Proceedings of Machine Learning Research vol vvv:1–17, 2025

Neural Port-Hamiltonian Differential Algebraic Equations for Compositional Learning of Electrical Networks

Cyrus Neary* CYRUS.NEARY@MILA.QUEBEC
Mila – Quebec AI Institute & Université de Montréal, Canada

Nathan Tsao* NATHAN.TSAO@UTEXAS.EDU
The University of Texas at Austin, USA

Ufuk Topcu UTOPCU@UTEXAS.EDU
The University of Texas at Austin, USA

Abstract

We develop compositional learning algorithms for coupled dynamical systems. While deep learning has proven effective at modeling complex relationships from data, compositional couplings between system components typically introduce algebraic constraints on state variables, posing challenges to many existing data-driven approaches to modeling dynamical systems. Towards developing deep learning models for constrained dynamical systems, we introduce *neural port-Hamiltonian differential algebraic equations* (N-PHDAEs), which use neural networks to parametrize unknown terms in both the differential and algebraic components of a port-Hamiltonian DAE. To train these models, we propose an algorithm that uses automatic differentiation to perform index reduction, automatically transforming the neural DAE into an equivalent system of so-called *neural ordinary differential equations* (N-ODEs), for which established model inference and back-propagation methods exist. The proposed compositional modeling framework and learning algorithms may be applied broadly to learn control-oriented models of dynamical systems in a variety of application areas, however, in this work, we focus on their application to the modeling of electrical networks. Experiments simulating the dynamics of nonlinear circuits exemplify the benefits of our approach: the proposed N-PHDAE model achieves an order of magnitude improvement in prediction accuracy and constraint satisfaction when compared to a baseline N-ODE over long prediction time horizons. We also validate the compositional capabilities of our approach through experiments on a simulated D.C. microgrid: we train individual N-PHDAE models for separate grid components, before coupling them to accurately predict the behavior of larger-scale networks.

Keywords: Physics-informed machine learning, port-Hamiltonian neural networks, neural differential algebraic equations, compositional deep learning

1. Introduction

Many physical systems, such as electrical networks, chemical reaction networks, and multi-body mechanical systems, comprise many interacting subsystems. Such systems not only exhibit complex dynamics, but are often subject to algebraic constraints that enforce compositional relationships between the subsystems (e.g., energy balance, conservation laws, or geometric couplings).

Deep learning methods that use physics-inspired architectures and training losses have shown promise in learning data-efficient models of *unconstrained* dynamical systems (Raissi et al., 2019; Chen et al., 2018; Djeumou et al., 2022, 2023). However, such methods are currently unable to learn models that respect the aforementioned algebraic constraints. This limitation renders compositional

* Indicates equal contribution. Authors listed alphabetically.

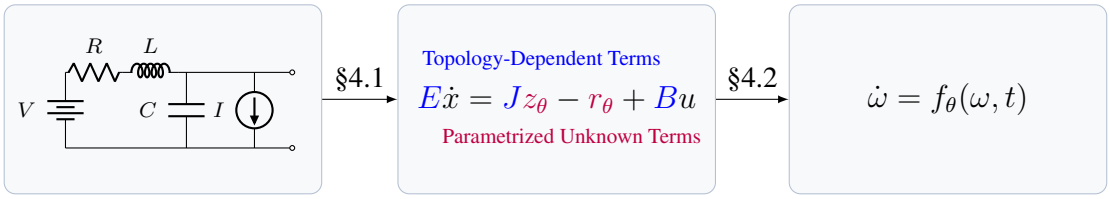

Figure 1: The proposed neural port-Hamiltonian differential algebraic equations (N-PHDAE). The proposed algorithm uses the topology of the subsystem interconnections to automatically construct a differential algebraic equation (DAE) whose unknown terms are parametrized using neural networks (§4.1). It then transforms the resulting neural DAE into an equivalent system of neural ordinary differential equations (§4.2) for inference and training (§4.3).

approaches to modeling challenging, acting as a barrier to the application of deep-learning-based models for the prediction and control of many real-world systems. For example, the complexity that arises from large numbers of interacting components can make monolithic approaches to learning system models intractable. Furthermore, in many applications, system-level data may not be available for training by a single algorithm. Moreover, the inability of existing deep learning methods to enforce critical constraints can result in models that are not interpretable or robust, and prone to failure in scenarios outside the training data distribution.

Towards addressing these limitations, we introduce *Neural Port-Hamiltonian Differential Algebraic Equations* (N-PHDAEs), a physics-informed and compositional deep learning approach to modeling dynamical systems subject to algebraic constraints. Such models may be used broadly to learn control-oriented models of dynamical systems in a variety of application areas, however, in this work, we focus primarily on their application to modeling electrical networks.

Figure 1 illustrates the proposed approach. The method begins by using the graph structure of the circuit to automatically construct the interconnection terms of a Port-Hamiltonian Differential Algebraic Equation (PHDAE) (Duindam et al., 2009; Van Der Schaft et al., 2014; Mehrmann and Morandin, 2019; Günther et al., 2021), which describes the interconnection topology of the system's components. The remaining terms in the PHDAE (which capture the nonlinear dynamics of the circuit components) are parametrized using neural networks (§4.1). We then use automatic differentiation to transform the resulting neural DAE into a system of Neural Ordinary Differential Equations (N-ODEs) (§4.2), which can be more easily trained and evaluated (§4.3).

To learn *compositional* models of electrical networks, we additionally propose a framework and algorithms to compose N-PHDAEs by coupling their inputs and outputs (§5), as is illustrated in Figure 2. Separate N-PHDAEs are trained on data generated by individual subsystems. The trained models are then coupled through an *interconnection matrix* that defines the input-output relationships at the circuit nodes where the couplings occur. The result is a composite N-PHDAE whose solution yields predictions for the overall system.

