol{R}_i\right)\frac{\mathrm{d}\theta_i}{\mathrm{d}t} + \boldsymbol{Y}(\theta_i)\frac{\mathrm{d}\boldsymbol{R}_i}{\mathrm{d}t}\\ &= \boldsymbol{f}\Big(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i\Big) + \varepsilon\Big(\widetilde{\boldsymbol{f}}_i\left(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i\right)\\ &\quad + \sum_{j\in N_i}\boldsymbol{c}_{ij}\Big(\boldsymbol{x}_s(\theta_i)+\boldsymbol{Y}(\theta_i)\boldsymbol{R}_i,\boldsymbol{x}_s(\theta_j)+\boldsymbol{Y}(\theta_j)\boldsymbol{R}_j\Big)\Big).\end{aligned} \tag{19}$$

Multiplying to the left by $\boldsymbol{v}_1^T(\theta_i)$, using the bi-orthogonality condition, and rearranging terms, we obtain (14). Conversely, multiplying to the left by $\boldsymbol{Z}^T(\theta_i)$, and rearranging, we find (15).

□

The idea behind the decomposition is illustrated in figure 1. For the sake of clarity, we consider a second order oscillator, whose state can be represented on the plane. The solid black line represents the limit cycle $\boldsymbol{x}_s(t)$ of the unperturbed reference oscillator, while the red line represents the trajectory of the perturbed oscillator $\boldsymbol{x}_i(t)$. The arrows on the curves denote the orientation of the trajectories. The initial conditions of the unperturbed and perturbed systems are denoted by $\boldsymbol{x}_s(0)$ and $\boldsymbol{x}_i(0)$, respectively.

At each time instant t, the state of the perturbed system is represented as a time-shifted version of the reference limit cycle $\boldsymbol{x}_s(\theta_i(t))$ plus a deviation from the cycle, measured along the direction spanned by the vector $\boldsymbol{Y}(\theta_i(t))\,\boldsymbol{R}_i(t)$. Note that, for second order oscillators, $\boldsymbol{Y}(\theta_i(t))\,\boldsymbol{R}_i(t) = \boldsymbol{u}_2(\theta_i)\,R_i(t)$, and $R_i(t)$ is a scalar.

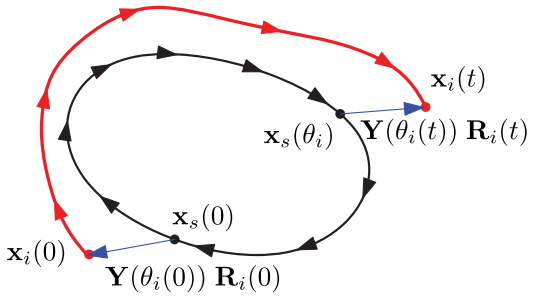


Figure 1. Illustration of the phase-amplitude decomposition for a second order oscillator. The explanation for the curves and the notation is given in the text.

Theorem 1 gives the equations governing the time evolution of the time shift $\theta_i(t)$, interpreted as the phase, and of the amplitude deviation $\boldsymbol{R}_i(t)$. These amplitude–phase equations are exact, since their derivation involves no approximations. Consequently, they are not simpler to solve than the original state equations. However, they can be integrated numerically and used as a benchmark to assess the accuracy of further reduced models.

The second step in the derivation of a phase reduced model is to show that, for practical applications, the phase equation (14) encodes the most relevant information. This is shown in two steps: First, we prove that the amplitude deviation inherits the stability property of the limit cycle. The following theorem is an adaptation of a result presented in [45, 46].

Theorem 2. *Consider the uncoupled oscillator* $\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}(t))$*, with a T-periodic limit cycle* $\boldsymbol{x}_s(t)$*. Let* $1, \mu_2, \ldots, \mu_n$ *be the characteristic multipliers of the periodic variational equation*

$$\frac{\mathrm{d}\boldsymbol{y}(t)}{\mathrm{d}t} = \boldsymbol{A}(t)\,\boldsymbol{y}(t)\,, \tag{20}$$

where $\boldsymbol{A}(t) = \boldsymbol{A}(t+T) = \dfrac{\partial \boldsymbol{f}(\boldsymbol{x}_s(t))}{\partial \boldsymbol{x}}$*. Then* $\boldsymbol{R} = 0$ *is an equilibrium point for the amplitude equation of the unperturbed system,*

$$\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}t} = \boldsymbol{L}(\theta)\,\boldsymbol{R} + \boldsymbol{a}_{\boldsymbol{R}}(\theta, \boldsymbol{R})\,, \tag{21}$$

with characteristic multipliers $\mu_2, \ldots, \mu_n$*.*

Proof. For $\varepsilon = 0$, the phase and amplitude deviation equations reduce to (index i is omitted for simplicity)

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = 1 + a_\theta(\theta, \boldsymbol{R}) \tag{22a}$$

$$\frac{\mathrm{d}\boldsymbol{R}}{\mathrm{d}t} = \boldsymbol{L}(\theta)\,\boldsymbol{R} + \boldsymbol{a}_{\boldsymbol{R}}(\theta, \boldsymbol{R})\,. \tag{22b}$$

Using (16), it is straightforward to verify that $a_\theta(\theta, 0) = 0$ and $\boldsymbol{a}_{\boldsymbol{R}}(\theta, 0) = 0$, thus $\boldsymbol{R} = 0$ is an equilibrium point for (21). Expanding of $\boldsymbol{f}(\boldsymbol{x})$ in Taylor series around $\boldsymbol{x}_s(\theta)$ and using $\boldsymbol{Z}^T(\theta)\boldsymbol{f}(\boldsymbol{x}_s(\theta)) = 0$ we obtain the amplitude deviation variational equation

$$\frac{\mathrm{d}\widetilde{\boldsymbol{R}}}{\mathrm{d}t} = \left(-\boldsymbol{Z}^T(\theta)\,\frac{\partial \boldsymbol{Y}(\theta)}{\partial \theta} + \boldsymbol{Z}^T(\theta)\,\boldsymbol{A}(\theta)\,\boldsymbol{Y}(\theta)\right)\widetilde{\boldsymbol{R}}. \tag{23}$$

Let $[\alpha_2, \ldots, \alpha_n]$ be a row vector of real constants. The matrix

$$\boldsymbol{\Phi}(\theta) = \left[\frac{\partial \boldsymbol{x}_s(\theta)}{\partial \theta} \qquad \frac{\partial \boldsymbol{x}_s(\theta)}{\partial \theta}\,[\alpha_2 \ldots \alpha_n] + \boldsymbol{Y}(\theta)\,\boldsymbol{\Psi}(\theta)\right], \tag{24}$$

is a fundamental matrix of the variational equation (20), if and only if $\boldsymbol{\Psi}(\theta)$ is a fundamental matrix of (23). Let $\mathbf{C} = \boldsymbol{\Phi}(T)$ be a principal matrix of system (20), then $\mathbf{C}$ has the structure[1]

$$\mathbf{C} = \begin{bmatrix} 1 & \mathbf{C}_2 \\ 0 & \mathbf{C}_1 \end{bmatrix}, \tag{25}$$

[1] The principal matrix $\mathbf{C}$ is not unique, but they are all similar.

