Journal of Nonlinear Science (2024) 34:15
https://doi.org/10.1007/s00332-023-09994-y

Journal of Nonlinear Science

Phase-Amplitude Coordinate-Based Neural Networks for Inferring Oscillatory Dynamics

Talha Ahmed[1] · **Dan Wilson**[1]

Received: 3 March 2023 / Accepted: 16 October 2023

Abstract
The dynamics of a periodic nonlinear system can be represented accurately beyond the limit cycle in a reduced-order phase-amplitude coordinate-based model reduction framework. When only observable time series data is available, data-driven strategies must be employed for model inference. In this work, we propose a data-driven approach that can predict the unknown, periodic terms of a phase-amplitude coordinate-based reduced-order model by considering their Fourier series expansions and reframing the terms as a composition of a known nonlinear function with an unknown linear function. These linear functions can be structured as weights of a feed-forward neural network and learned to obtain a reduced-order model representation valid to arbitrary orders of accuracy in an expansion of amplitude coordinates by training the network on observable data. The proposed approach can be used in conjunction with other recently developed reduced-order modeling approaches to yield very high accuracy reduced-order models. The proposed strategy is illustrated in a variety of examples that consider the dynamics of a synaptically coupled neuronal population.

Keywords Model identification · Periodic orbit · Artificial neural networks · Phase-isostable coordinates · Dynamical systems · Model order reduction

Mathematics Subject Classification 34E10 · 37Cxx · 92Bxx · 68Txx

1 Introduction

Many naturally occurring phenomenon, especially complex dynamical systems, lack a mathematical model to explain their underlying dynamics. In most cases, one only has

Communicated by Christian Bick.

✉ Talha Ahmed
tahmed4@vols.utk.edu

[1] Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996, USA

Published online: 11 November 2023

access to a finite set of observables for the system; to extract useful information from the available data, data-driven strategies have been developed to infer low-dimensional, predictive dynamical models (Kutz et al. 2016; Brunton and Kutz 2019). These models attempt to approximate the full model dynamics of the underlying system in response to any general stimulus without prespecifying an underlying dynamical structure. Data-driven strategies are particularly effective in situations where a clear mapping between state variables and model observables is not available and where the underlying model mechanisms are not well understood.

To infer a set of equations of a reduced model from a given dataset, numerous data-driven techniques have been proposed. Eigensystem realization algorithm (ERA) (Juang and Pappa 1985) works by identifying a linear model for a given nonlinear system based on temporal data which is then transformed into modal space for parameter identification. Similarly, authors in Holmes et al. (1996), Towne et al. (2018), Berkooz et al. (1993) utilize proper orthogonal decomposition (POD) which is another data-driven model reduction technique that finds an orthogonal basis to fit the provided temporal dataset. A dynamical system can subsequently be obtained using Galerkin projection (Holmes et al. 1996; Noack et al. 2003). Approaches in Schmid (2010), Rowley et al. (2009), Kutz et al. (2016) employ dynamic mode decomposition (DMD) which computes a set of linear modes from time series data. Other techniques employ a prespecified function library from which relevant terms are sparsely selected in a way to effectively replicate the observable data without overfitting (Mangan et al. 2019; Brunton et al. 2016; Pantazis and Tsamardinos 2019; Schaeffer 2017). Alternatively, by incorporating adaptive parameter sets (Wilson 2021b, 2022a, b; Wilson and Djouadi 2020), reduced-order modeling methods can be devised that can capture the salient features of the nonlinear system. These adaptive parameters can efficiently capture the effect of large inputs, that perturb the state far from a nominal family of attractors that emerge as the adaptive parameter is changed. Various methods are also based on Koopman operator theory to identify reduced linear model representations. Koopman analysis can be used to represent nonlinear system dynamics with a linear representation by lifting its dynamics to an infinite-dimensional linear space (Budišić et al. 2012; Mezić 2013, 2019). In a data-driven setting, approaches (Geneva and Zabaras 2020; Kaiser et al. 2021; Lusch et al. 2018) have accomplished this by finding representative eigenmodes of the Koopman operator.

Variations of neural network-based approaches have also been used in previous works in order to infer dynamics of nonlinear systems. For example, Ortin et al. (2005) compares performance of feed-forward neural networks with modular neural networks for inferring chaotic time-delay system dynamics. Reference (Tan and Saif 2000) uses external recurrent neural networks (RNNs) for model identification of specific processes occurring in an automotive engine. To predict the solution of various time-dependent and autonomous systems, (Omidi et al. 2022) proposes the idea of orthogonal neural networks. The predicted systems generally follow singular Emden–Fowler dynamics, and hence, their behavior can generally be described by either an ordinary differential equation (ODE) or partial differential equation (PDE). Physics informed neural networks (PINNs) (Raissi et al. 2019) are a class of neural networks for model identification of systems involving nonlinear partial differential equations (PDEs). The approach incorporates PDE-based domain knowledge into essential com-

ponents of the neural network's training process, thus allowing for convergence to the optimal solution even with limited training input data. The PINN-based approaches can also be adjusted to solve both the forward and inverse problems for PDEs given a set of noisy measurements from a dynamical system. Feed-forward artificial neural networks have also been used in literature to perform model identification of nonlinear dynamics. Presented in Wray and Green (1994), the approach computes Volterra kernel representations for dynamical systems by assuming equivalency of Volterra series to feed-forward neural networks and utilizing the network's internal parameters. Work in Masri et al. (1992) uses the idea of dynamic neurons in feed-forward neural networks in order to capture system nonlinearities. For long-time predictive modeling of nonlinear dynamical systems, authors of Pan and Duraisamy (2018) modify feed-forward neural networks to augment Jacobian regularization in the network's loss function.

In this work, we consider data-driven strategies for reduced-order model identification of oscillatory dynamical systems. In these applications, phase-amplitude coordinates provide a robust modeling framework for identifying a reduced-order model that can effectively represent the full model dynamics of the underlying periodic system (Wilson 2020c; Ahmed and Wilson 2021). If the mathematical model is known, various approaches have been proposed that employ the direct method to compute the necessary phase and amplitude response curves (Wilson 2020c; Wilson and Ermentrout 2019). However, there are no general strategies available for systems with multiple amplitude coordinates and for systems that require accuracy beyond linear order. An alternative data-driven framework for phase-amplitude reduction was derived in Wilson (2020a) which employed a proper orthogonal decomposition (POD) reduction strategy to identify important features of the transient decay of solutions to the limit cycle. The extracted features can be explicitly related to phase-amplitude coordinates and ultimately define so-called data-driven phase-amplitude coordinates that are valid in the entire basin of attraction of a limit cycle. The efficacy of the approach is also illustrated in the work done in Ahmed and Wilson (2021) which also utilized these data-driven phase-amplitude reduced-order model representations to obtain optimal control inputs for mitigating the effects of jet lag. Related work in Wilson and Djouadi (2019) and Wilson (2021a) proposed a least-squares model fitting approach using the steady-state response to periodic forcing. However, the strategy only works with sinusoidal inputs limiting its utility for model fitting, requires a substantial amount of data and is sensitive to noise.

Recently, a strategy for reduced-order model identification was developed for dynamical systems with fixed points employing neural networks based on isostable coordinates (Mauroy et al. 2013; Wilson and Moehlis 2015), which represent level sets of the slowest decaying eigenmodes of the Koopman operator. This is done by reframing the isostable coordinate-based input–output dynamics in terms of the composition of a set of known nonlinear functions and unknown linear functions such that the unknown coefficients correspond to the neural network's weights and learning them through training the network. In this work, we extend this idea for nonlinear periodic systems by structuring neural networks in terms of phase and isostable coordinates to extract a reduced-order model representation. Unlike other data-driven approaches such as SINDy (Fasel et al. 2021) which utilize neural networks to identify terms of the reduced model's right-hand side, our approach uses the neural network to learn the

reduced model's unknown coefficients and provide a solution that can approximate the full model dynamics given an initial condition. The learned coefficients can also be utilized independently to obtain the corresponding isostable and phase response curves which can be used for further analysis of the full model dynamics. Our proposed approach restructures the neural network based on phase-amplitude dynamics in order to derive a reduced-order model. By contrast, the structure of PINNs is tailored to a given application. Moreover, our approach does not require any prior domain knowledge including any physical laws that govern the dynamical system in consideration. Also, our proposed approach builds the artificial neural network according to the derived reduced-order phase-amplitude coordinate dynamics rather than utilizing the network as a black-box like most vanilla neural network approaches do.

The organization of this paper is as follows: Sect. 2 provides necessary background on the phase and amplitude coordinates as well as previously developed phase-amplitude-coordinate-based model order reduction frameworks for oscillatory dynamical systems. Section 3 describes the mathematical formulation that allows for the implementation of the model identification strategy using artificial neural networks and discusses practical matters related to implementation. Results are given in Sect. 4 where we illustrate the proposed technique through a simple dynamical model along with a more complicated illustration in a model that captures neural spiking behavior before using adaptive phase-isostable reduction in conjunction with the proposed framework to emulate full model dynamics of a neuron population. Section 5 provides a discussion of the proposed strategy in the context of the results and gives concluding remarks.

2 Background

2.1 Phase and Isostable Coordinates

To begin, consider a general dynamical system of the form (1)

$$\begin{aligned} \dot{x} &= F(x, u(t)), \\ y &= F_{\text{out}}(x) \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^N$ is the system state, F represents the nominal dynamics, F_{out} maps the state to the single observable output $y \in \mathbb{R}$, and the system $\dot{x} = F(x, u(t))$ has a stable limit cycle $x^\gamma(p_0, t)$ that emerges when holding $u(t) \in \mathbb{R}$ constant at p_0. One can define a phase $\theta \in [0, 2\pi)$ valid for all locations on the limit cycle that results when $u(t)$ is held constant at the nominal value p_0. The phase is generally scaled so that $\mathrm{d}\theta/\mathrm{d}t = 2\pi/T = \omega$. Phase can be defined in the entire basin of attraction of the limit cycle using the notion of isochrons which are defined such that when $u(t) = p_0$, for any initial condition $a(0) \in x^\gamma(p_0, t)$, the isochron associated with $a(0)$ is defined to be the set of all $b(0)$ where $b(0)$ is in the basin of attraction of $x^\gamma(p_0, t)$ such that

$$\lim_{t \to \infty} ||a(t) - b(t)|| = 0, \tag{2}$$

where $||\cdot||$ can be any vector norm. Using the isochron-based definition of phase, we can encode for the infinite time behavior of solutions. To effectively capture the transient decay of solutions toward the periodic orbit, it is also useful to consider the amplitude dynamics. This is possible using Floquet theory (Jordan and Smith 2007). First define $\Delta x(t) = x(t) - x^\gamma(p_0, t)$ so that to a linear approximation, the dynamics of Eq. (1) are

$$\Delta \dot{x} = J\Delta x, \tag{3}$$

where J is the time-varying Jacobian of F evaluated at both $x^\gamma(p_0, t)$ and $u = p_0$. Letting Φ be the monodromy matrix defined such that $\Delta x(T) = \Phi \Delta x(0)$, consider the eigenvalues and associated left and right eigenvectors of Φ denoted by λ_j, w_j and v_j, respectively. Letting λ_1 be the nonunity eigenvalue (i.e., Floquet multiplier) of largest magnitude, it is possible to define a set of isostable coordinates valid in the basin of attraction of the limit cycle according to Wilson and Ermentrout (2018)

$$\psi_1(x) = \lim_{k\to\infty} \left[w_1^T (\nu(t_\Gamma^k, x) - x_0) \exp(-\kappa_1 t_\Gamma^k) \right], \tag{4}$$

where t_Γ^k denotes the time of the kth transversal of the $\theta = 0$ isochron, $\nu(t, x)$ gives the flow of Eq. (1) under the constant application of $u = p_0$, x_0 is the intersection of the periodic orbit and the $\theta = 0$ isochron, and $\kappa_1 = \log(\lambda_1)/T$ is the associated Floquet exponent. The explicit definition in Eq. (4) is only generally valid for the slowest decaying isostable coordinate, but additional isostable coordinates $\psi_2, \ldots, \psi_{N-1}$ can be defined implicitly by considering level sets of Koopman eigenfunctions associated with the nonunity Floquet multipliers of the linearized dynamics. A detailed discussion about the relationship of isostable coordinates to the Koopman operator can be found in Kvalheim and Revzen (2021), Mezić (2019).