We demonstrate the advantages of the N-PHDAE through case studies on simulated electrical networks. A N-PHDAE model of the nonlinear FitzHugh-Nagumo circuit demonstrates the model's data efficiency and accuracy, as well as its ability to learn to satisfy the system's algebraic equations an order of magnitude more accurately than the considered baseline approach—a N-ODE that does not leverage prior physics knowledge or explicitly model coupling constraints. Next, we showcase the compositional modeling capabilities of the N-PHDAE by using it to simulate a large DC micro-

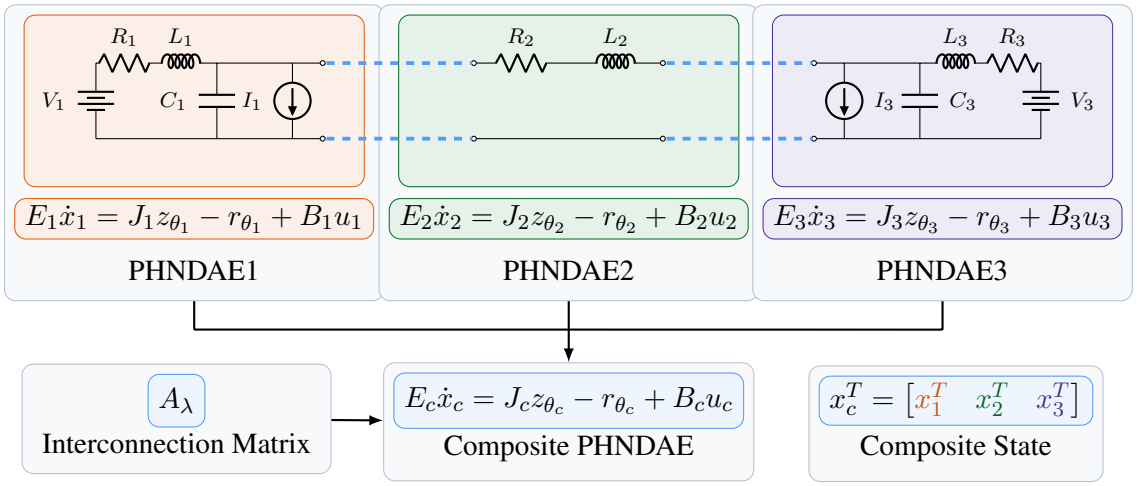

Figure 2: The compositionality of the proposed N-PHDAEs (§5). By coupling individually trained N-PHDAE models of subsystem dynamics, we obtain a composite N-PHDAE model that generates accurate predictions of more complex system dynamics, such as the DC microgrid example illustrated above and discussed further in §6.

grid. More specifically, we train multiple N-PHDAE models of distributed generation units (DGU) separately, before composing the learned DGU models in unseen configurations to accurately simulate larger networks of interacting electrical systems.

2. Related Work

Methods that leverage physics knowledge in deep learning algorithms have been studied extensively in recent years. For example, Lu et al. (2021); Raissi et al. (2019); Raissi (2018); Karniadakis et al. (2021); Han et al. (2018); Sirignano and Spiliopoulos (2018); Long et al. (2018, 2019) use neural networks and deep learning training algorithms to solve partial differential equations. By contrast, our work focuses on the problem of system identification—learning unknown dynamics from time series data. More closely related to our work, Neural Ordinary Differential Equations (N-ODEs) are a family of deep learning architectures that use neural networks to parametrize the right-hand side of ODEs. Originally proposed by Chen et al. (2018) in the context of generative modeling, N-ODEs provide a flexible framework for incorporating prior physics knowledge with data-driven models of dynamical systems, yielding models with improved data efficiency and generalization capabilities, especially when training data is limited (Djeumou et al., 2022, 2023; Rackauckas et al., 2020; Kidger, 2021; Neary, 2024; Zhong et al., 2021; Greydanus et al., 2019). However, N-ODEs struggle to model constrained dynamical systems, often relying on penalty-based methods to enforce known algebraic equations.

In particular, without significant modification, N-ODEs are unable to model systems that include algebraic constraints on the state variables. Such DAEs—systems of both differential and algebraic equations—arise frequently in the modeling of physical systems, especially when accounting for couplings between distinct subsystems. Recently, several approaches to learning DAEs from data have been proposed (Xiao et al., 2022; Moya and Lin, 2023; Huang et al., 2024; Koch et al., 2024). These methods rely on learned predictors for the algebraic states or latent variables, which are then

used to integrate the DAE. However, such learned predictors are often trained to fit the training data directly, without leveraging prior physics-based inductive biases that could enhance data efficiency and generalization. By contrast, we directly incorporate algebraic constraints into the neural network architecture and inference procedure, enabling the training of constraint-respecting and data-efficient models of dynamical systems.

Closely related to our work, Zhong et al. (2020); Desai et al. (2021); Neary and Topcu (2023); Tan et al. (2024); Duong et al. (2024); Beckers (2023) also develop algorithms that leverage the mathematical structures defining port-Hamiltonian (PH) systems to learn dynamics models that enjoy key PH properties, e.g., passivity. Meanwhile, Xu et al. (2022); Furieri et al. (2022); Plaza et al. (2022) use neural networks to parametrize controllers for systems with known port-Hamiltonian dynamics. Neary and Topcu (2023) focus on leveraging the properties of PH systems to build compositional learning algorithms. We extend this work by introducing the first port-Hamiltonian neural networks that explicitly account for algebraic constraints, broadening their applicability to a broader range of problems and enabling more intuitive compositional coupling of subsystem models.

3. Background

Port-Hamiltonian Differential Algebraic Equations. The port-Hamiltonian (PH) framework enables compositional approaches to modeling complex, interconnected systems in a structured and modular way. Conceptually, the dynamics of individual PH systems are governed by the system's Hamiltonian function H, energy dissipation terms, and control inputs. Separate PH systems can be coupled via energy exchanges through so-called port variables to obtain new PH systems that represent the dynamics of larger composite systems. We refer to Van der Schaft and Maschke (2013); Van Der Schaft et al. (2014) for further details.

In this work, we consider port-Hamiltonian differential algebraic equations (PHDAEs)—a broad class of PH systems that include both differential and algebraic equations, and can be written as

$$\frac{d}{dt}Ex(t) = Jz(x(t))) - r(z(x(t))) + Bu(t). \tag{1}$$

Here $x \in \mathbb{R}^n$ is the system's state, $z : \mathbb{R}^n \to \mathbb{R}^k$ is the effort (a vector-valued function that includes gradients of the system's Hamiltonian function $H(x)$), $E \in \mathbb{R}^{k\times n}$ is the flow matrix, $J \in \mathbb{R}^{n\times n}$ is the skew-symmetric interconnection matrix, $r : \mathbb{R}^n \to \mathbb{R}^k$ is the dissipation term, $B \in \mathbb{R}^{n\times m}$ is the port matrix, and $u \in \mathbb{R}^m$ is the control input (Mehrmann and Morandin, 2019).