where $\mathbf{C_2} \in \mathbb{R}^{1,n-1}$, and $\mathbf{C_1} \in \mathbb{R}^{n-1,n-1}$ has eigenvalues $\mu_2, \ldots, \mu_n$. Taking into account that the principal matrix satisfies $\mathbf{\Phi}(t+T) = \mathbf{\Phi}(t)\mathbf{C}$, multiplying to the left for $\boldsymbol{Z}^T(\theta)$ and using the periodicity of $\boldsymbol{x}_s(\theta)$ and $\boldsymbol{Y}(\theta)$ we have

$$\mathbf{\Psi}(\theta + T) = \mathbf{\Psi}(\theta)\mathbf{C_1} \tag{26}$$

that completes the proof. □

The consequence is that, for the coupled network, the phase is the most important variable. For an asymptotically stable limit cycle, $\mathrm{Re}[\mu_i] < 0$, $\forall i = 2, \ldots, n$. The amplitude deviation exhibits limited variations, and thus we can use the approximation $\boldsymbol{R}_i \approx 0$ for all $i = 1, \ldots, N$. Taking into account that $a_\theta(\theta, 0) = 0$, the phase equations take the form

$$\frac{\mathrm{d}\theta_i}{\mathrm{d}t} = 1 + \varepsilon \left(\delta a_\theta(\theta_i) + \sum_{j \in N_i} C_{ij}(\theta_i, \theta_j) \right), \tag{27}$$

where

$$\delta a_\theta(\theta_i) = \frac{\boldsymbol{v}_1^T(\theta_i)}{||\boldsymbol{f}(\boldsymbol{x}_s(\theta_i))||} \widetilde{\boldsymbol{f}}_i(\boldsymbol{x}_s(\theta_i)) \tag{28}$$

$$C_{ij}(\theta_i, \theta_j) = \frac{\boldsymbol{v}_1^T(\theta_i)}{||\boldsymbol{f}(\boldsymbol{x}_s(\theta_i))||} c_{ij}(\theta_i, \boldsymbol{R}_i, \theta_j, \boldsymbol{R}_j) \big|_{\boldsymbol{R}_i = \boldsymbol{R}_j = 0}. \tag{29}$$

Equation (27) represents the phase reduced model for the ONN (2). Instead of the oscillators' phases θ_i, it is often convenient considering the phase deviations (the deviation with respect to the phase of an unperturbed reference oscillator with null initial phase) $\psi_i = \theta_i - t$, obtaining:

$$\frac{\mathrm{d}\psi_i(t)}{\mathrm{d}t} = \varepsilon \left(\delta a_\theta(\psi_i + t) + \sum_{j \in N_i} C_{ij}(\psi_i + t, \psi_j + t) \right). \tag{30}$$

For small [38, 40–42] values of ε, the phase deviations ψ_i are nearly constants (slow variables), thus we can average over one period without introducing a big error, obtaining:

$$\frac{\mathrm{d}\psi_i}{\mathrm{d}t} \cong \varepsilon \left(\overline{\delta a}_\theta(\psi_i) + \sum_{j \in N_i} \overline{C}_{ij}(\psi_i - \psi_j) \right), \tag{31}$$

where

$$\overline{\delta a}_\theta(\psi_i) = \frac{1}{T} \int_0^T \delta a_\theta(\psi_i + t)\, \mathrm{d}t, \tag{32}$$

$$\overline{C}_{ij}(\psi_i - \psi_j) = \frac{1}{T} \int_0^T C_{ij}(\psi_i + t, \psi_j + t)\, \mathrm{d}t. \tag{33}$$

The dependence on the phase difference $\psi_i - \psi_j$ is justified as follows. Because θ_i is T-periodic, $\psi_i + t$ is T-periodic as well, for all $i = 1, \ldots, N$. Thus we can expand $C_{ij}(\psi_i + t, \psi_j + t)$ in double Fourier series

$$C_{ij}(\psi_i + t, \psi_j + t) = \sum_{m,n=-\infty}^{\infty} A_{mn}\, \mathrm{e}^{\mathrm{j}(m\psi_i + n\psi_j + (m+n)t)} = \sum_{m,n=-\infty}^{\infty} A_{mn}\, \mathrm{e}^{\mathrm{j}(m\psi_i + n\psi_j)} \mathrm{e}^{\mathrm{j}(m+n)t}. \tag{34}$$

Averaging over one period

$$\overline{C}_{ij}(\psi_i - \psi_j) = \frac{1}{T} \sum_{m,n=-\infty}^{\infty} A_{mn}\, \mathrm{e}^{\mathrm{j}(m\psi_i + n\psi_j)} \int_0^T \mathrm{e}^{\mathrm{j}(m+n)t} \mathrm{d}t = \sum_{m=-\infty}^{\infty} A_{m,-m}\, \mathrm{e}^{\mathrm{j}m(\psi_i - \psi_j)}. \tag{35}$$

Remark: It is well known that if $\boldsymbol{x}_s(t)$ is a solution of the nonlinear ODE $\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{f}(\boldsymbol{x}(t))$, then $\boldsymbol{f}(\boldsymbol{x}_s(t))$ is a solution of the variational equation $\frac{\mathrm{d}\boldsymbol{y}(t)}{\mathrm{d}t} = \boldsymbol{A}(t)\boldsymbol{y}(t)$. Moreover, if $\boldsymbol{q}(t)$ is a solution of the adjoint equation $\frac{\mathrm{d}\boldsymbol{z}(t)}{\mathrm{d}t} = -\boldsymbol{A}(t)^T\boldsymbol{q}(t)$, then

$$\boldsymbol{q}(t)^T \boldsymbol{f}(\boldsymbol{x}_s(t)) = \text{constant}. \tag{36}$$

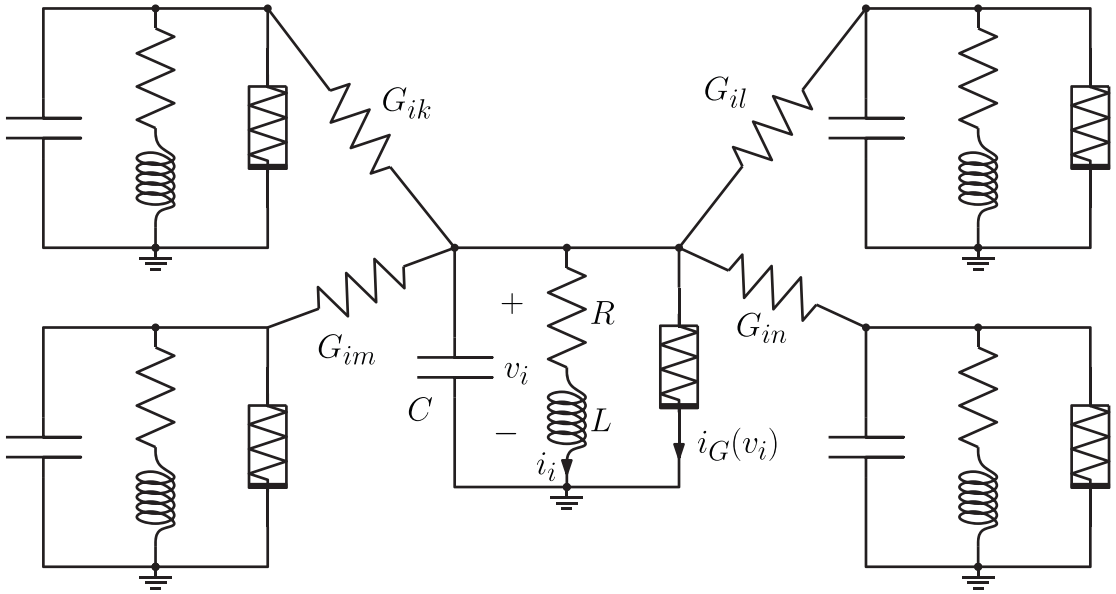


Figure 2. Portion of a network of nonlinear oscillators.