2.2 Model Order Reduction Using Phase-Amplitude Coordinates

By assuming that both $u(t) - p_0$ and $x - x^\gamma(p_0, t)$ are order ϵ terms at all times where $0 < \epsilon \ll 1$ and $p_0 = 0$, one can asymptotically expand Eq. (1) about $x^\gamma(p_0, t)$ to yield

$$\dot{x} = F(x, p_0) + \frac{\partial F}{\partial u} U(t) + O(\epsilon^2), \tag{5}$$

where the partial derivative is evaluated at both $x^\gamma(p_0, t)$ and p_0 and $U(t) = u(t) - p_0$ is the input. Changing to phase and isostable coordinates via the chain rule as in Wilson and Moehlis (2016), Wilson and Ermentrout (2018) gives

$$\begin{aligned} \dot{\theta} &= \frac{\partial \theta}{\partial x}^T \left(F(x, p_0) + \frac{\partial F}{\partial u} U(t) \right) + O(\epsilon^2) \\ &= \omega + Z(\theta) U(t) + O(\epsilon^2), \\ \dot{\psi}_j &= \frac{\partial \psi_j}{\partial x}^T \left(F(x, p_0) + \frac{\partial F}{\partial u} U(t) \right) + O(\epsilon^2) \end{aligned} \tag{6}$$

$$
\begin{aligned}
&= \kappa_j \psi_j + I_j(\theta) U(t) + O(\epsilon^2), \\
&j = 1, \ldots, \beta,
\end{aligned} \tag{7}
$$

where $Z(\theta) = \frac{\partial \theta}{\partial x} \cdot \frac{\partial F}{\partial u}$ and $I_j(\theta) = \frac{\partial \psi_j}{\partial x} \cdot \frac{\partial F}{\partial u}$ are the phase and isostable response curves. It is often possible to ignore isostable coordinates ψ_j for which the corresponding Floquet exponents κ_j are large in magnitude so that they decay rapidly (Monga et al. 2019; Wilson and Ermentrout 2019; Wilson 2020b); in this work, only β isostable coordinates are considered so that the resulting model has a lower dimensionality than the full model.

Although standard phase reduction has been used extensively in literature for model reduction of systems with periodic orbits, it has its own set of drawbacks. The main limitation of the phase-based framework is its inability to account for large inputs, which makes it unusable in many real applications. As a workaround to this limitation, one can consider amplitude coordinates in conjunction with the phase dynamics. Here, a slight modification of the phase-amplitude coordinates from Wilson (2020c) is used to yield the reduced-order model of the form

$$
\begin{aligned}
\dot{\theta} &= \omega + Z(\theta, \psi_1, \ldots, \psi_\beta) U(t), \\
\dot{\psi}_j &= \kappa_j \psi_j + I_j(\theta, \psi_1, \ldots, \psi_\beta) U(t), \\
&j = 1, \ldots, \beta, \\
y(p_0, \theta, \psi_1, \ldots, \psi_\beta) &= y(x^\gamma(p_0, \theta)) + G(\theta, \psi_1, \ldots, \psi_\beta),
\end{aligned} \tag{8}
$$

where $Z(\theta, \psi_1, \ldots, \psi_\beta) \in \mathbb{R}$ and $I_j(\theta, \psi_1, \ldots, \psi_\beta) \in \mathbb{C}$ provides an approximation for the phase and isostable response curves for arbitrary orders of accuracy in an expansion of isostable coordinates centered at the periodic orbit $x^\gamma(p_0, \theta)$. $G(\theta, \psi_1, \ldots, \psi_\beta) \in \mathbb{R}^N$ provides a good approximation for $y(p_0, \theta, \psi_1, \ldots, \psi_\beta) - y(x^\gamma(p_0, \theta))$. Considering the Taylor expansion of Z, I_k and G in a basis of the nontruncated isostable coordinates near the periodic orbit, one finds

$$
Z(\theta, \psi_1, \ldots, \psi_\beta) \approx Z(\theta) + \sum_{k=1}^{\beta} \left[\psi_k Z^k(\theta)\right] + \sum_{j=1}^{\beta} \sum_{k=1}^{j} \left[\psi_j \psi_k Z^{jk}(\theta)\right] + \sum_{i=1}^{\beta} \sum_{j=1}^{i} \sum_{k=1}^{j} \left[\psi_i \psi_j \psi_k Z^{ijk}(\theta)\right] + \ldots, \tag{9}
$$

$$
I_n(\theta, \psi_1, \ldots, \psi_\beta) \approx I_n(\theta) + \sum_{k=1}^{\beta} \left[\psi_k I_n^k(\theta)\right] + \sum_{j=1}^{\beta} \sum_{k=1}^{j} \left[\psi_j \psi_k I_n^{jk}(\theta)\right] + \sum_{i=1}^{\beta} \sum_{j=1}^{i} \sum_{k=1}^{j} \left[\psi_i \psi_j \psi_k I_n^{ijk}(\theta)\right] + \ldots, \tag{10}
$$

$$
G(\theta, \psi_1, \ldots, \psi_\beta) \approx \sum_{k=1}^{\beta} \left[\psi_k g^k(\theta)\right] + \sum_{j=1}^{\beta} \sum_{k=1}^{j} \left[\psi_j \psi_k g^{jk}(\theta)\right] + \sum_{i=1}^{\beta} \sum_{j=1}^{i} \sum_{k=1}^{j} \left[\psi_i \psi_j \psi_k g^{ijk}(\theta)\right] + \ldots, \tag{11}
$$

for $n = 1, \ldots, \beta$. Note that in the reduced-order model (8) given above, it is assumed that both $U(t)$ and each isostable coordinate $\psi_1, \ldots, \psi_\beta$ are order $O(\epsilon)$ terms. Therefore, $I_n(\theta)$, $I_n^k(\theta)$, $I_n^{jk}(\theta)$ as well as $Z(\theta)$, $Z^k(\theta)$, $Z^{jk}(\theta)$ from the expansion (10) and (9) are commonly referred to as first-, second-, third-order terms, respectively, of the reduced-order model. Likewise, from Eq. (11), $g^k(\theta)$, $g^{jk}(\theta)$ and $g^{ijk}(\theta)$ will be referred to as the first-, second-, and third-order terms, respectively, with this convention also being followed for higher orders.

3 Problem Formulation and General Approach

3.1 Problem Description

It is possible to directly solve for each of the $g^{ijk\cdots}(\theta)$, $Z^{ijk\cdots}(\theta)$ and $I_n^{ijk\cdots}(\theta)$ terms from Eqs. (9), (10), (11) if the underlying model equations from (1) are known as described in Wilson (2020c). In case the model dynamics are unknown, the necessary terms must be identified using data-driven methods. Our proposed framework trains an artificial neural network through gradient descent by considering a collection of output measurements $y_1(t), y_2(t), \ldots, y_n(t)$ that result from the application of an arbitrary collection of applied inputs $U_1(t), U_2(t), \ldots, U_n(t)$ for $t \in [t_0, t_1]$ on a general system of the form of Eq. (1). Using this strategy, we infer the unknown terms from Eqs. (9), (10) and (11) to arbitrary orders of accuracy in the expansion of phase and isostable coordinates. We emphasize that Sect. 3 assumes that the value of p_0 is fixed, i.e., that the resulting terms of the reduction in Eqs. (9), (10), (11) are associated with a single periodic orbit $x^\gamma(p_0, t)$. The resulting model inference strategy can be repeated for different values of p_0 if necessary.

3.2 Reframing the Dynamics for the Neural Network

Using the phase-amplitude coordinate-based reduction of the form (8), we reframe the problem so that neural networks can provide an appropriate solution. Each of the terms $Z^{ijk\cdots}(\theta)$, $g^{ijk\cdots}(\theta)$ and $I_n^{ijk\cdots}(\theta)$ from Eqs. (9), (10), (11) will be written as a series of Fourier coefficients, i.e.,

$$X(\theta) \approx a_{X0} + \sum_{j=1}^{m} [a_{Xj} \sin(j\theta) + b_{Xj} \cos(j\theta)], \tag{12}$$

where $X \in [Z^{ijk\cdots}, I_n^{ijk\cdots}, g^{ijk\cdots}]$ and m is the number of Fourier coefficients considered. By learning the coefficients of these terms to a desired order of accuracy in both the isostable coordinate and the Fourier series expansion, a reduced order model can be obtained that accurately replicates nonlinear behaviors of the underlying system (1). Letting $\Psi = [\psi_1, \ldots, \psi_\beta]^T$ and considering the phase and isostable dynamics, we can use a forward Euler method of solution with a timestep of Δt to represent the phase and isostable coordinates as well as the corresponding outputs at $t + \Delta t$ as

$$\begin{aligned}
\theta(t+\Delta t) &= f(\theta(t), \Psi(t), U(t)) \\
&= \theta(t) + \omega\Delta t + Z(\theta(t))U(t)\Delta t + \sum_{j=1}^{k} [\psi_k(t) Z_k^1(\theta(t))] U(t)\Delta t + \ldots, \\
&= \theta(t) + \omega\Delta t + \left[a_{Z0} + \sum_{j=1}^{m} [a_{Zj} \sin(j\theta(t)) + b_{Zj} \cos(j\theta(t))] \right] U(t)\Delta t
\end{aligned}$$

$$+\sum_{k=1}^{\beta}\left[a_{Z_k^1 0}+\sum_{j=1}^{m}[a_{Z_k^1 j}\sin(j\theta(t))+b_{Z_k^1 j}\cos(j\theta(t))]\right]\psi_k(t)U(t)\Delta t+\ldots,$$

$$\begin{aligned}
\psi_k(t+\Delta t) &= h_k(\theta(t),\Psi(t),U(t))\\
&= (1+\kappa_k\Delta t)\psi_k(t)+I_k(\theta(t))U(t)\Delta t+\sum_{j=1}^{k}[\psi_k(t)I_k^1(\theta(t))]U(t)\Delta t+\ldots,\\
&= (1+\kappa_k\Delta t)\psi_k(t)+\left[a_{I_k 0}+\sum_{j=1}^{m}[a_{I_k j}\sin(j\theta(t))+b_{I_k j}\cos(j\theta(t))]\right]U(t)\Delta t\\
&\quad+\sum_{k=1}^{\beta}\left[a_{I_k^1 0}+\sum_{j=1}^{m}[a_{I_k^1 j}\sin(j\theta(t))+b_{I_k^1 j}\cos(j\theta(t))]\right]\psi_k(t)U(t)\Delta t+\ldots,\\
k &= 1,\ldots,\beta,\\
y(t)-y(x^\gamma(p_0,\theta(t))) &= \hat{g}(\theta(t),\Psi(t)),\\
&= \sum_{k=1}^{\beta}\left[a_{g^k 0}+\sum_{j=1}^{m}[a_{g^k j}\sin(j\theta(t))+b_{g^k j}\cos(j\theta(t))]\right]\psi_k(t)\\
&\quad+\sum_{l=1}^{\beta}\sum_{k=1}^{l}\left[a_{g^{lk}0}+\sum_{j=1}^{m}[a_{g^{lk}j}\sin(j\theta(t))+b_{g^{lk}j}\cos(j\theta(t))]\right]\psi_l(t)\psi_k(t)+\ldots.
\end{aligned}\tag{13}$$