Port-Hamiltonian Differential Algebraic Equations for Electrical Networks. PHDAEs provide a general framework for energy-based modeling and control of complex dynamics. Electrical circuits may be modeled in PHDAE form as follows (Günther et al., 2021).

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} A_C & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} q_C \\ \phi_L \\ e \\ j_V \end{bmatrix} &= \begin{bmatrix} 0 & -A_L & 0 & -A_V \\ A_L & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ A_V & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} e \\ \nabla H(\phi_L) \\ q(q_C) \\ j_V \end{bmatrix} \\ &\quad - \begin{bmatrix} A_R g(A_R^T e) \\ 0 \\ A_C^T e - q(q_C) \\ 0 \end{bmatrix} + \begin{bmatrix} -A_I & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -I \end{bmatrix}\begin{bmatrix} i(t) \\ v(t) \end{bmatrix}\end{aligned} \tag{2}$$

We note that equation (2) follows the form of (1), where the state $x = \begin{bmatrix} q_C^T & \phi_L^T & e^T & j_V^T \end{bmatrix} \in \mathbb{R}^n$ is composed of the capacitor charges $q_C \in \mathbb{R}^{n_C}$, inductor magnetic fluxes $\phi_L \in \mathbb{R}^{n_L}$, nodal voltages excluding ground $e \in \mathbb{R}^{n_v}$, and current across voltage sources $j_V \in \mathbb{R}^{n_V}$. The port-Hamiltonian system matrices E, J, and B depend on incidence matrices $(A_C, A_R, A_L, A_V, A_I)$, where each $A_s \in \{-1, 0, 1\}^{n_v \times n_s}$ denotes the incidence matrices associated with the capacitors, resistors, inductors, voltage sources, and current sources, respectively. Here, n_s denotes the number of elements of each component type. The effort function z comprises the resistor voltage-current relation $g : \mathbb{R}^{n_R} \to \mathbb{R}^{n_R}$, capacitor voltage-charge relation $q : \mathbb{R}^{n_C} \to \mathbb{R}^{n_C}$, and the Hamiltonian function $H : \mathbb{R}^{n_L} \to \mathbb{R}$. The system input $u^T = \begin{bmatrix} i(t)^T & v(t)^T \end{bmatrix} \in \mathbb{R}^m$ is comprised of the time-dependent magnitudes of the circuit's current and voltage sources.

Semi-Explicit, Index-1 Differential Algebraic Equations. More generally, the algorithms we propose apply to any *index-1* DAEs that may be expressed in the following *semi-explicit* form:

$$\dot{v} = f(v, w, t), \qquad 0 = h(v, w, t). \tag{3}$$

That is, we assume the system state $x \in \mathbb{R}^n$ may be separated into so-called *differential* $v \in \mathbb{R}^d$ and *algebraic* $w \in \mathbb{R}^a$ components, such that $x = (v, w) \in \mathbb{R}^n$. An index-1 DAE may be transformed into an equivalent system of ODEs by taking the time derivative of the algebraic equations $0 = h(\cdot)$. We note that the PHDAE (2) can be rewritten in the form of (3), with differential states $v = \begin{bmatrix} q_C^T & \phi_L^T \end{bmatrix}^T$ and algebraic states $w = \begin{bmatrix} e^T & j_V^T \end{bmatrix}^T$.

Neural Ordinary Differential Equations. Neural ordinary differential equations (neural ODEs) are a class of deep learning model that uses neural networks to parameterize the right-hand side of an ODE. That is, neural networks are used to parametrize the unknown components of the time derivative $f_\theta(\cdot)$ of the state y, as in (4). f_θ is then numerically integrated to obtain predictions of the state at some future time $k + T$, as in (5).

$$\dot{y} = f_\theta(y, t) \tag{4}$$

$$y(k + T) = y(k) + \int_k^{k+T} f_\theta(y, t) dt \tag{5}$$

4. Neural Port-Hamiltonian Differential Algebraic Equations

We now present neural port-Hamiltonian differential algebraic equations (N-PHDAEs): a physics-informed and compositional deep learning approach to modeling dynamical systems subject to algebraic constraints. As described in §1, N-PHDAEs may be used to learn control-oriented models of dynamical systems in a variety of application areas, however, in this work, we focus on their application to modeling electrical networks.

4.1. Constructing Neural Port-Hamiltonian DAEs

To begin, we assume that the interconnection topology of the system's components is known a priori. However, models of the individual components are unknown and must be learned from data. Mathematically, we use the interconnection topology to derive the matrices E, J, and B in equation (1), and we parametrize the unknown effort $z(\cdot)$ and dissipation $r(\cdot)$ functions using neural networks, as is illustrated in Figure 1.

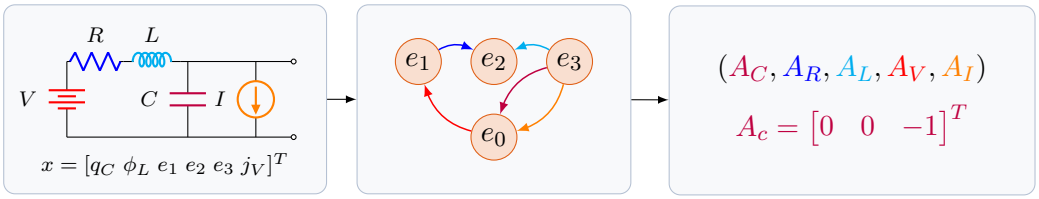

Figure 3: Constructing the matrix terms in the N-PHDAE from the interconnection topology of the system components. In the context of electrical circuits, a circuit schematic is used to construct a directed graph whose nodes represent junctions and whose edges represent electrical components. Component-specific incidence matrices are then obtained from the directed graph, which are in turn used to construct the matrices E, J, and B in equation (1).

In the context of electrical networks, matrices E, J, and B are expressed in terms of the component-specific incidence matrices A_C, A_R, A_L, A_V, and A_I. Figure 3 illustrates the process of extracting these incidence matrices from a circuit schematic. The procedure begins by transforming the schematic into a directed graph $(\mathcal{V}, \mathcal{E})$. The nodes $\mathcal{V}$ correspond to the circuit nodes, and the directed edges $\mathcal{E}$ represent the electrical components connecting the nodes (e.g. resistors or capacitors), where the edge direction aligns with the chosen convention for positive current flow. The component incidence matrices (A_i for $i \in \{C, R, L, V, I\}$) are then obtained by constructing the incidence matrix of the subgraph $(\mathcal{V}, \mathcal{E}_i)$ containing only the edges $\mathcal{E}_i$ corresponding to each component of type i (Günther et al., 2021). The matrices E, J, and B of the N-PHDAE are then defined in terms of the component incidence matrices using equation (2).