Remark: The weak-coupling assumption used for the derivation of the phase reduced model, should be interpreted in relation to the stability of the underlying limit cycle. In particular, the coupling must be sufficiently small so that the limit cycle of each uncoupled oscillator does not undergo bifurcation due to the interaction. The stability of the limit cycle can be quantified through the Floquet multipliers, i.e. the eigenvalues of the matrix $\mathbf{C}_1$ introduced in Theorem 2. These quantities characterize the transversal stability of the periodic orbit and provide a quantitative criterion to assess the admissible coupling strength. In this respect, the proposed amplitude–phase decomposition offers an additional advantage, as it naturally yields the matrix $\mathbf{C}_1$ and thus provides a direct way to evaluate the stability margin of the limit cycle and the validity of the weak-coupling assumption.

During the training of the network, the coupling strength between oscillators is modified, which may raise the question of whether the weak-coupling hypothesis remains valid. However, electronic oscillators generally exhibit very stable limit cycles, allowing relatively large coupling strengths without affecting their qualitative dynamics, while synchronization is typically achieved with much smaller couplings. Consistently with this observation, in all our numerical simulations we did not observe any limit-cycle bifurcation.

2.3. Application to a ONN of nonlinear electrical circuits

As an example of derivation of the phase deviation equation for an ONN, we consider a network of weakly connected electrical circuits. A portion of the network is sketched in figure 2. Applying Kirchhoff laws to the ith node, we obtain the equations

$$\frac{\mathrm{d}v_i}{\mathrm{d}t} = \frac{1}{C}\left(-i_i - i_G(v_i) - \varepsilon \sum_{j\in N_i} G_{ij}\left(v_i - v_j\right)\right) \tag{37}$$

$$\frac{\mathrm{d}i_i}{\mathrm{d}t} = \frac{1}{L}\left(v_i - Ri_i\right). \tag{38}$$

We shall assume[2] $i_G(v_i) = -g_a v_i + g_b v_i^3$. Introducing the adimensional time $t' = \frac{R}{L}t$ and the voltages $x_i = v_i$, $y_i = Ri_i$ the state equations can be rewritten in the form

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \beta x_i - \alpha y_i - \gamma x_i^3 - \varepsilon \sum_{j\in N_i} \kappa_{ij}\left(x_i - x_j\right) \tag{39}$$

$$\frac{\mathrm{d}y_i}{\mathrm{d}t} = x_i - y_i \tag{40}$$

[2] A cubic nonlinear current–voltage characteristic is a common assumption in nonlinear circuits, i.e. the Chua's circuit, and can be readily realized using operational amplifier–based circuits, as in the Chua's diode.

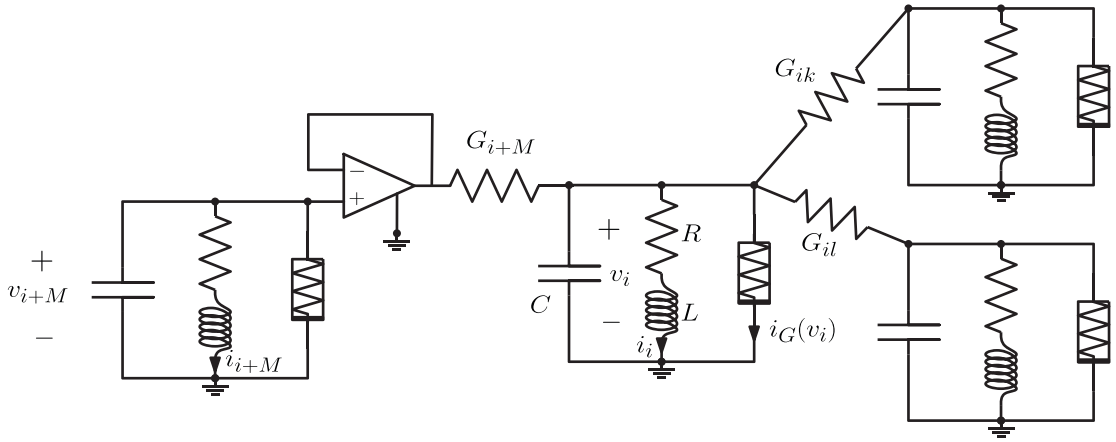


Figure 3. Portion of a network of nonlinear oscillators with a master-slave connection.

where $\alpha = L/(R^2C)$, $\beta = g_aL/(RC)$, $\gamma = g_bL/(RC)$, and $\kappa_{ij} = LG_{ij}/(RC)$.

The limit cycle of each uncoupled node can be conveniently approximated through the Describing Function technique [35]. The following first harmonic approximation of the limit cycle is found:

$$\boldsymbol{x}_s(t) = \begin{pmatrix} A\sin(\omega t) \\ \frac{1}{\alpha}\left(\beta - \frac{3\gamma}{4}A^2\right)A\sin(\omega t) - \frac{\omega}{\alpha}A\cos(\omega t) \end{pmatrix} \tag{41}$$

where $A = \sqrt{\frac{4}{3\gamma}(\beta - 1)}$ and $\omega = \sqrt{\alpha - 1}$.

The first harmonic approximation for the solution of the adjoint equation is

$$\boldsymbol{q}(t) = \begin{pmatrix} D\cos(\omega t) \\ \omega D\sin(\omega t) + \left(\frac{3\gamma}{4}A^2 - \beta\right)D\cos(\omega t) \end{pmatrix}, \tag{42}$$

where the constant $D = \frac{\alpha}{\omega A}$ is determined through the normalization condition

$$\boldsymbol{q}^T(t)\boldsymbol{f}(\boldsymbol{x}_s(t)) = 1. \tag{43}$$

A straightforward application of the theory developed in the previous section gives the phase deviation equation

$$\frac{\mathrm{d}\psi_i}{\mathrm{d}t} = \varepsilon\frac{\alpha}{2\omega}\sum_{j\in N_i}\kappa_{ij}\sin\left(\psi_i - \psi_j\right). \tag{44}$$

Remark: figure 3 shows a portion of a network implementing the special model described by (6) and (7).

The corresponding phase deviation equation takes the form

$$\frac{\mathrm{d}\psi_i}{\mathrm{d}t} = \varepsilon\frac{\alpha}{2\omega}\left(\sum_{j\in N_i}\kappa_{ij}\sin\left(\psi_i - \psi_j\right) + \kappa_i\sin\left(\psi_i - \psi_{i+M}\right)\right) \tag{45}$$

$$[1ex]\ \frac{\mathrm{d}\psi_{i+M}}{\mathrm{d}t} = 0 \Rightarrow \psi_{i+M}(t) = \psi_{i+M}(0) \tag{46}$$

where $\kappa_i = LG_{i,i+M}/(RC)$.

2.4. EP training

We now illustrate an application-oriented example of a weakly connected ONN trained via the EP algorithm, applied to the energy-based model (EBM) representation of the oscillator dynamics [28, 29, 32, 47]. The objective is to demonstrate how the EBM framework can be used to define and study a numerically specified oscillator network whose equilibrium states are associated with prescribed target

configurations. Such a configuration provides a convenient supporting example for discussing de-noising and classification of input phase states.

EBMs associate a scalar energy function to the network's phase configuration. In the EBMs framework, inference is defined as the recording of the phase-state $\boldsymbol{\psi} = \psi_1, \ldots, \psi_N$ of the system at equilibrium (i.e. asymptotically after the system governed by (44) has relaxed to steady state).

For consistency with the machine learning literature, we define the coupling weight matrix as:

$$W = \operatorname{set} w_{ij} = \operatorname{set}\left\{\varepsilon, \frac{\alpha}{2\omega}, \kappa_{ij}\right\}, \tag{47}$$

where indices i and j denote oscillator pairs, under the assumption of a symmetric W matrix.