Above, the functions $h_1,\ldots,h_\beta$ and f consider the current phase and isostable coordinates as well as the perturbation input and map to the phase and isostable coordinates Δt time units later. Each h_k with associated IRC I_k and f associated with PRC Z, defined according to the Taylor expansion in (9) and (10), respectively, is linear in a basis of lifted coordinates consisting of both the state and input. For instance, when using only one isostable coordinate (i.e., when $\beta=1$), one can write

$$\begin{aligned}
\psi_1(t+\Delta t) &= (1+\kappa_1\Delta t)\psi_1(t)+I_1(\theta(t))U(t)\Delta t+I_1^1(\theta(t))\psi_1(t)U(t)\Delta t\ldots,\\
&= (1+\kappa_1\Delta t)\psi_1(t)+\left[a_{I_1 0}+\sum_{j=1}^{m}[a_{I_1 j}\sin(j\theta(t))+b_{I_1 j}\cos(j\theta(t))]\right]U(t)\Delta t\\
&\quad+\left[a_{I_1^1 0}+\sum_{j=1}^{m}[a_{I_1^1 j}\sin(j\theta(t))+b_{I_1^1 j}\cos(j\theta(t))]\right]\psi_1(t)U(t)\Delta t+\ldots,
\end{aligned}\tag{14}$$

Keeping this in consideration, each h_k can be taken as the composition of two functions

$$h_k = w_k\circ n_I,\tag{15}$$

for $k=1,\ldots,\beta$, where $n_I:\mathbb{S}\times\mathbb{C}^\beta\times\mathbb{R}\to\mathbb{C}^\varrho$, $w_k:\mathbb{C}^\varrho\to\mathbb{C}$, and ϱ is the size of the lifted basis. For example, when $\beta=1$ and taking the asymptotic expansion to third order of accuracy in the isostable coordinate with first-order Fourier series expansion,

$$n_I(\theta, \psi_1, U) = \begin{bmatrix} \psi_1 \\ U \\ \sin(\theta)U \\ \cos(\theta)U \\ \psi_1 U \\ \sin(\theta)\psi_1 U \\ \cos(\theta)\psi_1 U \\ \psi_1^2 U \\ \sin(\theta)\psi_1^2 U \\ \cos(\theta)\psi_1^2 U \end{bmatrix}, \quad w_1 = n_I(\theta, \psi_1, U)^T \begin{bmatrix} 1 + \kappa_1 \Delta t \\ a_{I_1 0}\Delta t \\ a_{I_1 1}\Delta t \\ b_{I_1 1}\Delta t \\ a_{I_1^1 0}\Delta t \\ a_{I_1^1 1}\Delta t \\ b_{I_1^1 1}\Delta t \\ a_{I_1^{11} 0}\Delta t \\ a_{I_1^{11} 1}\Delta t \\ b_{I_1^{11} 1}\Delta t \end{bmatrix}. \tag{16}$$

The terms $\psi_1^3 U$ and $I_1^{111}(\theta)$ and higher terms are truncated at third order of accuracy as U and ψ_1 are both $O(\epsilon)$. The coefficients that comprise each w_k need to be learned alongside the corresponding coefficients for both Z and G in order to obtain the reduced-order model representations. Similarly, f can also be written as the composition of two functions as

$$f = w_Z \circ n_Z, \tag{17}$$

where $n_Z : \mathbb{S} \times \mathbb{C}^\beta \times \mathbb{R} \to \mathbb{C}^{\varrho+1}$, $w_Z : \mathbb{C}^{\varrho+1} \to \mathbb{R}$. Here, $\varrho + 1$ is the size of the lifted basis. As done before, taking the asymptotic expansion to third order of accuracy for one isostable coordinate with first-order Fourier series expansion,

$$n_Z(\theta, \psi_1, U) = \begin{bmatrix} \theta \\ \omega \\ U \\ \sin(\theta)U \\ \cos(\theta)U \\ \psi_1 U \\ \sin(\theta)\psi_1 U \\ \cos(\theta)\psi_1 U \\ \psi_1^2 U \\ \sin(\theta)\psi_1^2 U \\ \cos(\theta)\psi_1^2 U \end{bmatrix}, \quad w_Z = n_Z(\theta, \psi_1, U)^T \begin{bmatrix} 1 \\ \Delta t \\ a_{Z0}\Delta t \\ a_{Z1}\Delta t \\ b_{Z1}\Delta t \\ a_{Z_1^1 0}\Delta t \\ a_{Z_1^1 1}\Delta t \\ b_{Z_1^1 1}\Delta t \\ a_{Z_1^{11} 0}\Delta t \\ a_{Z_1^{11} 1}\Delta t \\ b_{Z_1^{11} 1}\Delta t \end{bmatrix}. \tag{18}$$

Finally, we can consider $\hat{g}$ as the composition of two additional functions as well

$$\hat{g} = w_G \circ n_G, \tag{19}$$

where $n_G : \mathbb{S} \times \mathbb{C}^\beta \to \mathbb{C}^{\varrho-1}$, $w_G : \mathbb{C}^{\varrho-1} \to \mathbb{R}$. Here, $\varrho - 1$ is the size of the lifted basis. Taking the asymptotic expansion to third order of accuracy with one isostable

coordinate with first-order Fourier series expansion,

$$n_G(\theta, \psi_1) = \begin{bmatrix} \psi_1 \\ \sin(\theta)\psi_1 \\ \cos(\theta)\psi_1 \\ \psi_1^2 \\ \sin(\theta)\psi_1^2 \\ \cos(\theta)\psi_1^2 \\ \psi_1^3 \\ \sin(\theta)\psi_1^3 \\ \cos(\theta)\psi_1^3 \end{bmatrix}, \quad w_G = n_G(\theta, \psi_1)^T \begin{bmatrix} a_{g^1 0} \\ a_{g^1 1} \\ b_{g^1 1} \\ a_{g^{11} 0} \\ a_{g^{11} 1} \\ b_{g^{11} 1} \\ a_{g^{111} 0} \\ a_{g^{111} 1} \\ b_{g^{111} 1} \end{bmatrix}. \tag{20}$$

For a general system of the form (1), the nonlinear functions n_Z, n_I and n_G are known once the order of accuracy along with the number of isostable coordinates is specified. On the other hand, the functions $w_Z, w_1, \ldots, w_\beta$ and w_G are linear and contain the unknown coefficients that need to be learned and are comprised of the Fourier series coefficients of the terms $Z^{ijk\cdots}(\theta)$, $I_n^{ijk\cdots}(\theta)$ and $g^{ijk\cdots}(\theta)$ from Eqs. (9), (10) and (11), respectively. Hence, the phase and isostable coordinate update rules from (13) are specified through the identification of these unknown Fourier series coefficients.

3.3 A Data-Driven Approach for Model Identification of Oscillatory Dynamics Using Neural Networks

Ultimately, the goal is to infer the Fourier series coefficients comprising the unknown terms from Eqs. (15), (17) and (19) as described in Sect. 3.2, along with the decay rates $\kappa_1, \ldots, \kappa_\beta$ using discrete sets of observable measurements $[y_j(t_0), y_j(t_0 + \Delta t), \ldots, y_j(t_0 + \eta\Delta t)]$ that result when applying inputs $[U_j(t_0), U_j(t_0 + \Delta t), \ldots, U_j(t_0 + (\eta - 1)\Delta t)]$ to the system for $j = 1, \ldots, \nu$, yielding a reduced order model of the form in Eq. (8). Here, ν is the number of distinct input profiles, $U(t)$ and each of these input profiles has η measurements. Also, it is assumed that at time t_0, the system state is on the limit cycle with initial phase and isostable coordinates $\theta(t_0) = \psi_1(t_0) = \psi_2(t_0) = \cdots = \psi_\beta(t_0) = 0$.

The implementation for the reduced-order phase-amplitude coordinate-based representation is done using a multi-layer simple feed-forward network architecture structured for model learning purposes. A representation of the network structure is shown in Fig. 1 which is based on the mathematical formulation from Sect. 3.2. First, the neural network's input layer accepts a concatenation of the phase coordinate $\theta(t_0)$, isostable coordinates $\Psi(t_0)$ and the applied input $U(t_0)$ which is then fed to a nontrainable function layer for both phase and isostable coordinate. This layer lifts the input layer variables to a higher dimension implementing the (known) functions n_Z and n_I. The lifted states are passed through two respective hidden layers that implement the linear functions w_Z and $w_1, \ldots, w_\beta$ to yield both the updated isostable coordinate $\Psi(t_0 + \Delta t)$ and the updated phase $\theta(t_0 + \Delta t)$. This intermittent state, comprised of the phase and isostable coordinate update, is then fed through another nontrainable function layer which implements the lifting described by n_G. This second lifted state is

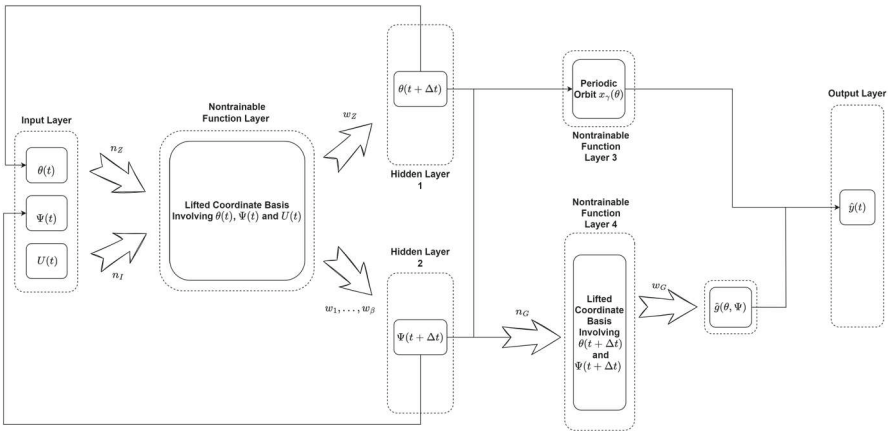

Fig. 1 General architecture for the proposed neural network-based strategy

fed through a fully connected trainable output layer that implements w_G and results in an intermediate output denoted by $\hat{g}(\theta, \Psi)$. Another intermediate output is generated by a nontrainable function layer containing the parametric function of the system's periodic orbit; this layer takes the phase coordinate update as the input. Summing up both these intermediate inputs ultimately generates the final output.