We parametrize the N-PHDAE's effort and dissipation functions using the additional physics information that is relevant to these terms in the context of modeling electrical circuits, see equation (2). More specifically, $z_\theta(x) := [e,\ \nabla H_\theta(\phi_L),\ q_\theta(q_C),\ j_V]^T$ and $r_\theta(x) := [A_R g_\theta(A_R^T e),\ 0,\ A_C^T e - q_\theta(q_C),\ 0]^T$ where $g_\theta(\cdot)$, $q_\theta(\cdot)$, and $H_\theta(\cdot)$ are parametrized as neural networks.

4.2. Transforming Neural Port-Hamiltonian DAEs into Systems of Neural ODEs

Evaluating and training the constructed N-PHDAEs directly is difficult due to the challenge of solving DAEs in general. However, under appropriate conditions on the system's interconnection topology (Günther et al., 2021), the N-PHDAEs that result from §4.1 will always be index-1 equations that may be converted into semi-explicit form (as described in §3).

In §4.1, we construct N-PHDAEs in the form of (1). We proceed by automatically identifying the differential $v \in \mathbb{R}^d$ and the algebraic $w \in \mathbb{R}^a$ components of the state x, and by converting the N-PHDAE into semi-explicit form $\dot{v} = f_\theta(v, w, u, t)$ and $0 = h_\theta(v, w, u, t)$, where $f_\theta(v, w, u, t)$ and $h_\theta(v, w, u, t)$ are functions of $E, J, B, z_\theta(\cdot), r_\theta(\cdot)$, and $u(t)$. We include derivations for these terms in Appendix A, however, we omit them here due to space constraints.

After converting the N-PHDAE to semi-explicit form, we use automatic differentiation to transform it into an equivalent system of N-ODEs via index reduction (Wanner and Hairer, 1996).

$$\begin{bmatrix} \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} f_\theta(v, w, u, t) \\ (\nabla_w h_\theta(v, w, u, t))^{-1} (\nabla_v h_\theta(v, w, u, t) f_\theta(v, w, u, t) + \nabla_t h_\theta(v, w, u, t)) \end{bmatrix} \tag{6}$$

Here $\nabla_v h_\theta(\cdot)$ and $\nabla_w h_\theta(\cdot)$ denote the Jacobian matrices of the algebraic constraint equations $h_\theta(\cdot)$ with respect to the differential v and algebraic w components of the state, respectively.

4.3. Evaluating and Training Neural Port-Hamiltonian Differential Algebraic Equations

The N-PHDAE inputs are the state $x(t)$ and control input $u(t)$ at time t, and the prediction horizon T. The model output is the predicted state at time $t+T$, i.e. $\hat{x}(t+T) = \text{N-PHDAE}(x, u, t, T)$, obtained by numerically integrating the right-hand side of (6) using any ODE solver.

Given a training dataset $\mathcal{D}$, consisting of trajectories τ of states and control inputs, we optimize the parameters θ of the N-PHDAE by minimizing the objective (7) using gradient-based methods.

$$\mathcal{L}(\theta, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \sum_{(x,u,t,T,y) \in \tau} \underbrace{||y - \text{N-PHDAE}(x, u, t, T)||_2^2}_{\text{State MSE}} + \alpha \underbrace{||h_\theta(x, u, t)||_2^2}_{\text{Algebraic Eqn. Penalty}} \tag{7}$$

Here, the target y is defined by the true state at the prediction time $y = x(t+T)$. The first term of $\mathcal{L}$ ensures the model predictions fit the trajectory data, and the second term of $\mathcal{L}$ encourages $h_\theta(x, u, t)$ to be as close to zero as possible. We note that in order to evaluate (6) and to train the N-PHDAE, the Jacobian $\nabla_w h_\theta(v, w, t)$ must be invertible. Experimentally, we found that incorporating the algebraic equation penalty into the loss function was both essential and empirically effective in ensuring that $\nabla_w h_\theta(v, w, t)$ remained invertible, thereby maintaining training stability. A detailed investigation of methods to ensure this property holds in all cases is left for future work.

5. Composing Port-Hamiltonian Neural Differential Algebraic Equations

We now propose a method to compose previously learned subsystem N-PHDAEs to obtain an accurate dynamics model for larger composite systems, without requiring additional training.

We define an *interconnection matrix* A_λ to specify the couplings between an arbitrary number of pre-defined subsystem N-PHDAEs. Intuitively, the entries of A_λ define couplings between the inputs and outputs of the various subsystems. In the context of electrical networks, the coupling relations are defined by introducing n_λ new edges between the nodes of distinct subsystems. Each new edge models a physical connection in the composite circuit, which may be modeled as a voltage source with a voltage drop of zero and a coupling current λ (Günther et al., 2021). The coupling leads to an additional term in Kirchhoff's current law, modeled with an incidence matrix $A_\lambda \in \{-1, 0, 1\}^{n_{v_c} \times n_\lambda}$ describing all the new edges of the composite system, where $n_{v_c} = \sum_{i=1}^{N} n_{v_i}$ is the number of non-grounded nodes in all subsystems. Given the subsystem N-PHDAEs and the interconnection matrix A_λ, the composite N-PHDAE is defined by

$$\begin{aligned}
\begin{bmatrix} A_C & 0 & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_C \\ \phi_L \\ e \\ j_V \\ \lambda \end{bmatrix} &= \begin{bmatrix} 0 & -A_L & 0 & -A_V & -A_\lambda \\ A_L & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ A_V & 0 & 0 & 0 & 0 \\ A_\lambda & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \nabla H_\theta(\phi_L) \\ q_\theta(q_C) \\ j_V \\ \lambda \end{bmatrix} \\
&\quad - \begin{bmatrix} A_R g_\theta(A_R^T e) \\ 0 \\ A_C^T e - q_\theta(q_C) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} -A_I & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} i(t) \\ v(t) \end{bmatrix},
\end{aligned} \tag{8}$$

where all subsystem state vectors are concatenated and matrices are stacked diagonally to obtain the corresponding quantities for the composite system.

$$s = \begin{bmatrix} s_1^T & \dots & s_N^T \end{bmatrix}^T, \qquad s \in \{q_C, \phi_L, e, j_V, g_\theta(A_R^T e), q_\theta(q_C), \nabla H_\theta(\phi_L)\}$$
$$A_P = \text{diag}(A_{P_1}, \dots, A_{P_N}), \qquad P \in (C, R, L, V, I)$$

The N-PHDAE of the composite system (8) can then be transformed into equivalent N-ODEs as in §4.3. The inputs to the composite N-PHDAE are defined by the concatenation of the subsystem differential states, algebraic states, and control inputs.