Based on these definitions, the system in (44) can be recast as a gradient system with respect to the slow-time variable τ associated with the phase dynamics, and an energy function $E(\boldsymbol{\psi}, \boldsymbol{W})$, as follows:

$$\frac{\mathrm{d}\psi_i}{\mathrm{d}\tau} = -\frac{\partial E}{\partial \psi_i}(\boldsymbol{\psi}, \boldsymbol{W}), \tag{48}$$

with

$$E(\boldsymbol{\psi}, \boldsymbol{W}) = -\frac{1}{2}\sum_{i,j=1}^{N} w_{ij}\cos\left(\psi_i - \psi_j\right). \tag{49}$$

The equilibria of (44) correspond to the minima of (49).

Learning consists in minimizing a scalar cost function $C(\boldsymbol{T}, \boldsymbol{\psi})$ that quantifies the distance between the target $\boldsymbol{T}$ and the equilibrium phase configuration $\boldsymbol{\psi}$ of the network:

$$C(\boldsymbol{T}, \boldsymbol{\psi}) = \frac{1}{N}\sum_{i=1}^{N}\left(\cos\left(T_i\right) - \cos\left(\psi_i\right)\right). \tag{50}$$

Following the formulation proposed in [32], we introduce an augmented energy function:

$$F\left(\boldsymbol{\psi}^{\beta}, \boldsymbol{T}, \boldsymbol{W}, \beta\right) = E\left(\boldsymbol{\psi}^{\beta}, \boldsymbol{W}\right) + \beta, C\left(\boldsymbol{T}, \boldsymbol{\psi}^{\beta}\right), \tag{51}$$

where β is a small forcing parameter. In light of (51), the gradient system in (48) can be reformulated as

$$\frac{\mathrm{d}\psi_i^{\beta}}{\mathrm{d}\tau} = -\frac{\partial F}{\partial \psi_i^{\beta}}\left(\boldsymbol{\psi}^{\beta}, \boldsymbol{T}, \boldsymbol{W}, \beta\right). \tag{52}$$

As demonstrated in [47], the EP update rule for the weight values w_{ij}—in a network modeled using a simplified Kuramoto-type oscillator system as in (44)—can be approximated as:

$$\Delta w_{ij} \approx -\eta \frac{\mathrm{d}}{\mathrm{d}\beta}\left[\frac{\partial F}{\partial W_{ij}}\left(\boldsymbol{\psi}^{\beta}, \boldsymbol{T}, \boldsymbol{W}, \beta\right)\right]_{\beta=0} = \eta \frac{\mathrm{d}}{\mathrm{d}\beta}\cos\left(\psi_i^{\beta} - \psi_j^{\beta}\right)\Big|_{\beta=0} \tag{53}$$

$$= \eta \frac{\cos\left(\psi_i^{\beta} - \psi_j^{\beta}\right) - \cos\left(\psi_i - \psi_j\right)}{\beta}, \quad \text{with } \beta \simeq 0. \tag{54}$$

In practical terms, the training algorithm proceeds as follows:

1. Initialize the network state as $\boldsymbol{\psi}_{t=0} = \boldsymbol{T} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is a small perturbation that initiates the convergence of the system toward the nearest equilibrium state.
2. Allow the network to relax and converge to its equilibrium $\boldsymbol{\psi}_{t\to\infty}$; record the phase deviations ψ_i and the corresponding gradient terms $\cos(\psi_i - \psi_j)$.
3. Apply a small perturbation of amplitude $-\beta, \nabla_{\boldsymbol{\psi}} C(\boldsymbol{T}, \boldsymbol{\psi})$ to the right-hand side of (44).
4. Let the network relax to the new equilibrium according to the perturbed system, and record the updated phase deviations ψ_i^{β} and gradient terms $\cos(\psi_i^{\beta} - \psi_j^{\beta})$.
5. Update the weight matrix $\Delta \boldsymbol{W}$ according to (54).

The Kuramoto-type model adopted in the following simulations should be regarded as a representative instance of the general phase-reduced framework derived above. Its role is to provide a minimal and analytically tractable setting in which the connection between phase dynamics, energy structure, and EP could be explicitly verified. While more complex oscillator models could be considered within the same framework, they would not alter the core analytical results, and are therefore not pursued in this work.

3. Results

A set of ten hand-drawn digits (digits 0–9) was generated on a 16×16 raster, subsequently binarized and mapped to phase vaues in $\{0, \pi\}$, as shown in figure 4(a). The dataset was constructed to include both visually distinct patterns and partially overlapping traits, especially among digits 6, 8, and 9, so as to test retrieval under both separable and non-orthogonal conditions.

The oscillator network, comprising $N = 256$ nodes, was initialized with an initial weight matrix W_{init} drawn from a uniform random distribution within the range $[-10^{-4}, 10^{-4}]$.

After generation, the initial weight matrix W_{init} is symmetrized, and all self-coupling terms (i.e. diagonal elements) are set to zero. Training is performed sequentially by presenting the entire dataset to the network ten times per epoch, for a total of 250 epochs. The hyperparameters $\eta = 10^{-4}$ and $\beta = 0.1$ were selected based on preliminary exploratory studies.

The classification accuracy is defined by assigning a successful recognition event whenever, for a given target $\boldsymbol{T}_i$, the Euclidean distance $|f(\boldsymbol{T}_i) - f(\boldsymbol{\psi}_i)|$ equals zero, where $f(\nu) = \text{sign}(\cos(\nu))$. This transformation mitigates numerical artifacts and minimizes deviations due to spurious resonances.

Test inputs $\boldsymbol{\psi}_{t=0}$ are generated using two distinct perturbation strategies. In the *Gauss* method, additive Gaussian noise with standard deviation $\sigma = 1$ is applied to the phase vector. In the *Flip* method, individual phase components are randomly inverted with probability $p = 0.1$.

The system accuracy is evaluated after each training epoch using a set of 200 test samples. The classification accuracies of the *Flip* and *Gauss* perturbation methods across training epochs are shown as continuous lines in figure 4(b). Both metrics exhibit a rapid convergence towards values exceeding 90% accuracy, maintaining this performance level for a substantial range of epochs. A peak classification accuracy of 100% is achieved for both perturbation schemes at 128 epochs, while average accuracies above 98% are consistently observed starting from approximately 54 epochs. Accuracy measured with both methods then degrades after roughly 100 epochs. This late-stage degradation is consistent with the emergence of hybrid attractors and secondary phase-cluster splitting, illustrated in figure 5, and discussed later. These results show that the proposed phase-based EBM framework, combined with EP training, can support denoising and pattern-recognition tasks in the simulated setting. While no hardware-level cost model is evaluated here, the reliance on local interactions and simple phase dynamics suggests potential suitability for efficient hardware implementation.

Furthermore, these findings underline the importance of monitoring the training results during training. In this study, no adaptive scheduling of the learning rate η was employed, in order to explicitly illustrate this phenomenon.

To interpret the observed accuracy trends and the underlying retrieval mechanism, and to identify potential pathways toward improved training algorithms, we next examine the transient phase evolution for a representative test sample. An example of the temporal evolution of the system is presented in figure 5. A noisy instance of the digit '2' (generated via the *Gauss* method with an augmented standard deviation $\sigma = 3$, for demonstration purposes) is randomly generated and applied as input to the network. The initial phase distribution is shown in the ψ row, first column, where ψ denotes the phase deviation of each oscillator.

The temporal evolution[3] of the phase deviations is visualized on a cylindrical surface, where the polar coordinate corresponds to the instantaneous phase deviation value, and the longitudinal coordinate represents time progression. Each oscillator trajectory is color-coded according to the spatial legend displayed on the left. The row ψ illustrates the phase configuration in the indicated time frames.

As observed, during the initial stages of the system evolution ($t \leqslant 10$), only a limited number of oscillators exhibit noticeable phase variations. A pronounced collective phase shift occurs for $t > 100$, leading to a clearly distinguishable solution emerging at approximately $t = 10^3$. The system reaches a stable steady state around $t = 10^4$, beyond which no significant phase changes are detected.