The obtained observable estimate at a given time is denoted as $\hat{y}_j(t)$ and compared with a system measurement $y_j(t)$ that results from the application of the input $U_j(t)$. Then, we use the current approximation for the update function, for a given set of input–output measurements y_j and U_j, from Eq. (13) to yield a set of approximations $\hat{y}_D = [\hat{y}_j(t_0), \hat{y}_j(t_0 + \Delta t), \ldots, \hat{y}_j(t_0 + \eta\Delta t)]$. For computing the error between the estimate and the ground function, the mean squared error (MSE) is taken as the loss function

$$\text{MSE} = \frac{1}{\nu\eta} \sum_{j=1}^{\nu} \sum_{i=1}^{\eta} (y_j(t_0 + i\,\Delta t) - \hat{y}_j(t_0 + i\,\Delta t))^2. \tag{21}$$

Note that ν represents the number of inputs while η denotes the number of timesteps for each input. During backpropagation, the weights of the trainable layers, i.e., w_Z, $w_1, \ldots, w_\beta$ and w_G are updated. In the examples presented in this work, we assume that isostable coordinates are real and the weights are updated accordingly during backpropagation. To ensure that the network learns a solution that is generalizable for a variety of input profiles, the neural network is trained using mini-batch gradient descent which allows us to train on multiple inputs simultaneously.

As we are learning weights of a linear function, we use a linear activation function in the artificial neural network. A nonlinear reduced-order model of the form (9)–(11) can be obtained based upon the learned weights of the network once training is done that is independent of the artificial neural network. Note that in Eq. (13), the Fourier series expansion of both $Z(\theta)$ and $I_n(\theta)$ for $n = 1, \ldots, \beta$ is multiplied by Δt; as such, the learned Fourier coefficients for the terms $Z(\theta)$, $Z^k(\theta), \ldots$ and

$I_n(\theta), I_n^k(\theta), \ldots$ will be proportional to Δt. Therefore, appropriate scaling of the learned coefficients is required during conversion between the discrete time update rule (13) and the continuous reduced-order model (9)–(11).

Figure 1 illustrates a layer level diagram representing the neural network's general architecture employed in this work. Starting from the input layer, it is comprised of the phase and isostable coordinates value at the current timestep, i.e., $\theta(t)$, $\Psi(t)$, as well as the input, $U(t)$. These input layer components are then passed through the functions, n_I and n_Z that transforms these components to the lifted coordinate basis for both isostable and phase coordinates, respectively, denoted by the nontrainable function layer. Following that, both the isostable and phase coordinate update rules from Eq. (13) are implemented by multiplying the lifted coordinate basis with the coefficients comprising $w_1, \ldots, w_\beta$ and w_Z; these coefficients are the weights associated with the respective hidden layers in the neural network and are learned through training. The updated phase-amplitude coordinates, $\theta(t+\Delta t)$ and $\Psi(t+\Delta t)$, are then computed through the two hidden layers. The updated values are recursively fed back into the input layer of the network to generate further updates at successive timesteps. Once all phase and isostable coordinate updates are obtained, these are passed through n_G, to transform $\Psi(t+\Delta t)$ and $\theta(t+\Delta t)$ to a second set of lifted coordinates obtained through another nontrainable function layer. A transitional output, shown as $\hat{g}(\theta, \Psi)$, is generated though mapping from the phase and isostable coordinates to the state from Eq. (13) where w_G contains the associated weights. Another intermediate output is generated by a nontrainable function layer containing the periodic orbit function. Finally, the output layer is comprised of the summation of both of these intermediary outputs. The predicted outputs $[\hat{y}(t_0), \hat{y}(t_0+\Delta t), \hat{y}(t_0+2\Delta t), \ldots]$ are identified recursively and compared with the full model output $[y(t_0), y(t_0+\Delta t), y(t_0+2\Delta t), \ldots]$ to compute the prediction error needed for training through backpropagation. The next section details the specific steps required for the training process and the network implementation.

3.4 Implementation of the Model Identification Strategy Using a Neural Network

Implementation for our neural network-based approach for model identification described in Sect. 3.3 is detailed here; the network structure is feed-forward with no recurrent connections. Keras is used for building the neural network on top of Tensorflow, a machine learning platform within python; Keras is a deep learning application programming interface (API) written in python. Furthermore, the training data, i.e., the phase and isostable coordinates, needs to be generated simultaneously as the learning proceeds. This is different from conventional supervised learning methods which have the training data readily available. The process for the model identification strategy is outlined below. At the start of this procedure, it is assumed that a dataset of outputs $y_1(t), \ldots, y_n(t)$ generated by inputs $U_1(t), \ldots, U_n(t)$ has been measured from the full model. Also, the initial phase and isostable coordinates are taken to be zero and thus, each of the measured outputs are taken at the periodic orbit $x^\gamma(p_0, \theta)$ at $\theta = 0$. Finally, for the system under consideration, the representation for its periodic orbit as well as its natural frequency ω is obtained prior to training the neural network.

Step (1) Initially, specify the number of isostable coordinates β. Measure the periodic orbit for the full model and the natural frequency ω. Also, set the order of the Fourier series expansion for each of the terms of Z, I_k, and G denoted by m. To avoid overfitting, one should generally use the fewest isostable coordinates possible. Define functions for the nontrainable function layers that implement the lifting functions n_I, n_Z and n_G from (15), (17) and (19). Also, define the layer containing the representation of the periodic orbit.

Step (2) Define the artificial neural network's structure based on the order of accuracy (in the expansion in isostable coordinates) and the Fourier series expansion of the functions f, $h_1, \ldots, h_\beta$ and $\hat{g}$ from Eq. (13). To ensure that the trained weights directly correspond to the Fourier series coefficients from Eqs. (16), (18) and (20), linear activation functions are used in the network layers.

Step (3) Define an auxiliary function that uses the current network weights to compute the forward Euler step from Eq. (13). Output predictions $\hat{y}_D(t)$ for training the network are also generated by this function.

Step (4) Initialize network weights, specify an optimizer, a loss function and a learning rate. Mean square error loss from (21) is used for the approach.

Step (5) Every epoch consists of two loops. The first loop calculates the phase and isostable coordinate update for a given input $U(t)$ by implementing the forward Euler update rule from Eq. (13). The data generated by the first loop is fed into the second loop which updates the weights of the trainable layers. It is assumed that the system starts at the limit cycle, i.e., $\theta(t_0) = 0$, $\psi_k(t_0) = 0$ for $k = 1, \ldots, \beta$; all subsequent coordinate updates are generated while training without the need of direct measurements. Finally, an outer loop iterates over the epochs until the training loss converges.

Step (6) The trained neural network weights correspond to Fourier series coefficients of the Taylor expanded terms from Eqs. (9), (10) and (11) along with decay rates and the natural frequency. The learned weights are independent of the training network and can be extracted to yield a reduced-order model of the form (8).

Note that the proposed approach learns the Fourier series coefficients for each of the terms of Z, I and G from Eqs. (9), (10) and (11) to provide a good characterization of the full model dynamics but will not necessarily match with the true Z, I and G functions obtained, for instance, using the adjoint method (Monga et al. 2019) or other direct numerical strategies. For the training procedure detailed above, a few general notes about its implementation are given. The weight initialization in Step 4 can sometimes cause the proposed model identification strategy to perform poorly. As such, it is generally useful to initially train the network using small magnitude inputs, finding the Fourier series coefficients for the first-order terms $Z(\theta)$, $I_n(\theta)$ and $g^n(\theta)$, along with the Floquet exponents κ_n for $n = 1, \ldots, \beta$. For higher-order models, one can use larger magnitude inputs to obtain data for training by using the learned first-order coefficients for weight initialization from Step 4 in the procedure above. To ensure that these baseline first-order coefficients do not deviate substantially from their linear order values that were already fit to the lower-order model, weight regularization can be used during learning for the higher-order weights and added as a penalty term to the loss function. For first-order accuracy, the isostable response curves are unique

to constant scaling. As such, the coefficients must be constrained. A regularization scheme is incorporated within the strategy that forces the sum of squares of the learned weights to be equal to 1 by penalizing the network training if their sum is far from 1. Furthermore, to yield a reduced model that can generalize to novel inputs, the training dataset consists of multiple batches corresponding to different inputs. A single batch is comprised of the total number of timesteps for a specific input–output pair, and thus, through the mini-batch stochastic gradient descent algorithm, training on all network inputs in each epoch is important for the network. As indicated previously, periodically updating the training dataset is crucial as the inputs for subsequent timesteps depend on the network itself; it is done after each epoch in the presented approach using the current weight updates. As the network inputs themselves are generated from incorrect updates, there is no need to reach convergence before updating weights. However, for a more generalizable solution, training is allowed to occur on all input–output pairs by waiting for an entire epoch.

4 Results

4.1 Illustration in a Model With Dynamics Near a Hopf Bifurcation

We consider the modified version of the radial isochron clock (Winfree 2001) illustrating the proposed phase-amplitude model identification strategy

$$\begin{aligned} \dot{x}_1 &= C\sigma x_1(\mu - x_1^2 - x_2^2) - x_2(1 + \rho(x_1^2 + x_2^2 - \mu)) + u(t), \\ \dot{x}_2 &= C\sigma x_2(\mu - x_1^2 - x_2^2) + x_1(1 + \rho(x_1^2 + x_2^2 - \mu)), \\ y(t) &= x_1(t), \end{aligned} \tag{22}$$

where $u(t)$ is an external input applied directly to the x_1 variable and $C = \frac{2\pi}{T}$ with $T = 24.2$. Above, we choose the parameters $\sigma = 0.04$, $\rho = 0.12$ and $\mu = 1$. Here, μ is a bifurcation parameter; when $\mu = 0$, a Hopf bifurcation occurs resulting in a stable limit cycle for $\mu > 0$. With these parameters, the model has a limit cycle at $x_1^2 + x_2^2 = 1$ when taking $u(t) = 0$.

The neural network model is trained on data from output measurements considering a single isostable coordinate for the reduced model; Fig. 1 shows the general architecture for the associated neural network. For optimizing and learning the weights, adaptive moment estimation (ADAM) optimizer is used with an MSE loss of the form (21) and a learning rate of 0.001. For the update rule from (13), the time step is taken to be $\Delta t = 0.05$. For training, a summation of step inputs of the form

$$u(t) = \sum_{i=0}^{5} \epsilon_{i+1}(\xi(t - t_i) - \xi(t - t_{i+1})), \tag{23}$$

is used where $\xi(t)$ is the Heaviside step function. Meanwhile, $\epsilon_1, \ldots, \epsilon_6$ sets the magnitude of the respective components of the input, and finally, $t_0, \ldots, t_6$ controls

the duration of the input application. It should be noted that each of the successive duration is greater than the previous one by a certain amount, i.e., $t_6 > \cdots > t_0$. For this example, one hundred training inputs are generated using $\epsilon_n = a + (b - a)\text{rand}(0, 1)$ with $n = 1, \ldots, 6$ by randomly taking values between $a = -0.030$ and $b = 0.030$ drawn from a uniform distribution for first-order accuracy. Meanwhile, the input durations are assigned using $t_n = a_1 + (b_1 - a_1)\text{rand}(0, 1) + t_{n-1}$ for $n = 1, \ldots, 6$ with $t_0 = 0$ by drawing from a random distribution between 0 and 1 and choosing randomly between $a_1 = 10$ and $b_1 = 20$ with each successive time value greater than the previous one.

Considering expansion of isostable coordinates to first order of accuracy around the limit cycle, a total of 36 coefficients are learned with a single isostable coordinate and fifth order of Fourier series expansion for each of the terms of Z, I and G. The natural frequency for the model is specified to be 0.2596 and the functional representation for the periodic orbit is given by $y(t) = \sin(\theta)$; this is done before training the model as specified in Sect. 3.4. The learned first-order coefficients are then used for initialization of weights and fixed as a baseline through regularization for subsequent training of higher-order models up to third order of accuracy. The learning rate is kept at the same value of 0.001 for higher-order models.