6. Experimental Results

To demonstrate the strengths of the proposed N-PHDAE, we present two simulation-based case studies. We begin by training a N-PHDAE model of a well-studied circuit with nonlinear dynamics, before demonstrating the proposed approach to compositional modeling through experiments involving interconnected *DC microgrids*. In all experiments, we parameterize the unknown electrical component relations g_θ, q_θ and Hamiltonian H_θ of the N-PHDAE (§4.1) as multi-layer perceptrons. The training datasets $\mathcal{D}$ consist of only 30 state trajectories, highlighting the data efficiency of the N-PHDAE. We refer the reader to the Appendix B for additional experimental details. Project code is available at https://github.com/nathan-t4/PHNDAE.

6.1. Learning the Nonlinear Dynamics of the FitzHugh-Nagumo Circuit

As an illustrative example, we begin by training a N-PHDAE model of the FitzHugh-Nagumo circuit (Izhikevich and FitzHugh, 2006), illustrated in Figure 4, which includes a nonlinear resistor R_1. We compare the performance of the N-PHDAE to a black-box N-ODE, i.e., a neural ODE that does not leverage physics-based priors, as presented in §2. The baseline N-ODE is parameterized using an MLP and trained on the same dataset to minimize the mean-squared error of the model's predictions.

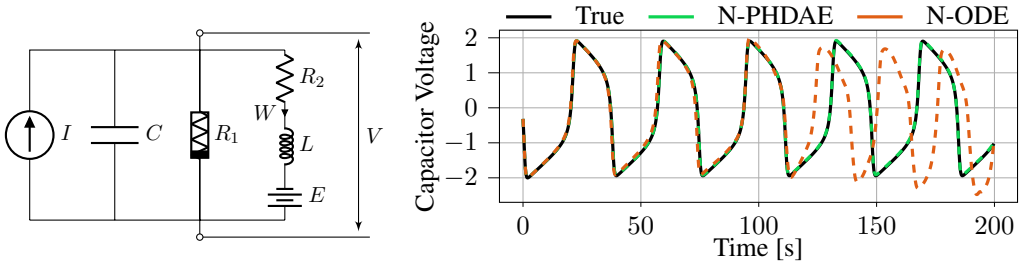

Figure 4: Left: The FitzHugh-Nagumo circuit. Right: The predicted dynamics of the system's state. The baseline N-ODE becomes increasingly inaccurate over long time horizons, while the N-PHDAE maintains consistently accurate predictions.

N-PHDAEs efficiently learn accurate models of complex dynamics. Figure (4) illustrates the predictions of the proposed N-PHDAE model over a 200-second horizon. For comparison, we also illustrate the predictions of a baseline N-ODE. The predictions from the N-PHDAE are more accurate than the N-ODE, especially for long horizons. This improvement is likely due to the N-PHDAE's ability to better satisfy the system's algebraic constraints, discussed below.

N-PHDAEs learn to satisfy constraint functions orders of magnitude more accurately than the baseline approach. Figure (5) illustrates the N-PHDAE's effectiveness at approximately satisfying the ground truth algebraic equations. In particular, we observe that the norm $||h(\hat{x}, u, t)||_2^2$ of the true algebraic equations is roughly an order of magnitude lower for the N-PHDAE than the N-ODE baseline, resulting in more accurate long-term predictions.

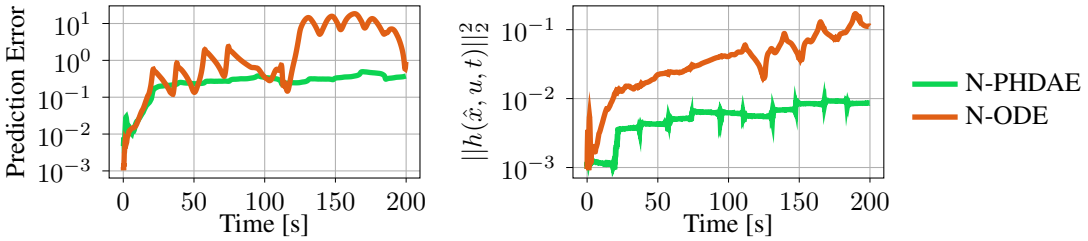

Figure 5: Left: The mean square error of the state predictions, as a function of prediction time. Right: Model violations of true algebraic equations. The N-PHDAE model satisfies the ground truth algebraic equations an order of magnitude more effectively than the N-ODE baseline.

6.2. Learning Compositional Models of Coupled DC Microgrids

Next, we demonstrate the compositional approach to learning introduced in §5 via experiments simulating a DC microgrid—a low-voltage power grid operating independently from a main grid (Cucuzzella et al., 2018). The DC microgrids we consider are powered by distributed generation units (DGUs), such as renewal energy sources, and are interconnected by transmission lines to disperse and store energy. We first train a N-PHDAE to model the dynamics of a DGU. Figure (6), illustrates the N-PHDAE's state predictions, prediction error, and a measure of the extent to which its predictions violate constraints. As for the FitzHugh-Nagumo circuit, the N-PHDAEs modeling individual DGUs output accurate and constraint-satisfying state predictions.