[3] In this work, time is treated as a dimensionless variable; however, for clarity, in the simulations it represents the number of oscillator periods.

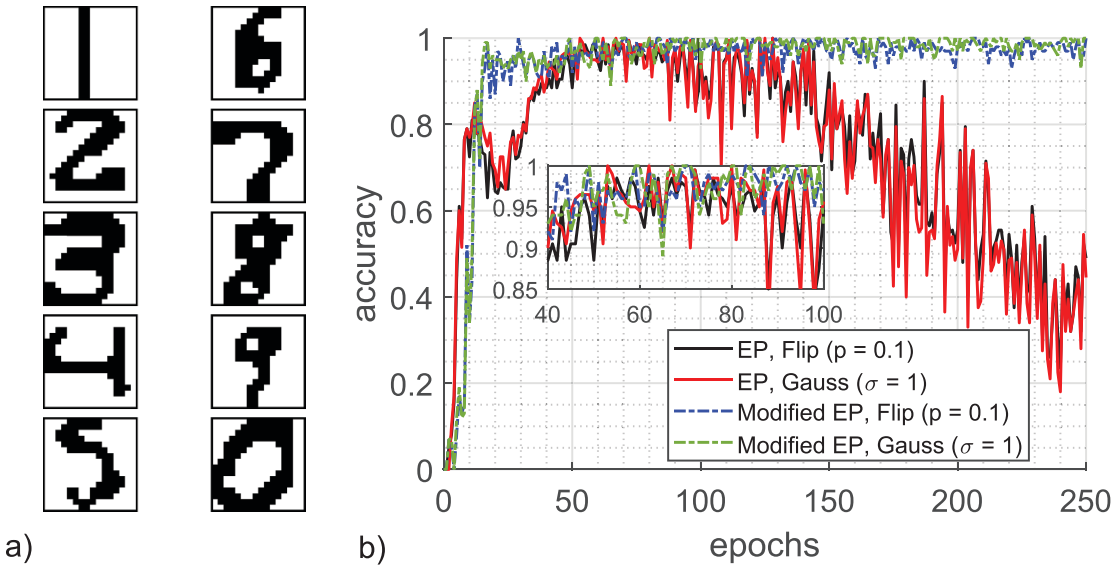


Figure 4. Training dataset and results. (a) The 16×16 target dataset, where black pixels correspond to a phase deviation $\psi = 0$ and white pixels correspond to a phase $\psi = \pi$. (b) Training accuracy as a function of the number of epochs. Continuous lines represent the test accuracy of the system trained using the EP algorithm (black: *Flip* perturbation, red: *Gauss* perturbation). Dash-dotted lines represent the system test accuracy for the system trained with the modified EP algorithm (blue: *Flip* perturbation, green: *Gauss* perturbation).

At this stage, both the cylindrical phase plot and the corresponding two-dimensional phase raster reveal two distinct phase clusters centered around $0 \equiv 2\pi$ and π. The solution obtained at $t = 10^4$ visually matches the unperturbed reference sample. The system's classification output (or 'guess') is derived from the sign of the phase cosine, depicted in the second row. It is evident that a matching system guess can already be obtained for $t \lesssim 10^3$, producing an almost noise-free reconstruction of the input digit. In most successful recognition events, the inferred state of the system remains stable over time once convergence is achieved.

However, the network equilibrium does not always exhibit perfectly binarized phase clustering. In many cases, the equilibrium phase distribution displays $2n$ distinct clusters. This behavior is consistent with the emergence of spurious attractors, well documented in associative memory models [48–51], arising from cross-pattern interference, particularly in non-orthogonal datasets. This effect adversely impacts classification accuracy.

Figure 6 represents the behavior of the network under prolonged training and exhibiting low accuracy, where this phenomenon becomes more pronounced. To highlight the phase deviations, the cylindrical phase plot from the previous example is unwrapped onto a rectangular surface. In this case, the input perturbation follows the *Flip* method, with a flipping probability of $p = 0.2$ (used here for demonstration purposes).

The resulting phase plot reveals that the network initially converges toward, binarized clustering of the phase vector—similar to that observed in figure 5. As time progresses beyond $t \geqslant 10^4$, each cluster undergoes a secondary split, with a subset of oscillators crossing the $(1 + 2n)\pi/2$ boundaries. This phase transition reverses the cosine sign for these elements, leading to an incorrect recognition outcome.

The phase rasters indicate that the system first converges toward the correct attractor (digit '6'), but subsequently drifts toward a mixture of stored patterns, here involving features of digits '4' and '1'. The resulting equilibrium therefore corresponds to a hybrid attractor state rather than a stored pattern.

These results emphasize that, within such oscillator networks, the intrinsic meta-stability of the targeted equilibria may lead to convergence toward hybrid or mixed states. Further analysis indicates that the degradation observed under prolonged training is primarily associated with oscillators converging near the decision boundary of the readout $\hat{x}_i = \mathrm{sign}(\cos\psi_i)$, which reduces robustness to perturbations. To mitigate this effect while preserving locality of the learning rule, we implemented a stabilization mechanism combining an Oja-type activity–dependent decay [52] with synaptic scaling [53]. We refer to this learning rule as *Modified EP*. Such homeostatic mechanisms are widely recognized as essential for stabilizing Hebbian learning in recurrent networks and preventing runaway weight growth [54].

The amended update modifies the EP increment according to

$$\Delta W_{ij} \leftarrow \Delta W_{\mathrm{EP}} - \gamma \cos\left(\psi_i\right)^2 W_{ij}. \tag{55}$$

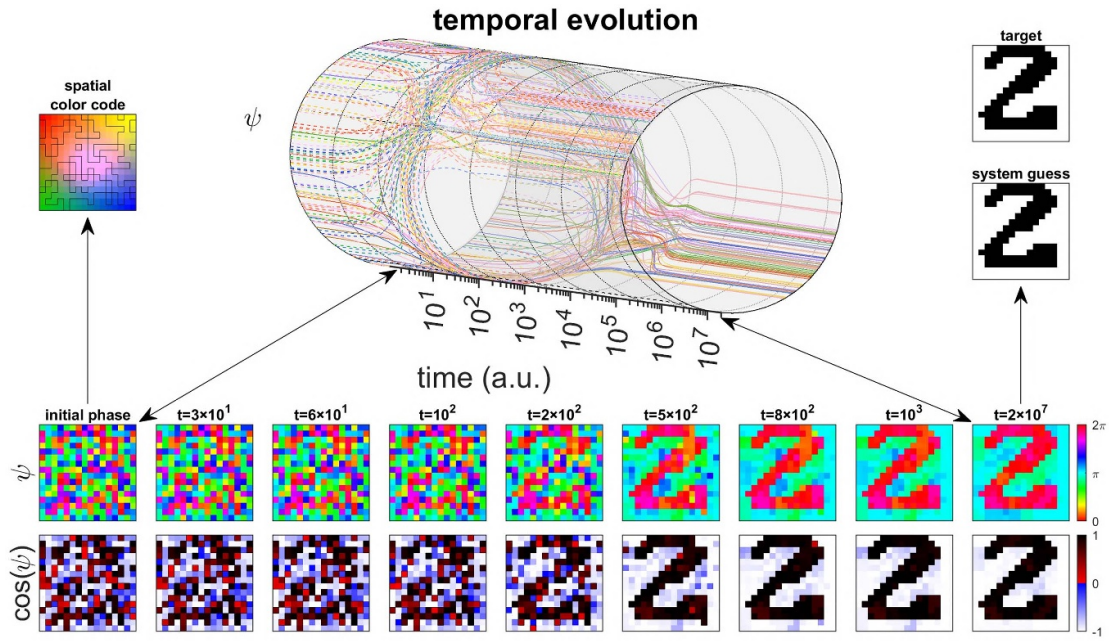


Figure 5. Phase-set evolution for a trained 16 × 16 oscillator network. The initial system state corresponds to a noisy instance of the digit '2' (ψ row, first column). The temporal evolution of the oscillator phases is depicted in the central cylindrical plot, where each trajectory follows a color code according to the spatial legend shown on the left. Note that the time axis is represented on a logarithmic scale. Dashed and continuous curves trace the evolution of the oscillators toward their expected steady states, with dashed lines converging to the phase deviation value ψ and continuous lines converging to zero. The corresponding phase raster plots are shown for selected time frames in the ψ row. Below this, the cosine of the phase distribution is displayed to illustrate the numerical data from which the system's recognition output is derived. The target and inferred (guessed) states are also displayed on the right as two-dimensional spatial maps for direct comparison.