The second-order model is trained by generating a set of inputs with same range of magnitude and input duration as done for the first order. For the third-order model, another set of one hundred training inputs of the form (23) is generated at random such that $\epsilon_n = a + (b - a)\text{rand}(0, 1)$ with $n = 1, \ldots, 6$ by taking values between $a = -0.032$ and $b = 0.032$ and input duration, i.e., $t_n = a_1 + (b_1 - a_1)\text{rand}(0, 1) + t_{n-1}$ for $n = 1, \ldots, 6$ with $t_0 = 0$ generated within the values of $a_1 = 10$ and $b_1 = 20$. The set with larger magnitude inputs is used here to drive the state far from the limit cycle so that the contribution from nonlinear terms can be captured efficiently. In terms of training time, the first-, second- and third-order reduced models take 92, 147 and 225 s, respectively, to achieve convergence using a desktop computer with a midgrade processor. The subsequent increase in learning time for the higher order accuracy models can be attributed to the increasing number of coefficients at higher orders of accuracy, i.e., that comprise the Taylor expansions from Eqs. (9), (10) and (11).

For validation, three test inputs of the form $u(t) = \alpha(1 - \xi(t - t_s))$ are used with $\alpha \in \{0.020, 0.030, 0.040\}$ and $t_s = 24.2$. In Fig. 2, Panels A, B and C show outputs in response to these test inputs (shown in Panel G, H and I). Panels D, E and F show the associated magnitudes of the error. For all the step inputs shown, the third-order model has substantially smaller error than the first and second-order accurate models. One can see that for the test input results shown in Panel A, the predicted outputs for all orders of accuracy are nearly identical to the full model output. The difference between the first to third-order accuracy models and the full model output is more apparent in panel B; error for first-order accuracy is the highest followed by second order with third-order accuracy error being the lowest. For higher magnitude inputs in panels H and I, the error plots in panel E and F follow the same trend even though the error magnitudes increase. This increase in error is also evident in the outputs shown in panels B and C when compared to panel A.

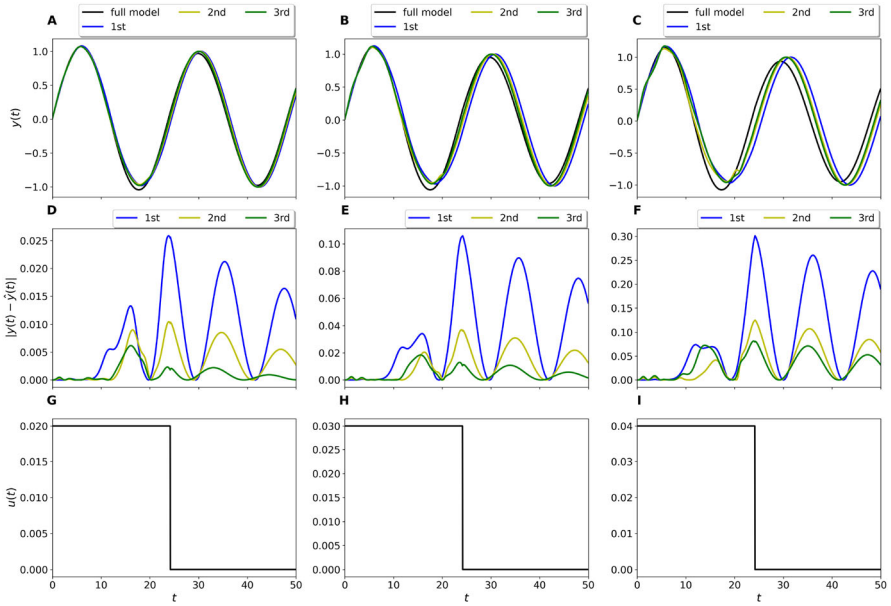

Fig. 2 The proposed model identification algorithm is applied to Eq. (22). For this illustration, a single isostable coordinate is considered and models of the form (8) are obtained that are valid to first through third orders of accuracy in the expansion in the isostable coordinates. Three inputs of varying magnitudes shown in panel (**G**)-(**I**) are applied and the obtained outputs from full and phase-amplitude-coordinate-based models are compared in panels (**A**)–(**F**)

For additional validation of the learned models, a sinusoidal test input of the form $u(t) = 0.03\sin(0.1t)$ is used. The obtained results are shown in Fig. 3. Panels A and Panel B follow the same pattern as observed in Fig. 2.

4.2 Spike Rates of Neural Populations

As a second example, a more complicated system is considered for our proposed model identification strategy. The system consists of a large, coupled population of neurons that captures spiking rates in response to external inputs. Dynamical equations are based on a model for thalamic neurons from Rubin and Terman (2004):

$$\begin{aligned} C_m \dot{V_k} &= -I_{\mathrm{L}}(V_k) - I_{\mathrm{Na}}(V_k, h_k) - I_{\mathrm{K}}(V_k, h_k) - I_{\mathrm{T}}(V_k, r_k) \\ &\quad + I_k^b - 0.2\sum_{j=1}^{N}(V_k - V_j) + i_k(t), \\ \dot{h_k} &= (h_\infty(V_k) - h_k)/\tau_h(V_k), \\ \dot{r_k} &= (r_\infty(V_k) - r_k)/\tau_r(V_k), \end{aligned}$$

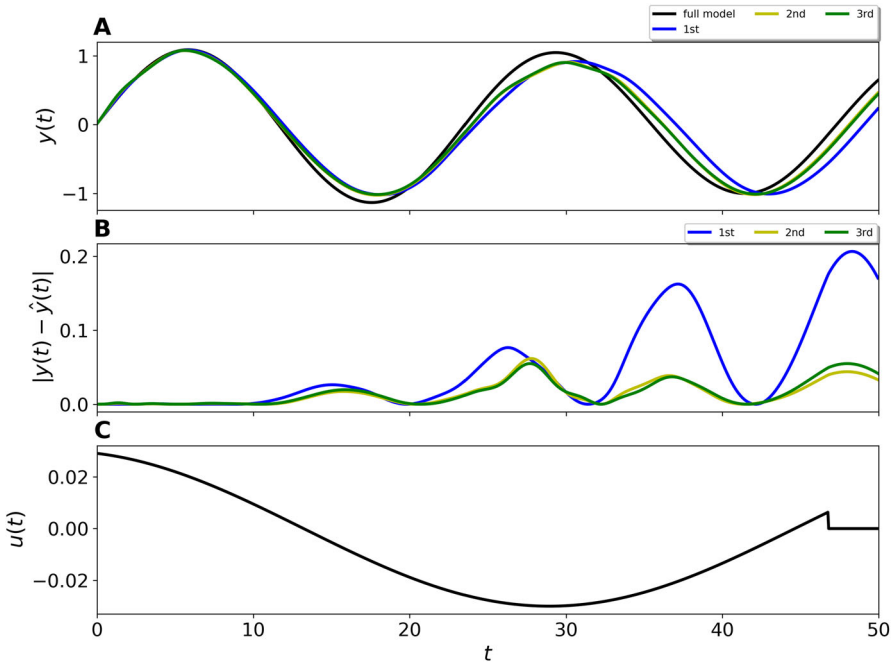

Fig. 3 Results are illustrated for the proposed model identification strategy when a sinusoidal input shown in panel (**C**) is applied to the radial isochron clock model from Eq. (22). The obtained outputs for first to third-order accuracy phase-isostable reduced models are compared with the full model output in panel (**A**) and the reduced model errors are shown in panel (**B**). The results follow the same trend as shown in Fig. 2

$$y(t) = \frac{1}{N}\sum_{j=1}^{N}(V_j(t)). \tag{24}$$

For the model above, $N = 10$ total neurons are considered in the population while V_k, h_k and r_k represent the transmembrane voltage and two gating variables, respectively, that determine the ionic currents for neuron k. $C_m = 1\ \mu F/\text{cm}^2$ and $i_k(t) = \upsilon_k u(t)$ where υ_k is a sensitivity parameter for the model incorporated through the input given by $\upsilon_k = 1 + 0.05k$ for $k = 1, \ldots, N$. The baseline current of the kth neuron, I_k^b, is defined by using $I_k^b = 5 + 0.35k$ for $k = 1, \ldots, N$. Finally, $u(t)$ represents a transmembrane current applied identically to each neuron. Reference (Rubin and Terman 2004) contains a full description of the remaining ionic currents and gating variables. For the neural model, we consider the mean of the transmembrane voltage of all the ten neurons as the observable; this is represented by $y(t)$ in (24).

In the limit as time approaches infinity, taking u = 0, the system observable exhibits dynamics that settle to a nearly periodic orbit. We apply our proposed model identification strategy using multiple isostable coordinates to derive a reduced phase-amplitude model of the form (8). This reduced model is identical to the one obtained in Sect. 4.1 except that we consider multiple isostable coordinates up to first-order accuracy for model identification. The network is based upon the architecture repre-

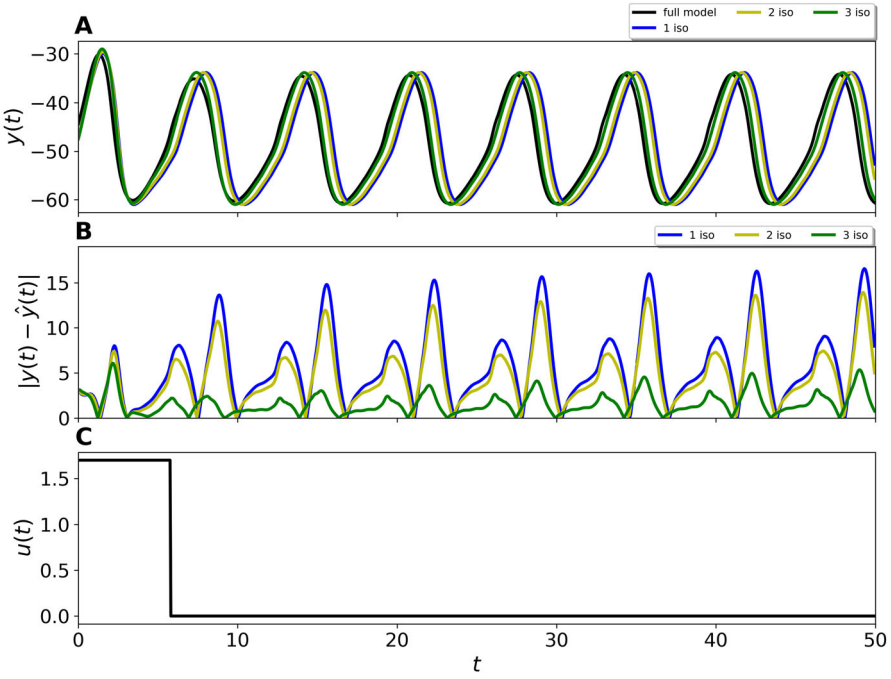

Fig. 4 The proposed model identification algorithm is applied to the neural model from Eq. (24). Models of the form (8) are obtained for one, two and three isostable coordinates that are valid to first order of accuracy in the expansion in the phase-amplitude coordinates. Panel (**A**) shows the response to the applied input from panel (**C**). Panel (**B**) shows the corresponding error between the full- and reduced-order models

sented in Fig. 1. To generate the training dataset, a set of hundred inputs of the form $u(t) = \epsilon(1 - \xi(t - t_s))$ for $t \in [0, 50]$ are considered where $\xi(t)$ is the Heaviside step function, ϵ represents the input magnitude, and t_s defines the input duration. For our set of one hundred trials, both the input magnitude and duration are chosen randomly according to $\epsilon = a_2 + (b_2 - a_2)\text{rand}(0, 1)$ and $t_s = a_3 + (b_3 - a_3)\text{rand}(0, 1)$ with $a_2 = -0.6$, $b_2 = 0.6$, $a_3 = 2$ and $b_3 = 8$. Once the resulting outputs have been extracted, the ADAM optimizer is used for training with MSE loss function from (21) and taking the learning rate to be 0.001. Also, the timestep for the update rule from Eq. (13) is taken to be $\Delta t = 0.05$ ms. The training follows the same implementation as described in Sect. 3.4.