Compositions of pre-trained N-PHDAE submodels accurately simulate a larger DC microgrid without additional training. As illustrated in Figure 2 we then compose the learned DGU models by connecting them with transmission lines represented by known PHDAE models. To do so, we use the composition procedure detailed in §5. We configure the DC microgrid as a complete graph with 10 nodes, with DGUs as the nodes and transmission lines as the edges. In Figure (7), we plot

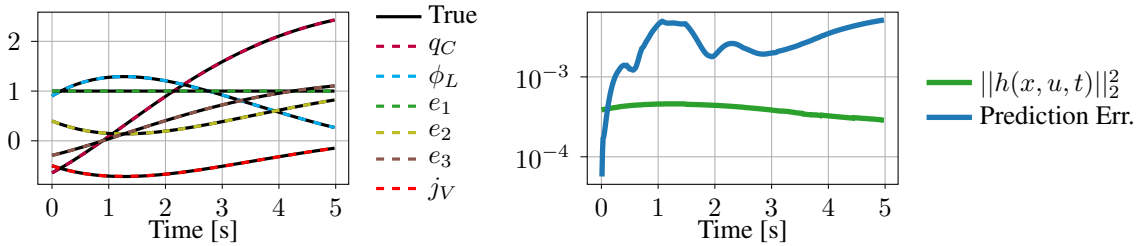

Figure 6: Left: N-PHDAE predictions of a DGU state trajectory. Right: Mean squared error, and violation of ground truth algebraic equations, of the predicted trajectory.

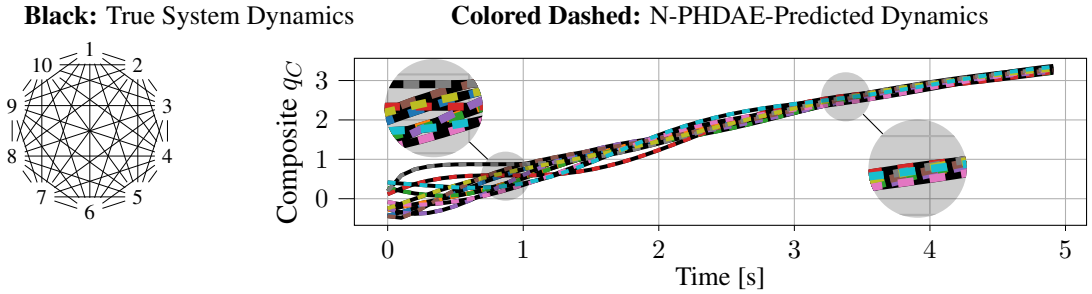

Figure 7: Left: The fully connected DC microgrid topology, consisting of 10 interconnected N-PHDAE submodels. Right: The charge of each DGU in the microgrid, as predicted by the composite N-PHDAE. The composite model accurately predicts the grid dynamics without additional training.

the trajectories of the capacitor charges q_C within the 10 interconnected DGUs, as predicted by the composite N-PHDAE representing the entire microgrid. We observe that the composite N-PHDAE model outputs accurate predictions for the entire DC microgrid, despite requiring no additional training data from the true composite system itself.

The composite N-PHDAE model continues to satisfy system constraints. Despite the interconnected dynamics of the DC microgrid, Figure 8 illustrates that the composition of learned N-PHDAE submodels continues to exhibit small prediction errors and only small violations of the true algebraic constraints. Indeed, we observe that the magnitude of these values are comparable to those of N-PHDAE submodels representing the individual DGUs, illustrated in Figure 6. The proposed N-PHDAEs enable effective compositional approaches to learning models of coupled systems.

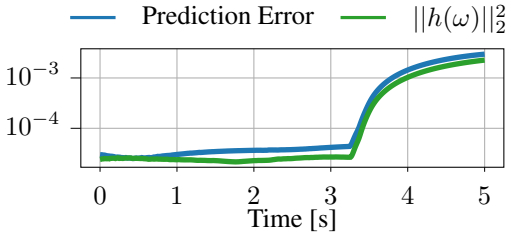

Figure 8: The state and constraint error trajectories of the composite DC microgrid.

7. Conclusions

We introduce Neural Port-Hamiltonian Differential Algebraic Networks (N-PHDAEs), a class of physics-informed neural networks that enables compositional approaches to learning system dynamics. The proposed N-PHDAEs use neural networks to parameterize unknown terms in port-Hamiltonian dynamical systems, even when said unknown terms are included in algebraic constraints between state variables. We propose algorithms for model inference and training that use automatic differentiation to transform the parametrized differential algebraic equations into equivalent systems of Neural Ordinary Differential Equations (N-ODEs), which may be evaluated more easily. Our experimental results demonstrate that N-PHDAEs learn data-efficient, accurate, and compositional models of electrical networks, enjoying an order of magnitude improvement to constraint satisfaction, in comparison to a baseline N-ODE. Future work will apply the proposed compositional learning algorithms to a broader range of constrained dynamical systems.

References

Thomas Beckers. Data-driven bayesian control of port-hamiltonian systems. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 8708–8713. IEEE, 2023.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Michele Cucuzzella, Sebastian Trip, Claudio De Persis, Xiaodong Cheng, Antonella Ferrara, and Arjan van der Schaft. A robust consensus algorithm for current sharing and voltage regulation in dc microgrids. *IEEE Transactions on Control Systems Technology*, 27(4):1583–1595, 2018.

Shaan A Desai, Marios Mattheakis, David Sondak, Pavlos Protopapas, and Stephen J Roberts. Port-hamiltonian neural networks for learning explicit time-dependent dynamical systems. *Physical Review E*, 104(3):034312, 2021.

Franck Djeumou, Cyrus Neary, Eric Goubault, Sylvie Putot, and Ufuk Topcu. Neural networks with physics-informed architectures and constraints for dynamical systems modeling. In *Learning for Dynamics and Control Conference*, volume 168 of *PMLR*, 2022.

Franck Djeumou, Cyrus Neary, and Ufuk Topcu. How to learn and generalize from three minutes of data: Physics-constrained and uncertainty-aware neural stochastic differential equations. In *Conference on Robot Learning*, volume 229. PMLR, 2023.

Vincent Duindam, Alessandro Macchelli, Stefano Stramigioli, and Herman Bruyninckx. *Modeling and control of complex physical systems: the port-Hamiltonian approach*. Springer Science & Business Media, 2009.

Thai Duong, Abdullah Altawaitan, Jason Stanley, and Nikolay Atanasov. Port-hamiltonian neural ode networks on lie groups for robot dynamics learning and control. *IEEE Transactions on Robotics*, 40:3695–3715, 2024. doi: 10.1109/TRO.2024.3428433.

Richard FitzHugh. Impulses and physiological states in theoretical models of nerve membrane. *Biophysical Journal*, 1(6):445–466, 1961. ISSN 0006-3495. doi: https://doi.org/10.1016/S0006-3495(61)86902-6. URL https://www.sciencedirect.com/science/article/pii/S0006349561869026.