This Oja-like term introduces an activity-dependent decay that counteracts the reinforcement of strongly active synapses and limits the progressive amplification of correlated modes.

After the EP update, a synaptic scaling step is applied

$$W_{i:} \leftarrow s_i W_{i:}, \qquad s_i = \frac{\rho_i}{\|W_{i:}\|_2}, \tag{56}$$

where ρ_i is a reference row norm measured during early training. Synaptic scaling preserves the relative structure of synaptic weights while maintaining bounded total input to each unit. Together, these two local mechanisms stabilize the training dynamics by limiting weight growth and indirectly constraining spectral expansion of the coupling matrix, thereby preventing the late-stage accuracy degradation observed with the baseline EP rule (see dash-dotted curves in figure 4).

3.1. Validation on MNIST-derived dataset

To complement the controlled denoising experiments based on synthetic perturbations of stored phase patterns, we evaluated the proposed framework on a subset of the MNIST dataset. In order to preserve consistency with the associative-retrieval setting considered in this work, the benchmark was adapted as follows. For each digit class, a representative target was selected as the sample closest to the class mean in Euclidean distance. Training and test datasets were then constructed by selecting the closest class-consistent samples to this prototype (100 samples per class).

Unlike the synthetic perturbation setting, these inputs are not generated by adding controlled noise to the targets but correspond to naturally varying handwritten realizations. As a result, the benchmark represents a more heterogeneous and less controlled retrieval problem.

The training process (through the *Modified EP* rule) and evaluation results are shown in figure 7. The system achieves stable accuracy, with mean values calculated after 250 epochs of 91.16% on the training set and 90.26% on the test set (data is calculated testing 400 samples per datapoint). The confusion matrix, computed using the full test dataset at the end of the training, yields an overall classification accuracy of 88.40%.

To quantitatively assess the characteristic of this benchmark, we measured the mismatch between input patterns $\boldsymbol{\psi}_i$ and their corresponding target prototypes $\boldsymbol{T}_i$, using the cosine-domain

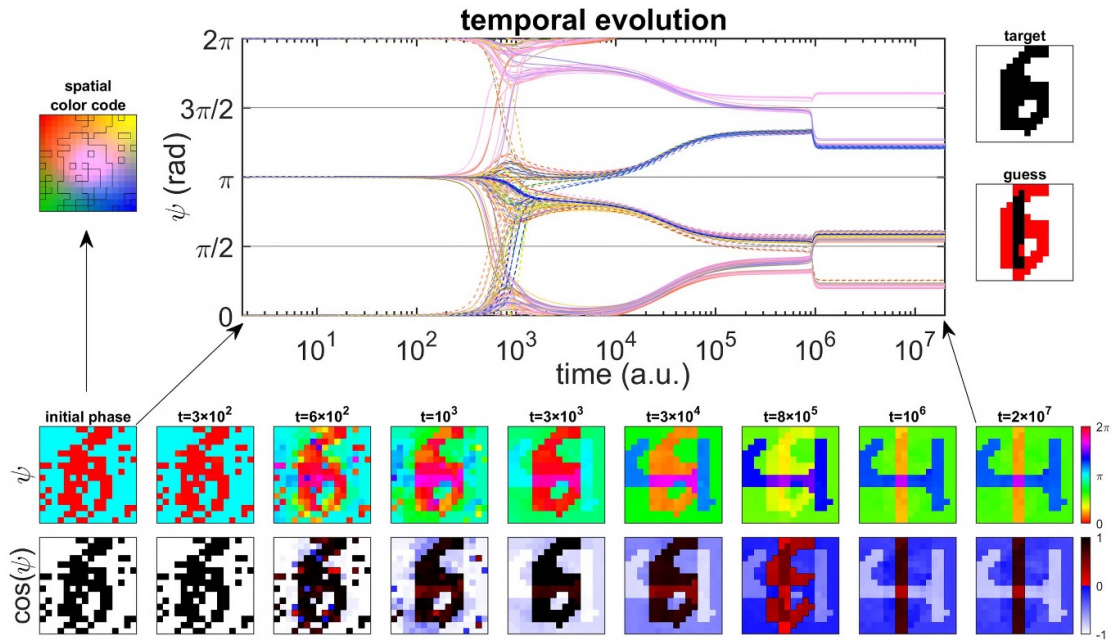


Figure 6. Phase set evolution for a 16 × 16 oscillator system subjected to prolonged training, shown using the same graphical representation adopted in figure 5. Here, the cylindrical phase plot is unwrapped and presented on its corresponding rectangular surface, allowing clearer visualization of phase deviations and cluster bifurcations occurring during the system's temporal evolution.

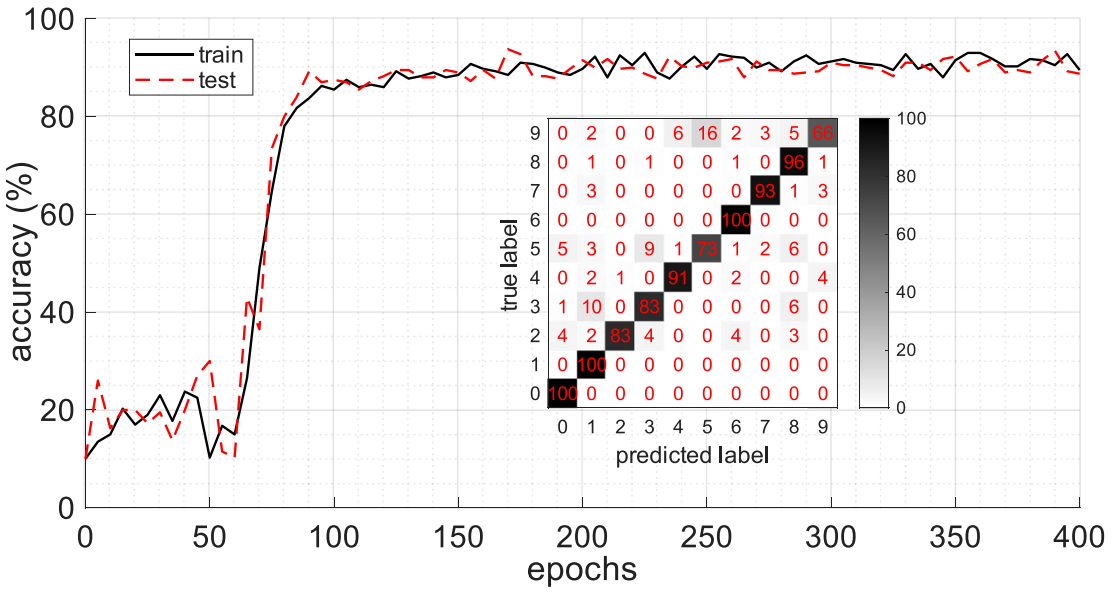


Figure 7. Training through the *Modified EP* rule, and evaluation, on the MNIST-derived dataset. Main plot: classification accuracy vs epochs, evaluated on a subset of 400 samples for both training and test datasets, for each datapoint. The system achieves approximately 91% training accuracy and 90% test accuracy after 250 epochs. Inset: confusion matrix computed over the full-test dataset, at the end of the training simulation; the resulting accuracy is 88.40%.