The results for the neural model are illustrated through two different setups. In the first setup, the number of isostable coordinates is varied from one to three and the Fourier series expansion of the periodic functions is taken to first order. Meanwhile, the second setup employs a single isostable coordinate with the order of the Fourier expansions of the periodic functions varying from 1st to third order. For both setups, coefficients for one isostable taken to first-order Fourier expansion are obtained first and subsequently used for initialization when training models for subsequent Fourier expansions and isostable coordinates. As observed for the model in Sect. 4.1, training times gradually increase as the number of isostables employed for the model increases

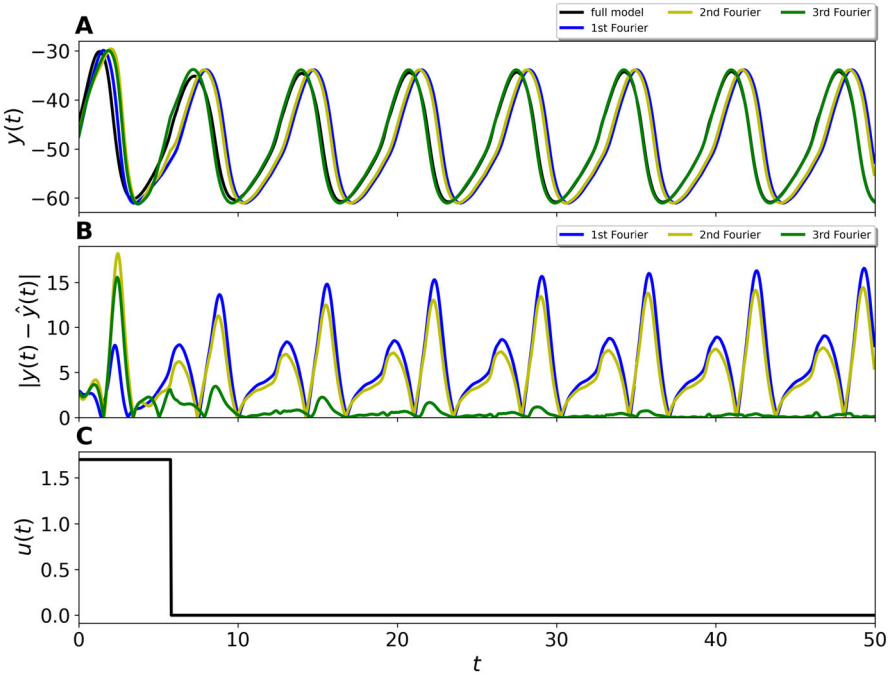

Fig. 5 The proposed model identification algorithm is applied to the neural model from Eq. (24). One isostable coordinate valid to first order of accuracy expansion is considered and models of the form (8) are obtained when considering first-, second- and third-order Fourier expansions for the reduced-order model terms. Panel (**A**) shows the response to the applied input from panel (**C**). Panel (**B**) shows the corresponding error

or higher Fourier series expansions are considered. For example, using a midgrade processor on a desktop computer, training reduced models with one, two and three isostable coordinates took 154, 190 and 227 s, respectively, to achieve convergence.

Figures 4 and 5 show the obtained results. All the learned models are validated using a test stimulus $u(t) = 1.7(1 - \xi(t - 5.8))$; note that the magnitude of the input used for validation is outside the magnitude range of the training inputs. In Fig. 4, Panel A shows the one, two and three isostable coordinate-based reduced-order model outputs compared to the full-order model output in response to the input from panel C. Panel B shows the associated error. The one isostable-based model has a MSE value of 48.9; considering two isostables subsequently reduces the MSE to 31.8 with three isostables giving the lowest error value of 3.8. Similarly, in Fig. 5, outputs from a single isostable coordinate-based reduced-order model valid to first-, second- and third-order Fourier series expansion are compared to the full-order model output when input from panel C is applied. Panel B shows the resulting error. The model with first-order Fourier expansion has a MSE value of 48.9; considering a second-order Fourier series expansion subsequently reduces the MSE to 38.5 with third-order Fourier series based model giving the lowest error value of 4.2.

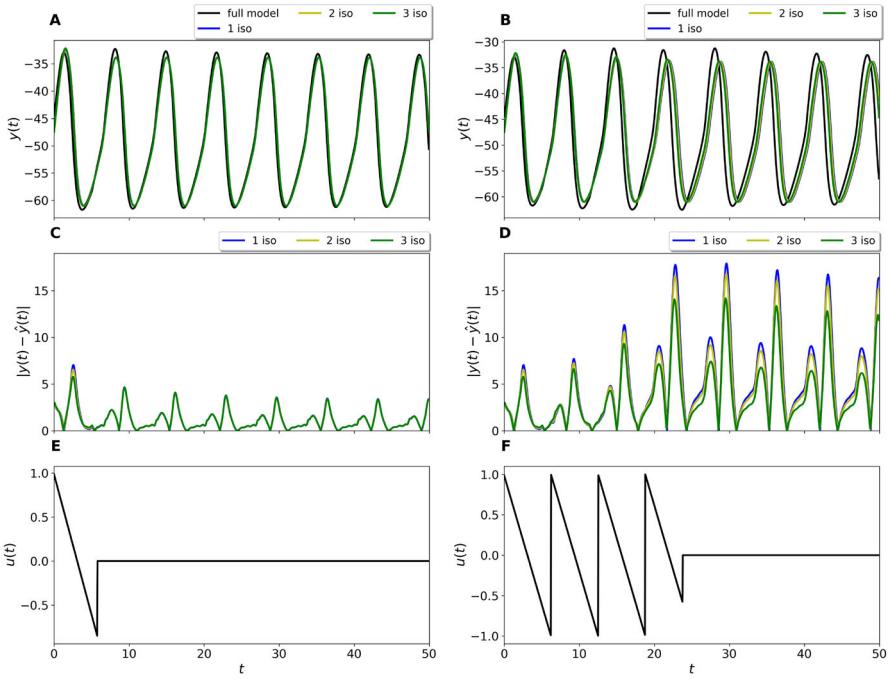

Fig. 6 Two sawtooth inputs applied for different durations to both the neural model from Eq. (24) and the reduced models are shown in panel (**E**) and (**F**). Varying number of isostable coordinates from one to three are considered for the reduced models through the proposed model identification algorithm and the resulting outputs are compared with the full model output in panels (**A**) and (**B**). To illustrate how effectively the reduced models can approximate the full model dynamics, error between the full model and the reduced model outputs is presented in panels (**C**) and (**D**)

To show that the proposed approach is valid for test inputs significantly different from the ones used for training, a saw tooth input of the form $u(t) = -2\left(\frac{t}{2\pi} - \left[\frac{1}{2} + \frac{t}{2\pi}\right]\right)$ and varying between -1 and 1 is applied for two different durations of $t_s = 5.85$ and $t_s = 23.85$ before being turned off as shown in Fig. 6. For the smaller duration test input in panel E, the reduced models are able to approximate the full model output as evident from panel A and the error shown in panel C. However, the difference in the reduced models for different number of isostable coordinates becomes more apparent when the longer duration input from panel F is applied. As seen in panels B and D, the one (resp., three) isostable reduced model has the highest (resp., lowest) error.

4.3 Model Reduction of Spike Rates of Neural Populations Using Adaptive Phase-Isostable Reduction

For the final illustration, the neural model from Eq. (24) is considered with the proposed model identification strategy in conjunction with the adaptive phase-amplitude reduction discussed in "Appendix 1." The only difference in the model is in $i_k(t) = \upsilon_k u(t)$

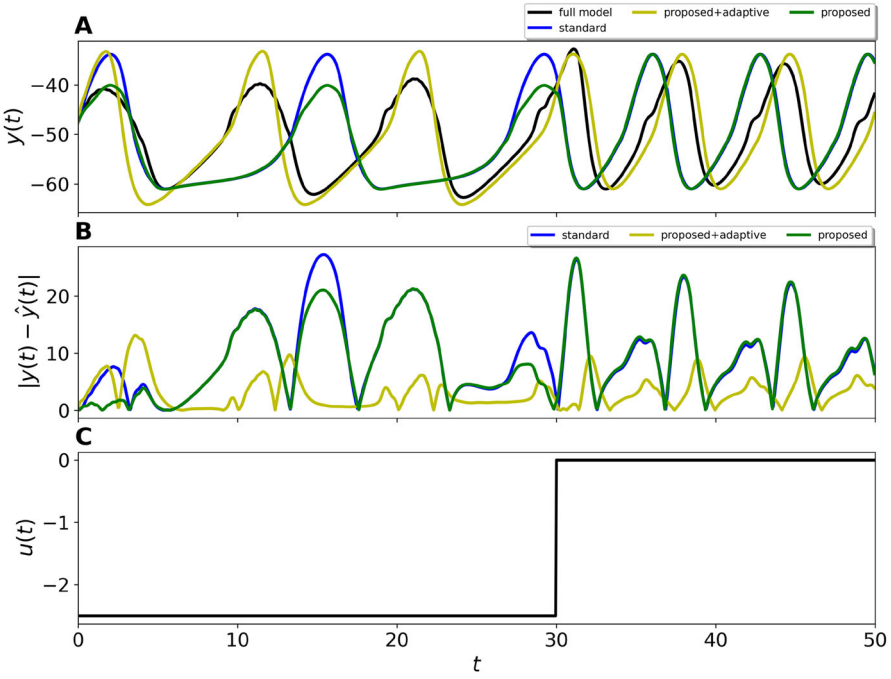

Fig. 7 The proposed model identification algorithm is combined with adaptive phase-amplitude reduction before being applied to the neural model from Sect. 4.2. Three isostable coordinates valid to first order of accuracy expansion are considered for the adaptive phase-amplitude reduction and then compared with the standard phase reduction as well as output from the proposed strategy alone. Panel (**A**) shows the respective responses to the applied input from panel (**C**). Panel (**B**) shows the corresponding error

for $\upsilon_k = 1 + 0.05k$ and $k = 1, \ldots, N$ where the adaptive parameter is now incorporated into the model though the applied input from a range of allowable values $u \in [-3, 0]$ in order to ultimately generate data for adaptive phase-amplitude reduction. When implementing the proposed model inference strategy, we consider three discrete, constant values of $p = \{-3, -1.5, 0\}$ and learn the corresponding phase and isostable response curves with the neural network following the same procedure as detailed in Sect. 3.4 for each discrete value of p. The terms $D(\theta, p)$ and $Q(\theta, p)$ are computed through the process described in "Appendix 1." Once training is finished, the terms of the resulting reduced-order model (A4) are interpolated for $p \in [0, 3]$ using the information inferred for the discrete values $p = \{-3, -1.5, 0\}$. We use the update rule

$$G_p = V^{\dagger}(-R(u - p)) - \zeta V^{*}\Psi, \tag{25}$$

where $\Psi = \begin{bmatrix}\psi_1 & \psi_2 & \psi_3\end{bmatrix}^T$, $R = \begin{bmatrix}I_1 & I_2 & I_3\end{bmatrix}^T$, $V = \begin{bmatrix}Q_1 & Q_2 & Q_3\end{bmatrix}^T$, and † represents the pseudoinverse. This update rule is similar to one used previously in Wilson (2022b) and is chosen to keep the amplitude coordinates small as discussed in "Appendix 1."