Luca Furieri, Clara Lucía Galimberti, Muhammad Zakwan, and Giancarlo Ferrari-Trecate. Distributed neural network control with dependability guarantees: a compositional port-hamiltonian approach. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168, pages 571–583. PMLR, 2022.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Michael Günther, Andreas Bartel, Birgit Jacob, and Timo Reis. Dynamic iteration schemes and port-hamiltonian formulation in coupled differential-algebraic equation circuit simulation. *International Journal of Circuit Theory and Applications*, 49(2):430–452, 2021.

Jiequn Han, Arnulf Jentzen, and E Weinan. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

Tom Hennigan, Trevor Cai, Tamara Norman, Lena Martens, and Igor Babuschkin. Haiku: Sonnet for JAX, 2020. URL http://github.com/deepmind/dm-haiku.

Yicun Huang, Changfu Zou, Yang Li, and Torsten Wik. Minn: Learning the dynamics of differential-algebraic equations and application to battery modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

E. M. Izhikevich and R. FitzHugh. FitzHugh-Nagumo model. *Scholarpedia*, 1(9):1349, 2006. doi: 10.4249/scholarpedia.1349. revision #123664.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Patrick Kidger. *On neural differential equations*. PhD thesis, University of Oxford, 2021.

James Koch, Madelyn Shapiro, Himanshu Sharma, Draguna Vrabie, and Jan Drgona. Neural differential algebraic equations. *arXiv preprint arXiv:2403.12938*, 2024.

Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, pages 3208–3216, 2018.

Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Lu Lu, Raphael Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G Johnson. Physics-informed neural networks with hard constraints for inverse design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132, 2021.

Volker Mehrmann and Riccardo Morandin. Structure-preserving discretization for port-hamiltonian descriptor systems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 6863–6868. IEEE, 2019.

Christian Moya and Guang Lin. Dae-pinn: a physics-informed neural network model for simulating differential algebraic equations with application to power networks. *Neural Computing and Applications*, 35(5):3789–3804, 2023.

J. Nagumo, S. Arimoto, and S. Yoshizawa. An active pulse transmission line simulating nerve axon. *Proceedings of the IRE*, 50(10):2061–2070, 1962. doi: 10.1109/JRPROC.1962.288235.

Cyrus Neary. *Engineering AI systems and AI for engineering: compositionality and physics in learning*. PhD thesis, The University of Texas at Austin, 2024.

Cyrus Neary and Ufuk Topcu. Compositional learning of dynamical system models using port-hamiltonian neural networks. In *Learning for Dynamics and Control Conference*, pages 679–691. PMLR, 2023.

Santiago Sanchez-Escalonilla Plaza, Rodolfo Reyes-Báez, and Bayu Jayawardhana. Total energy shaping with neural interconnection and damping assignment - passivity based control. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168, pages 520–531. PMLR, 2022.

Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning. *arXiv preprint arXiv:2001.04385*, 2020.

Maziar Raissi. Deep hidden physics models: Deep learning of nonlinear partial differential equations. *The Journal of Machine Learning Research*, 19(1):932–955, 2018.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

Kaiyuan Tan, Peilun Li, and Thomas Beckers. Physics-constrained learning of PDE systems with uncertainty quantified port-Hamiltonian models. In *Proceedings of the 6th Annual Learning for Dynamics amp; Control Conference*, pages 1753–1764. PMLR, 2024.

Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997. doi: 10.1137/1.9780898719574. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898719574.

Arjan Van Der Schaft, Dimitri Jeltsema, et al. Port-hamiltonian systems theory: An introductory overview. *Foundations and Trends in Systems and Control*, 1(2-3):173–378, 2014.

Arjan J Van der Schaft and Bernhard M Maschke. Port-hamiltonian systems on graphs. *SIAM Journal on Control and Optimization*, 51(2):906–937, 2013.

Gerhard Wanner and Ernst Hairer. *Solving ordinary differential equations II*, volume 375. Springer Berlin Heidelberg New York, 1996.

Tannan Xiao, Ying Chen, Shaowei Huang, Tirui He, and Huizhe Guan. Feasibility study of neural ode and dae modules for power system dynamic component modeling. *IEEE Transactions on Power Systems*, 38(3):2666–2678, 2022.

Liang Xu, Muhammad Zakwan, and Giancarlo Ferrari-Trecate. Neural energy casimir control for port-hamiltonian systems. In *IEEE 61st Conference on Decision and Control*, pages 4053–4058. IEEE, 2022.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Dissipative symoden: Encoding hamiltonian dynamics with dissipation and control into deep learning. *arXiv preprint arXiv:2002.08860*, 2020.

Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Benchmarking energy-conserving neural networks for learning dynamics from data. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144. PMLR, 2021.

Appendix A. Transforming the N-PHDAE into Semi-Explicit Form

We explain how to rewrite the N-PHDAE as an index-1 semi-explicit DAE, as described in §4.3. First, let the algebraic indices of E be the indices of the rows of E that are all zero, and the differential indices of E as the indices of the rows of E with nonzero elements. Since E is generally not invertible, we solve a least squares problem to rewrite the PHNDAE in semi-explicit form. For our experiments, we solve the least-square problem with reduced QR-decomposition (Trefethen and Bau, 1997).

Let $\bar{E}$ be the submatrix of E that contains the rows corresponding to differential equations (i.e. the rows with non-zero elements) and columns that multiply the differential states. Additionally, let Q and R denote the reduced QR-decomposition of $\bar{E}$, and $\bar{R}$ is the submatrix of R that contains the first d rows of R (where d is the number of differential state variables). The differential equations f of the PHDAE in semi-explicit form may then be expressed as

$$\dot{v} = \begin{bmatrix} \dot{q}_C \\ \dot{\phi}_L \end{bmatrix} = f_\theta(v, w, u, t) = \bar{R}^\dagger Q^T (J z_\theta(v, w, t) - r_\theta(v, w, t) + B u(t)).$$

The algebraic equations $h_\theta(\cdot)$ correspond to the remaining rows of the right-hand side of the N-PHDAE, i.e., the entries of the vector output of $J z_\theta(v, w, t) - r_\theta(v, w, t) + B u(t)$ that share indices with the algebraic variables.

$$0 = h_\theta(v, w, u, t) = [J z_\theta(v, w, t) - r_\theta(v, w, t) + B u(t)]_{\text{algIndeces}}.$$

Appendix B. Additional Experimental Details

B.1. Implementation Details

The code to reproduce all numerical experiments is implemented in Python using the Jax (Bradbury et al., 2018) and Haiku (Hennigan et al., 2020) libraries and available at https://github.com/nathan-t4/PHNDAE. All numerical experiments are trained for 100000 epochs with a batch size of 128, loss function hyper-parameter $\alpha = 0.01$ (§4.3), and optimized using Adam. The learning rate is initially set to 0.0001 for the PHNDAE and 0.001 for the baseline black-box NODE, and decays to zero with cosine annealing (Loshchilov and Hutter, 2016). The resistor and capacitor component relations g_θ, q_θ and Hamiltonian H_θ of the PHNDAE and the baseline black-box NODE are all parameterized with multi-layer perceptions, each with two hidden layers of 100 nodes and ReLU activation.