Euclidean distance $d(\boldsymbol{\psi}, \boldsymbol{T}) = \frac{1}{N}\|\cos(\boldsymbol{\psi}) - \cos(\boldsymbol{T})\|_2^2$, and the Hamming mismatch $h(\boldsymbol{\psi}, \boldsymbol{T}) = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\{\text{sign}(\cos(\boldsymbol{\psi}_i)) \neq \text{sign}(\cos(\boldsymbol{T}_i))\}_2^2$.

The MNIST training set exhibits a mean distance $|d| \approx 0.39$, comparable to the Flip perturbation case ($|d| \approx 0.40$), while the MNIST test set shows a significantly larger value ($|d| \approx 0.48$). In addition, the variability is substantially higher for the MNIST datasets (standard deviation $\sigma(d) \approx 0.15$ and $\sigma(d) \approx 0.17$, for training and test datasets respectively) compared to both of the synthetic setting ($\sigma(d) < 0.07$). The data measured for the Hamming mismatch report qualitatively similar features with largely unaltered ratios between datasets.

These results indicate that the MNIST benchmark constitutes a more challenging retrieval problem due to the intrinsic intra-class variability of handwritten digits. Despite this increased mismatch and dispersion, the network maintains comparable performance on training and test sets, indicating that the learned attractor structure generalizes across intra-class variations, rather than overfitting the prototype-based training set. This behavior is consistent with the energy-based interpretation of the model, where one expects learning shapes attractor basins (associated with class prototypes) rather than memorizing individual samples.

4. Conclusions

In this work, we presented a methodological framework for the analytical design of ONN circuitry. Rather than relying solely on numerical optimization or heuristic modeling, we demonstrated that the core mechanisms governing synchronization, adaptation, and learning in ONNs can be formally derived from first principles using phase deviation equations. By linking these analytical constructs with the EP learning rule, the proposed approach enables a systematic transition from dynamical modeling to physically realizable neuromorphic architectures.

The phase deviation equations derived in this work reduce the complexity of the full nonlinear system while preserving its essential coupling structure and stability characteristics. The number of ODEs required to describe the network is thereby reduced to a single ODE per node. Moreover, synchronous oscillations within the network are mapped to equilibrium points of the phase deviation equations. This reduction and simplification enable designers to analytically predict equilibrium configurations, synchronization conditions, and the influence of coupling parameters before resorting to numerical simulations or circuit-level implementations. Consequently, the proposed framework provides a direct pathway for the systematic design and optimization of ONN hardware through controllable parameters such as coupling strength, phase response, and network topology.

From an engineering perspective, the proposed methodology reframes ONNs modeling as a design-oriented problem. Beginning with the specification of the network topology–namely, the differential equations that define each node's internal dynamics and their interconnections–the corresponding phase deviation equation and the associated energy function can be analytically derived. The analytical form of the energy landscape not only enables the prediction of the network's convergence and stability properties, but also provides a principled foundation for designing ONNs as specialized hardware solvers analogous to Ising machines. By appropriately configuring the interconnection strengths, the energy landscape can be modulated such that its minima correspond to the solutions of complex combinatorial optimization problems, pattern recognition, associative memories, and classification tasks. In this regard, the proposed framework goes beyond a descriptive model of ONNs dynamics; it prescribes how ONNs can be designed and trained to realize targeted equilibrium states. This distinguishing feature sets the present work apart from purely data-driven or simulation-based approaches, grounding neuromorphic circuit design in analytical rigor and predictability.

The results obtained through EP-based and modified EP-based training[4] confirm this interpretation. The system demonstrates rapid convergence toward stable attractors and exhibits resilience to perturbations, confirming that the analytical energy formulation accurately captures the underlying dynamics of the physical system. The correspondence between the minima of the energy function and the steady-state phase configurations further demonstrates that circuit parameters can be directly optimized within the analytical domain, thereby reducing reliance on extensive numerical simulations and iterative hardware prototyping.

However, the aforementioned results are not without trade-offs. Although phase-only reduction enhances analytical tractability, it neglects higher-order amplitude interactions that may influence convergence in strongly coupled or heterogeneous networks. Future research should aim to extend the formalism by incorporating adaptive phase–amplitude coupling and stochastic effects, thus broadening the scope of circuits that can be designed through analytical methodologies.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

[4] The data and data-processing scripts used in this work are available in the accompanying GitHub repository: https://github.com/EmanueleGemo/ONN-training-with-EP

Author contributions

Emanuele Gemo 0000-0001-8349-6627
Formal analysis (equal), Methodology (supporting), Investigation (lead), Software (lead), Validation (lead), Visualization (lead), Writing – original draft (equal), Writing – review & editing (equal)

Michele Bonnin 0000-0002-9106-563X
Conceptualization (equal), Formal analysis (equal), Methodology (lead), Software (supporting), Writing – original draft (equal), Writing – review & editing (equal)

Fernando Corinto 0000-0003-4431-5701
Conceptualization (equal), Formal analysis (equal), Funding acquisition (lead), Project administration (lead), Supervision (lead), Writing – original draft (equal), Writing – review & editing (equal)