The derived update rule can then be used in conjunction with the rest of the equations from (A4) to obtain the adaptive phase-amplitude reduced model representation. The resulting model is validated by applying a step input of the form $u(t) = -2.5(1-\xi(t-30))$ for $t \in [0, 50]$ where ξ is the Heaviside function. To evaluate the performance of the proposed strategy combined with the adaptive reduction in approximating the full model dynamics, the output from the adaptive phase-amplitude reduced model is compared against standard phase reduction of the form shown in (6) where only phase information is used to approximate the full model output and also, with the standalone proposed strategy output which considers three isostable coordinates, valid to first order of accuracy around the limit cycle and expanded to fifth order of Fourier series expansion, without combining it with the adaptive reduction.

Figure 7 illustrates the results obtained when the input from Panel C is applied to the models. Looking at Panel A, the proposed strategy plus adaptive reduction performs far better than the other two strategies. Meanwhile, the standard phase reduction is limited to considering only small inputs and does not perform well. The proposed strategy without adaptive reduction performs moderately better than the standard phase reduction but fails as well due to the fact that it only considers the nominal periodic orbit which makes it unable to replicate the full model output effectively. Also, the magnitude of the applied input is much larger than what the proposed approach's reduced model form, shown in (8), can handle. Standard phase reduction has the highest MSE value of 142.1 followed by the proposed strategy without adaptive reduction with a MSE value of 126.4. The lowest MSE value of 17.6 is generated by the combination of proposed strategy and adaptive reduction.

5 Discussion and Conclusion

In this work, we propose a data-driven reduced-order model identification strategy for limit cycle oscillators using phase and isostable coordinates. We start by considering a phase-amplitude coordinate-based reduced model, shown in Eq. (8), to represent the full model dynamics in a low order basis. Equations (9), (10) and (11) show how the unknown terms are then Taylor expanded in the basis of isostable coordinates and then transformed into a lifted basis comprised of Fourier series coefficients that need to be learned. To facilitate learning, these equations are then written as a composition of known nonlinear functions and unknown linear functions containing the Fourier coefficients. Based upon this reframing, one can utilize an artificial neural network by using the training data to learn and identify an accurate approximation for the unknown weights of the linear relationships. The proposed model identification strategy is illustrated through a collection of nonlinear models and also utilized in conjunction with adaptive phase reduction while considering the dynamics of a synaptically coupled neuronal population.

Previous work Ahmed et al. (2022) presented a related model identification strategy for nonlinear dynamical systems with a stable fixed point. Our proposed strategy is an extension of the work from Ahmed et al. (2022) in that it is specifically developed for systems with oscillatory dynamics. To accomplish this, our proposed strategy must find unknown Fourier series coefficients comprising the terms from Eqs. (9), (10) and (11)

greatly increasing the number of coefficients needed to be learned as compared to the approach in Ahmed et al. (2022) for the same order of accuracy. Comparing the method from the current work with the strategy from prior work (Ahmed et al. 2022), the neural network for this proposed work requires more layers to account for the non-static phase coordinate including the nontrainable layer for the functional approximation of the periodic orbit. To account for this added complexity, Fourier expansion of the unknown model reduction coefficients is considered through deriving a non-trivial lifted basis involving the phase and isostable coordinates with the input $U(t)$ as shown in Eqs. (16), (18) and (20). The proposed strategy provides a reduced-order model for capturing the full model oscillatory dynamics while requiring relatively small amounts of training data. All the models considered as examples were trained using only a set of hundred random inputs and then tested using an input that was qualitatively different from the training set. This showcases the proposed strategy's ability to learn a generalized reduced-order model representation despite limited availability of data. Furthermore, both the phase and isostable coordinates embedded within the neural network as functions of state space do not require prior estimates.

To improve the effectiveness and accuracy of the proposed strategy, a number of extensions could be considered. For example, during training when the inner loop is implemented where $\hat{y}(t)$ is obtained by evaluating the relation (13) in a recursive manner, any initial errors arising at early timesteps are propagated and potentially amplified as training proceeds toward later timesteps. To limit this error propagation, it could be worthwhile to consider modifying the loss function in such a way that it gives more importance to initial solutions. This can be done by, for example, adding an additional weighting term of the form $\sum_{i=1}^{\eta} \vartheta(i)(y(t_0 + i\Delta t) - \hat{y}(t_0 + i\Delta t))^2$ where $\vartheta(i)$ is the weighting factor. Furthermore, although the MSE loss as defined in Eq. (21) provides adequate performance for the presented examples, it still has a few notable shortcomings. For instance, when the natural frequency ω is not approximated accurately, the MSE loss is unable to account for any discrepancy in the frequency and hence, the predicted reduced model output does not match the frequency of the full model output. To mitigate this issue, it might be worthwhile to investigate loss functions that take the frequency of the oscillations into account. Another improvement could be realized by devising a method to infer both phase and isostable coordinates directly from the observable which could then be used for learning the coefficients through training the neural network. This might also help mitigate the error propagation issue caused by generating the phase and isostable coordinates from the network itself as indicated in Step 5 in Sect. 3.4. A similar idea was used in strategies presented in Wilson (2020a), Ahmed and Wilson (2021) for extracting both the phase and isostable coordinates utilizing time-delay embeddings of observables.

Considering improvements from a deep learning perspective, it might be worthwhile to consider a more generalized training dataset with more samples containing inputs of different types, especially when training high-dimensional models. Additionally, for all the dynamical systems considered in this work, systematic weight initialization is not considered for the neural networks. Various works have shown that appropriate weight initialization improves the neural network's convergence rate significantly (Kumar 2017; Yam and Chow 2000). This can have a significant impact, especially when a large number of coefficients need to be learned for the reduced model. Batch

normalization could also be used which makes the network more stable during training and converge faster as shown in Santurkar et al. (2018), Ioffe and Szegedy (2015). To incorporate normalization into the current implementation, a workaround would need to be devised in such a way that the phase-isostable-based network structure is not altered. Using the previous timestep's output as an input for the next timestep, as done in our proposed approach, is one of the key ideas in recurrent neural networks which are primarily used in deep learning for time series forecasting and natural language processing. Error accumulation as training proceeds from one timestep to the next is quite common in these neural networks and can contribute to issues with either exploding or vanishing gradients (Pascanu et al. 2013). This gradient issue can potentially cause the network to stop learning in case of vanishing gradients; in case of exploding gradients, it poses a serious memory bottleneck. Traditional approaches to solve this gradient problem use weight regularization by introducing a penalty term within the loss function or by doing proper neural network initialization at the start of training. Such techniques from deep learning literature can be adapted to improve our proposed approach for future works.

This material is based upon work supported by the National Science Foundation Grant No. CMMI-2140527.

Appendix A Adaptive Phase-Isostable Reduction

Adaptive phase-amplitude reduction is able to provide reduced model representations that are valid far beyond the weakly perturbed limit by considering a collection of periodic orbits that results from a change in a given parameter. Following the formulation in Wilson (2022a), start by considering a set of additional variables to those in Eq. (1), represented by $p \in \mathbb{R}$. Assume that the system's differential equation $\dot{x} = F(x, p)$ has a stable periodic orbit $x^\gamma(\theta, p)$ if the value of p is held constant and chosen from a set of allowable p. A set of isostable coordinates denoted by $\psi_1(x, p), \ldots, \psi_\beta(x, p)$ and corresponding phase $\theta(x, p)$ is also defined for each periodic orbit $x^\gamma(\theta, p)$. For each $x^\gamma(\theta, p)$, the associated phase coordinates are unique up to a constant shift. To define the set of equations for adaptive reduction, one can rewrite the dynamics $\dot{x} = F(x, u)$ as

$$\dot{x} = F(x, p) + U_e(x, u, p), \tag{A1}$$

where

$$\begin{aligned} U_e(x, u, p) &= F(x, u) - F(x, p) \\ &= \frac{\partial F}{\partial u}(u - p) + O(||u - p||^2), \end{aligned} \tag{A2}$$

where all partial derivatives above are evaluated at x where $u = p$. Above, $F(x, p)$ from (A1) represents the nominal behavior of the system when u is held at p. The term $U_e(x, u, p)$ can be thought of as an effective input. Letting p be non-static and

changing the variables to phase and isostable coordinates yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\theta(x,p) &= \frac{\partial\theta}{\partial x}\cdot\frac{\mathrm{d}x}{\mathrm{d}t}+\frac{\partial\theta}{\partial p}\cdot\frac{\mathrm{d}p}{\mathrm{d}t},\\ \frac{\mathrm{d}}{\mathrm{d}t}\psi_j(x,p) &= \frac{\partial\psi_j}{\partial x}\cdot\frac{\mathrm{d}x}{\mathrm{d}t}+\frac{\partial\psi_j}{\partial p}\cdot\frac{\mathrm{d}p}{\mathrm{d}t},\\ j &= 1,\ldots,\beta,\\ \frac{\mathrm{d}p}{\mathrm{d}t} &= G_p(p,\theta,\psi_1,\ldots,\psi_\beta),\end{aligned} \tag{A3}$$

where G_p determines how p changes in time. As described in Wilson (2022a), to consider both phase and amplitude dynamics simultaneously like in (A3), limits must be placed on how the magnitude of the input changes especially when considering large magnitude inputs which can take the state far from the nominal periodic limit cycle. The equation for the parameter p in (A3) allows the user to keep the state close to the periodic orbit by updating and choosing the adaptive parameter p along with the corresponding nominal periodic orbit from the set defined by the allowable values of p while keeping the amplitude dynamics, i.e., the isostable coordinates small. To simplify Eq. (A3) further, one can note that $\frac{\partial\theta}{\partial x}\cdot\frac{\partial x}{\partial t}$ and each $\frac{\partial\psi_j}{\partial x}\cdot\frac{\partial x}{\partial t}$ capture the phase and isostable dynamics, respectively, when p is kept constant and hence, can be replaced with the terms from standard phase-amplitude reduction, i.e., $\omega(p)+Z(\theta,p)(u-p)$ for phase and $\kappa_j(p)\psi_j+I_j(\theta,p)(u-p)$ for isostable coordinates. Similarly, the remaining terms can be identified as in Wilson (2022a) to yield the set of equations for the adaptive phase-isostable reduction

$$\begin{aligned}\dot\theta &= \omega(p)+Z(\theta,p)(u-p)+D(\theta,p)\dot p,\\ \dot\psi_j &= \kappa_j(p)\psi_j+I_j(\theta,p)(u-p)+Q_j(\theta,p)\dot p,\\ j &= 1,\ldots,\beta,\\ \dot p &= G_p(p,\theta,\psi_1,\ldots,\psi_\beta).\end{aligned} \tag{A4}$$