The training datasets are generated by rewriting the electrical network dynamics as a Port-Hamiltonian differential algebraic equation (2). Then, the Port-Hamiltonian differential algebraic equation is transformed to an equivalent ordinary differential equation using index reduction (§4.2), and the state at the next time-step is obtained through numerical integration using the fourth-order Runge-Kutta method with a fixed time-step (§4.3). For all numerical experiments, we generate 30 trajectories of 1000 time-steps for the training dataset and 10 trajectories of 10000 time-steps for the validation dataset.

B.2. FitzHugh-Nagumo Circuit

The FitzHugh-Nagumo model is a well-studied nonlinear dynamics model of excitable biological systems first introduced by FitzHugh (1961), and with an equivalent circuit derived by Nagumo

et al. (1962). The governing equations of the membrane potential V and recovery variable W are provided below, where I is the stimulus current.

$$\begin{aligned} \dot{V} &= V - V^3/3 - W + I \\ \dot{W} &= 0.08(V + 0.7 - 0.8W) \end{aligned} \tag{9}$$

The equivalent circuit representation shown in Figure (4) has parameter values of $R_2 = 0.8$, $L = 1/0.08$, $C = 1.0$, $E = -0.7$, and $I = 1.0$. The capacitor voltage represents the membrane potential V, and the inductor current represents the recovery variable W. The electrical component values are set to match the governing equations of the circuit to (9). The equivalent PHDAE of the FitzHugh-Nagumo circuit (2) has incidence matrices:

$$A_C = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \; A_R = \begin{bmatrix} 1 & -1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \; A_L = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}, \; A_V = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}, \; A_I = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

and known component relations:

$$\begin{aligned} r &\colon \begin{bmatrix} V_{R1} & V_{R2} \end{bmatrix} \mapsto \begin{bmatrix} V_{R1}^3 \,/\, 3 - V_{R1} & V_{R2} \,/\, R_2 \end{bmatrix} \\ q &\colon V_C \mapsto C V_C \\ H &\colon \phi_L \mapsto \phi_L^2 \,/\, 2L \end{aligned} \tag{11}$$

The initial conditions for V and W for the training datasets are sampled from the uniform distribution $U(-3.0, 3.0)$ with $\Delta t = 0.1$. The baseline black-box neural differential ordinary equation (NODE) is trained on the same dataset as the PHNDAE, and is trained to optimize the mean-squared error on the state.

$$\mathcal{L}(\omega, t) = \frac{1}{|\mathcal{D}|} \sum_{\tau \in \mathcal{D}} \sum_{(\omega, u, t) \in \tau} ||\omega - \text{NODE}(\omega)||_2^2 \tag{12}$$

Here ω is the state and u is the control input.

B.3. Microgrids

We use the direct current (DC) microgrid model introduced by Cucuzzella et al. (2018) for the compositional learning experiment. DC microgrids are small-scale power grids composed of distributed generation units, loads, and energy storage systems.

B.3.1. DISTRIBUTED GENERATION UNIT MODEL

Distributed generation units (DGU) are small-scale electricity generators that are an alternative to traditional power plants. For example, renewable energy sources can act as the power generation unit in the DGU model. The equivalent PHDAE of the distributed generation unit (2) has incidence matrices:

$$A_C = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \; A_R = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}, \; A_L = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}, \; A_V = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \; A_I = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \tag{13}$$

and component relations:

$$\begin{aligned} r &: V_R \mapsto V_R \,/\, R \\ q &: V_C \mapsto C V_C \\ H &: \phi_L \mapsto \phi_L^2 \,/\, 2L \end{aligned} \tag{14}$$

The training dataset for the distributed generation unit has $\Delta t = 0.01$ and parameter values $R_{dgu} = 1.2$, $L_{dgu} = 1.8$, and $C_{dgu} = 2.2$.

B.3.2. TRANSMISSION LINE MODEL

The transmission lines interconnect two distributed generation units and model the grid loads. The equivalent PHDAE of the transmission line model has incidence matrices:

$$A_C = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ A_R = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix},\ A_L = \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix},\ A_V = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},\ A_I = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

and component relations:

$$\begin{aligned} r &: V_R \mapsto V_R \,/\, R_{tl} \\ H &: \phi_L \mapsto \phi_L^2 \,/\, 2L_{tl} \end{aligned} \tag{16}$$

The transmission line models have parameter values R_{tl} and L_{tl} sampled from the uniform distribution $U(0.1, 2.0)$.

B.3.3. OBTAINING THE MICROGRID WITH COMPOSITION

Towards simulating microgrids via composition, we need an interconnection matrix A_λ which specifies how the various distributed generation units and transmission lines are interconnected. In the compositional learning experiments, the microgrid has a complete graph configuration with 10 nodes, with DGUs at the nodes and transmission lines at the edges. Due to space constraints, we do not include $A_\lambda \in \{-1, 0, 1\}^{165 \times 90}$ for the microgrid configuration as a complete graph with 10 nodes. Instead, for illustration purposes, we include A_λ for the microgrid configuration as a complete graph with 2 nodes:

$$A_\lambda = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & -1 \\ -1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{17}$$

We can then derive the composite PHNDAE by stacking the vectors, i.e. $q = \begin{bmatrix} q_1^T & q_2^T \end{bmatrix}$ and $A_C = \text{diag}(A_{C1}, A_{C2})$. However, we emphasize that our choice of the microgrid configuration is arbitrary; we can simulate a microgrid with any configuration using the compositional learning framework introduced in §5 if given the corresponding interconnection matrix A_λ.