References

[1] Gooding M 2024 Global Data Center electricity use to double by 2026 - IEA report (available at: www.datacenterdynamics.com/en/news/global-data-center-electricity-use-to-double-by-2026-report/)
[2] Csaba G and Porod W 2020 Coupled oscillators for computing: a review and perspective *Appl. Phys. Rev.* **7** 011302
[3] Mohseni N, McMahon P L and Byrnes T 2022 Ising machines as hardware solvers of combinatorial optimization problems *Nat. Rev. Phys.* **4** 363–79
[4] Todri-Sanial A, Delacour C, Abernot M and Sabo F 2024 Computing with oscillators from theoretical underpinnings to applications and demonstrators *npj Unconv. Comput.* **1** 14
[5] Kuznetsov N, Mokaev T, Ponomarenko V, Seleznev E, Stankevich N and Chua L 2023 Hidden attractors in Chua circuit: mathematical theory meets physical experiments *Nonlinear Dyn.* **111** 5859–87
[6] Roychowdhury J 2015 Boolean computation using self-sustaining nonlinear oscillators *Proc. IEEE* **103** 1958–69
[7] Shukla N, Tsai W-Y, Jerry M, Barth M, Narayanan V and Datta S 2016 Ultra low power coupled oscillator arrays for computer vision applications *2016 IEEE Symp. on VLSI Technology* pp 1–2
[8] Yu Y, Han F and Wang Q 2022 Exploring phase–amplitude coupling from primary motor cortex-basal ganglia–thalamus network model *Neural Netw.* **153** 130–41
[9] Wu F, Meng H and Ma J 2024 Reproduced neuron-like excitability and bursting synchronization of memristive josephson junctions loaded inductor *Neural Netw.* **169** 607–21
[10] Gosti G, Milanetti E, Folli V, de Pasquale F, Leonetti M, Corbetta M, Ruocco G and Penna S D 2024 A recurrent Hopfield network for estimating meso-scale effective connectivity in meg *Neural Netw.* **170** 72–93
[11] Shahbazi H, Parandeh R and Jamshidi K 2016 Implementation of imitation learning using natural learner central pattern generator neural networks *Neural Netw.* **83** 94–108
[12] Breve F A, Zhao L, Quiles M G and Macau E E 2009 Chaotic phase synchronization and desynchronization in an oscillator network for object selection *Neural Netw.* **22** 728–37
[13] Ferrari F A, Viana R L, Lopes S R and Stoop R 2015 Phase synchronization of coupled bursting neurons and the generalized Kuramoto model *Neural Netw.* **66** 107–18
[14] Mathiyalagan K, Anbuvithya R, Sakthivel R, Park J H and Prakash P 2016 Non-fragile H_{∞} synchronization of memristor-based neural networks using passivity theory *Neural Netw.* **74** 85–100
[15] Delacour C, Carapezzi S, Abernot M, Boschetto G, Azemard N, Salles J, Gil T and Todri-Sanial A 2021 Oscillatory neural networks for edge AI computing *2021 IEEE Computer Society Annual Symp. on VLSI (ISVLSI)* pp 326–31
[16] Corti E, Jimenez J A C and Moselund K E 2021 Coupled vo_2 oscillators circuit as analog first layer filter in convolutional neural networks *Front. Neurosci.* **15** 628254
[17] Abernot M, Amara H, Gil T and Todri-Sanial A 2022 Oscillatory neural network for edge computing: a mobile robot obstacle avoidance application *2022 IEEE Int. Conf. on Metrology for Extended Reality, Artificial Intelligence and Neural Engineering (MetroXRAINE)* pp 181–6
[18] Ahmed I, Chiu P-W, Moy W and Kim C H 2021 A probabilistic compute fabric based on coupled ring oscillators for solving combinatorial optimization problems *IEEE J. Solid-State Circuits* **56** 2870–80
[19] Núñez J, Avedillo M J, Jiménez M, Quintana J M, Todri-Sanial A, Corti E, Karg S and Linares-Barranco B 2021 Oscillatory neural networks using vo_2 based phase encoded logic *Front. Neurosci.* **15** 655823
[20] Parihar A, Shukla N, Jerry M, Datta S and Raychowdhury A 2017 Vertex coloring of graphs via phase dynamics of coupled oscillatory networks *Sci. Rep.* **7** 911
[21] Delacour C, Haverkort B, Sabo F, Azemard N and Todri-Sanial A 2025 Lagrange oscillatory neural networks for constraint satisfaction and optimization *Neuromorph. Comput. Eng.* **5** 044004
[22] Abernot M, Gil T, Jiménez M, Núñez J, Avellido M J, Linares-Barranco B and Todri-Sanial A 2021 Digital implementation of oscillatory neural network for image recognition applications *Front. Neurosci.* **15** 713054
[23] Sharma D *et al* 2024 Linear symmetric self-selecting 14-bit kinetic molecular memristors *Nature* **633** 560–6
[24] Ricci S, Kappel D, Tetzlaff C, Ielmini D and Covi E 2023 Tunable synaptic working memory with volatile memristive devices *Neuromorph. Comput. Eng.* **3** 044004
[25] Itoh M and Chua L 2008 Memristor oscillators *Int. J. Bifurcation Chaos* **18** 3183–206
[26] Abdurahman A and Jiang H 2016 New results on exponential synchronization of memristor-based neural networks with discontinuous neuron activations *Neural Netw.* **84** 161–71
[27] Yang S, Li C and Huang T 2016 Exponential stabilization and synchronization for fuzzy model of memristive neural networks by periodically intermittent control *Neural Netw.* **75** 162–72

[28] Zoppo G, Marrone F, Bonnin M and Corinto F 2022 Computing with memristor-based nonlinear oscillators *2022 IEEE 22nd Int. Conf. on Nanotechnology (NANO)* pp 401–4
[29] Wang Q, Wanjura C C and Marquardt F 2024 Training coupled phase oscillators as a neuromorphic platform using equilibrium propagation *Neuromorph. Comput. Eng.* **4** 034014
[30] Rageau T and Grollier J 2025 Training and synchronizing oscillator networks with equilibrium propagation *Neuromorph. Comput. Eng.* **4** 034008
[31] Rudner T, Porod W and Csaba G 2024 Design of oscillatory neural networks by machine learning *Front. Neurosci.* **18** 1307525
[32] Scellier B and Bengio Y 2017 Equilibrium propagation: bridging the gap between energy-based models and backpropagation *Front. Comput. Neurosci.* **11** 24
[33] Scellier B and Bengio Y 2017 Equivalence of equilibrium propagation and recurrent backpropagation *Neural Comput.* **31** 312–29
[34] Bonnin M 2017 Phase oscillator model for noisy oscillators *Eur. Phys. J. Spec. Top.* **226** 3227–37
[35] Gilli M, Bonnin M and Corinto F 2005 On global dynamic behavior of weakly connected oscillatory networks *Int. J. Bifurcation Chaos* **15** 1377–93
[36] Hoppensteadt F C and Izhikevich E M 1997 *Weakly Connected Oscillators* (Springer) pp 247–93
[37] Hoppensteadt F C and Izhikevich E M 1999 Oscillatory neurocomputers with dynamic connectivity *Phys. Rev. Lett.* **82** 2983–6
[38] Bick C, Böhle T and Kuehn C 2024 Higher-order network interactions through phase reduction for oscillators with phase-dependent amplitude *J. Nonlinear Sci.* **34** 77
[39] Hongu J and Iba D 2022 Nonlinear oscillator design based on phase reduction method for closed-loop system *Int. J. Adapt. Control Signal Process.* **36** 1958–97
[40] von der Gracht S, Nijholt E, and Rink B 2023 A parametrisation method for high-order phase reduction in coupled oscillator networks (arXiv:2306.03320 [math.DS])
[41] Mau E T K, Rosenblum M and Pikovsky A 2023 High-order phase reduction for coupled 2d oscillators *Chaos* **33** 101101
[42] Gengel E, Teichmann E, Rosenblum M and Pikovsky A 2020 High-order phase reduction for coupled oscillators *J. Phys. Complex.* **2** 015005
[43] Wang H, Yan H, Rong C, Yuan Y, Jiang F, Han Z, Sui H, Jin D and Li Y 2024 Multi-scale simulation of complex systems: a perspective of integrating knowledge and data *ACM Comput. Surv.* **56** 1–38
[44] Acebrón J A, Bonilla L L, Pérez Vicente C J, Ritort F and Spigler R 2005 The kuramoto model: a simple paradigm for synchronization phenomena *Rev. Mod. Phys.* **77** 137–85
[45] Bonnin M 2017 Amplitude and phase dynamics of noisy oscillators *Int. J. Circuit Theory Appl.* **45** 636–59
[46] Hale J K 1969 *Ordinary Differential Equations* (Wiley)
[47] Zoppo G, Marrone F, Min K-S and Corinto F 2022 Energy-based memristor networks for pattern recognition in vision systems *Memristor Computing Systems* (Springer) pp 53–63
[48] Hopfield J J 1982 Neural networks and physical systems with emergent collective computational abilities *Proc. Natl Acad. Sci.* **79** 2554–8
[49] Amit D J, Gutfreund H and Sompolinsky H 1985 Spin-glass models of neural networks *Phys. Rev.* A **32** 1007–18
[50] Amit D J and Amit D J 1989 *Modeling Brain Function: The World of Attractor Neural Networks* (Cambridge University Press)
[51] Robins A V and McCallum S J 2004 A robust method for distinguishing between learned and spurious attractors *Neural Netw.* **17** 313–26
[52] Oja E 1982 Simplified neuron model as a principal component analyzer *J. Math. Biol.* **15** 267–73
[53] Turrigiano G G, Leslie K R, Desai N S, Rutherford L C and Nelson S B 1998 Activity-dependent scaling of quantal amplitude in neocortical neurons *Nature* **391** 892–6
[54] Zenke F, Gerstner W and Ganguli S 2017 The temporal paradox of hebbian learning and homeostatic plasticity *Curr. Opin. Neurobiol.* **43** 166–76