As shown in Wilson (2022a), for a given value of p held constant, the $D(\theta,p)\in\mathbb{R}$ term is given by $-\frac{\partial x^\gamma}{\partial p}\cdot\frac{\partial\theta}{\partial x}$ where $\frac{\partial x^\gamma}{\partial p}|_{\theta_0,p}\equiv\lim_{a\to0}(x^\gamma(\theta_0,p+a)-x^\gamma(\theta_0,p))/a$ and $\frac{\partial\theta}{\partial x}$ is evaluated in reference to x^γ_p. Similarly, $Q_j(\theta,p)\in\mathbb{R}$ is given by $-\frac{\partial x^\gamma}{\partial p}\cdot\frac{\partial\psi_j}{\partial x}$ where $\frac{\partial\psi_j}{\partial x}$ is evaluated with reference to $x^\gamma(\theta,p)$. As described in Wilson (2023), one can rewrite $D(\theta,p)$ as

$$D(\theta,p) = -H(\theta,p)-c_1(p), \tag{A5}$$

where $c_1(p)$ is a constant. As discussed in Wilson (2023), the term $H(\theta,p)$ can be found by substituting $\theta=\omega t$ along the limit cycle according to

$$H(\omega t,p) = \int_0^T \bar M(\omega s,p)\mathrm{d}s, \tag{A6}$$

where $\bar{M}(\omega t, p) = Z(\omega t, p) - \bar{Z}(p)$ and $\bar{Z}(p) = \frac{1}{T}\int_0^T Z(\omega s, p)\mathrm{d}s$. Note that above, the period T depends on p, which is constant in the integration. Similarly, for the Q_j term, as shown in Wilson (2023)

$$Q_j(\theta, p) = -\psi_{j,ss}(\theta, p), \tag{A7}$$

where $\psi_{j,ss}(\theta, p)$ is given by

$$\psi_{j,ss}(\theta, p) = -\frac{b_0}{\kappa_j} + \sum_{i=1}^{\infty}\left[-\frac{a_i\kappa_j\sin(i\theta)}{\kappa_j^2+\omega^2 i^2} - \frac{a_i i\omega\cos(i\theta)}{\kappa_j^2+\omega^2 i^2} - \frac{b_i i\omega\sin(i\theta)}{\kappa_j^2+\omega^2 i^2} - \frac{b_i\kappa_j\cos(i\theta)}{\kappa_j^2+\omega^2 i^2}\right], \tag{A8}$$

where b_0, a_i and b_i are the Fourier coefficients of the isostable response curve $I_j(\theta, p)$. Additionally, κ_j is the Floquet exponent and ω is the frequency from Eq. (A4), both of which are functions of p. A more detailed derivation is found in Wilson (2023). It is important to mention that the adaptive reduction model, shown in Eq. (A4), is specified by using the first-order terms of the standard phase-isostable reduction. Finally, in order to approximate the state when using reduced-order dynamics from Eq. (A4), one can write

$$x = x^{\gamma}(\theta, p) + \sum_{j=1}^{N-1}\psi_j g^j(\theta, p), \tag{A9}$$

where $g^j(\theta, p)$ is the Floquet eigenfunction associated with the periodic orbit $x^{\gamma}(\theta, p)$.

References

Ahmed, T., Wilson, D.: Exploiting circadian memory to hasten recovery from circadian misalignment. Chaos Interdiscip. J. Nonlinear Sci. **31**(7), 073130 (2021)

Ahmed, T., Sadovnik, A., Wilson, D.: Data-driven inference of low-order isostable-coordinate-based dynamical models using neural networks. Nonlinear Dyn. **111**, 2501–2519 (2022)

Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech. **25**(1), 539–575 (1993)

Brunton, S.L., Kutz, J.N.: Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge University Press, New York (2019)

Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016)

Budišić, M., Mohr, R., Mezić, I.: Applied Koopmanism. Chaos Interdiscip. J. Nonlinear Sci. **22**(4), 047510 (2012)

Fasel, U., Kaiser, E., Kutz, J.N., Brunton, B.W., Brunton, S.L.: Sindy with control: A tutorial. In: 2021 60th IEEE Conference on Decision and Control (CDC), pp. 16–21. IEEE (2021)

Geneva, N., Zabaras, N.: Transformers for modeling physical systems. arXiv:2010.03957 (2020)

Holmes, P., Lumley, J.L., Berkooz, G., Rowley, C.W.: Turbulence, Coherent Structures, Dynamical Systems and Symmetry. Cambridge University Press, New York (1996)

Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456. PMLR (2015)

Jordan, D., Smith, P.: Nonlinear Ordinary Differential Equations: An Introduction for Scientists and Engineers, vol. 10. Oxford University Press, Oxford (2007)
Juang, J.N., Pappa, R.S.: An eigensystem realization algorithm for modal parameter identification and model reduction. J. Guid. Control. Dyn. **8**(5), 620–627 (1985)
Kaiser, E., Kutz, J.N., Brunton, S.: Data-driven discovery of Koopman eigenfunctions for control. Mach. Learn. Sci. Technol. **2**, 035023 (2021)
Kumar, S.K.: On weight initialization in deep neural networks. arXiv:1704.08863 (2017)
Kutz, J.N., Brunton, S.L., Brunton, B.W., Proctor, J.L.: Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems. Society for Industrial and Applied Mathematics, Philadelphia (2016)
Kvalheim, M.D., Revzen, S.: Existence and uniqueness of global Koopman eigenfunctions for stable fixed points and periodic orbits. Phys. D Nonlinear Phenom. **425**, 132959 (2021)
Lusch, B., Kutz, J.N., Brunton, S.L.: Deep learning for universal linear embeddings of nonlinear dynamics. Nat. Commun. **9**(1), 1–10 (2018)
Mangan, N.M., Askham, T., Brunton, S.L., Kutz, J.N., Proctor, J.L.: Model selection for hybrid dynamical systems via sparse regression. Proc. R. Soc. A **475**(2223), 20180534 (2019)
Masri, S.F., Chassiakos, A.G., Caughey, T.K.: Structure-unknown non-linear dynamic systems: identification through neural networks. Smart Mater. Struct. **1**(1), 45 (1992)
Mauroy, A., Mezić, I., Moehlis, J.: Isostables, isochrons, and Koopman spectrum for the action-angle representation of stable fixed point dynamics. Physica D **261**, 19–30 (2013)
Mezić, I.: Analysis of fluid flows via spectral properties of the Koopman operator. Annu. Rev. Fluid Mech. **45**, 357–378 (2013)
Mezić, I.: Spectrum of the Koopman operator, spectral expansions in functional spaces, and state-space geometry. J. Nonlinear Sci. **30**, 2091–2145 (2019)
Monga, B., Wilson, D., Matchen, T., Moehlis, J.: Phase reduction and phase-based optimal control for biological systems: a tutorial. Biol. Cybern. **113**(1–2), 11–46 (2019)
Noack, B.R., Afanasiev, K., Morzynski, M., Tadmor, G., Thiele, F.: A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. J. Fluid Mech. **497**, 335–363 (2003)
Omidi, M., Arab, B., Rasanan, A.H., Rad, J.A., Parand, K.: Learning nonlinear dynamics with behavior ordinary/partial/system of the differential equations: looking through the lens of orthogonal neural networks. Eng. Comput. **38**(2), 1635–1654 (2022)
Ortin, S., Gutierrez, J.M., Pesquera, L., Vasquez, H.: Nonlinear dynamics extraction for time-delay systems using modular neural networks synchronization and prediction. Physica A **351**(1), 133–141 (2005)
Pan, S., Duraisamy, K.: Long-time predictive modeling of nonlinear dynamical systems using neural networks. Complexity (2018)
Pantazis, Y., Tsamardinos, I.: A unified approach for sparse dynamical system inference from temporal measurements. Bioinformatics **35**(18), 3387–3396 (2019)
Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning, pp. 1310–1318. PMLR (2013)
Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
Rowley, C.W., Mezić, I., Bagheri, S., Schlatter, P., Henningson, D.S.: Spectral analysis of nonlinear flows. J. Fluid Mech. **641**, 115–127 (2009)
Rubin, J.E., Terman, D.: High frequency stimulation of the subthalamic nucleus eliminates pathological thalamic rhythmicity in a computational model. J. Comput. Neurosci. **16**(3), 211–235 (2004)
Santurkar, S., Tsipras, D., Ilyas, A., Madry, A.: How does batch normalization help optimization? In: Advances in Neural Information Processing Systems, vol. 31 (2018)
Schaeffer, H.: Learning partial differential equations via data discovery and sparse optimization. Proc. R. Soc. A Math. Phys. Eng. Sci. **473**(2197), 20160446 (2017)
Schmid, P.J.: Dynamic mode decomposition of numerical and experimental data. J. Fluid Mech. **656**, 5–28 (2010)
Tan, Y., Saif, M.: Neural-networks-based nonlinear dynamic modeling for automotive engines. Neurocomputing **30**(1–4), 129–142 (2000)
Towne, A., Schmidt, O.T., Colonius, T.: Spectral proper orthogonal decomposition and its relationship to dynamic mode decomposition and resolvent analysis. J. Fluid Mech. **847**, 821–867 (2018)
Wilson, D.: A data-driven phase and isostable reduced modeling framework for oscillatory dynamical systems. Chaos Interdiscip. J. Nonlinear Sci. **30**(1), 013121 (2020a)

Wilson, D.: Optimal open-loop desynchronization of neural oscillator populations. J. Math. Biol. **81**(1), 25–64 (2020b)
Wilson, D.: Phase-amplitude reduction far beyond the weakly perturbed paradigm. Phys. Rev. E **101**(2), 022220 (2020c)
Wilson, D.: Data-driven inference of high-accuracy isostable-based dynamical models in response to external inputs. Chaos Interdiscip. J. Nonlinear Sci. **31**(6), 063137 (2021a)
Wilson, D.: Optimal control of oscillation timing and entrainment using large magnitude inputs: an adaptive phase-amplitude-coordinate-based approach. SIAM J. Appl. Dyn. Syst. **20**(4), 1814–1843 (2021b)
Wilson, D.: An adaptive phase-amplitude reduction framework without $\mathcal{O}(\epsilon)$ constraints on inputs. SIAM J. Appl. Dyn. Syst. **21**(1), 204–230 (2022a)
Wilson, D.: Data-driven identification of dynamical models using adaptive parameter sets. Chaos Interdiscip. J. Nonlinear Sci. **32**(2), 023118 (2022b)
Wilson, D.: A direct method approach for data-driven inference of high accuracy adaptive phase-isostable reduced order models. Phys. D Nonlinear Phenom. **446**, 133675 (2023)
Wilson, D., Djouadi, S.: Isostable reduction and boundary feedback control for nonlinear convective flows. In: 2019 IEEE 58th Conference on Decision and Control, pp. 2138–2143. IEEE (2019)
Wilson, D., Djouadi, S.M.: Adaptive isostable reduction of nonlinear PDEs with time varying parameters. IEEE Control Syst. Lett. **5**(1), 187–192 (2020)
Wilson, D., Ermentrout, B.: Greater accuracy and broadened applicability of phase reduction using isostable coordinates. J. Math. Biol. **76**(1–2), 37–66 (2018)
Wilson, D., Ermentrout, B.: Augmented phase reduction of (not so) weakly perturbed coupled oscillators. SIAM Rev. **61**(2), 277–315 (2019)
Wilson, D., Moehlis, J.: Extending phase reduction to excitable media: theory and applications. SIAM Rev. **57**, 201–222 (2015)
Wilson, D., Moehlis, J.: Isostable reduction of periodic orbits. Phys. Rev. E **94**(5), 052213 (2016)
Winfree, A.: The Geometry of Biological Time, 2nd edn. Springer, New York (2001)
Wray, J., Green, G.G.R.: Calculation of the volterra kernels of non-linear dynamic systems using an artificial neural network. Biol. Cybern. **71**(3), 187–195 (1994)
Yam, J.Y.F., Chow, T.W.S.: A weight initialization method for improving training speed in feedforward neural network. Neurocomputing **30**(1–4), 219–232 (2000)