Neuro-Reachability of Networked Microgrids

Yifan Zhou, *Member, IEEE*, and Peng Zhang, *Senior Member, IEEE*

***Abstract*—A neural ordinary differential equations network (ODE-Net)-enabled reachability method (*Neuro-Reachability*) is devised for the dynamic verification of networked microgrids (NMs) with unidentified subsystems and heterogeneous uncertainties. Three new contributions are presented: 1) An ODE-Net-enabled dynamic model discovery approach is devised to construct the data-driven state-space model which preserves the nonlinear and differential structure of the NMs system; 2) A physics-data-integrated (PDI) NMs model is established, which empowers various NM analytics; and 3) A conformance-empowered reachability analysis is developed to enhance the reliability of the PDI-driven dynamic verification. Extensive case studies demonstrate the efficacy of the ODE-Net-enabled method in microgrid dynamic model discovery, and the effectiveness of the *Neuro-Reachability* approach in verifying the NMs dynamics under multiple uncertainties and various operational scenarios.**

***Index Terms*—Networked microgrids, data driven, neural ordinary differential equation network, reachability analysis, conformance theory.**

I. INTRODUCTION

NETWORKED microgrids (NMs) allow microgrids to support coordinately various smart community functions [1] and help increase electricity resilience [2], [3]. However, two major challenges arise in the dynamic analysis of today's low-inertia NMs [4], which prevent NMs from serving as dependable resiliency resources: I) Lack of effective analytics to handle the combinatorial explosion in verifying the NMs dynamics under the infinitely many uncertain scenarios [5], and II) Unattainability of accurate models for each and every microgrid, especially the dynamic models of converters, loads and circuits [6], [7].

Reachability analysis is a novel method which can provably enclose all dynamic trajectories under uncertain perturbations and large disturbances in NMs [5], [8], [9]. It prevails over traditional time-domain simulations [10] and energy function approaches [11], [12] mainly due to the capability of processing infinitely many uncertain scenarios efficiently. Even though reachability analysis is proved to be a promising solution to Challenge I, Challenge II above has been a major obstacle that prevents it from being widely adopted in the planning and operations of NMs.

Learning reliable dynamic models for the unidentified subsystems from measurements, therefore, is of paramount importance for the data-driven NMs dynamic analysis. Koopman operator and dynamic mode decomposition are popular approaches to constructing linear approximation of nonlinear systems from data [13], [14], whereas they are inefficient to establish nonlinear ODE models for rapidly-fluctuating NMs subject to 'random walks' of operating points disturbed by uncertainties [15]. Machine learning emerges to be a promising approach to power system dynamics analysis, either for time-domain trajectory prediction [16] or for stability assessment [17]. While earlier studies have concentrated mainly on direct applications of neural networks to power system analysis [18], [19], a recent trend is to integrate power system characteristics with deep neural network (DNN) techniques, such as recurrent neural networks accounting for time-series power dynamics [20], [21], convolutional neural networks considering the grid-like data from power systems [22], [23], and physics-informed neural networks taking advantage of a priori knowledge of power system's physics models [24]. Nevertheless, discovering dynamic models behind data, which is a long standing open problem, is substantially more important in the sense of providing deep insights of the system dynamics and allowing for formal verification and control of the system.

The overarching goal of this paper is to establish a data-driven method well suited to discovering the strongly nonlinear NMs dynamics as well as to verifying the NMs dynamics under uncertainties. To this end, this paper devises a *Neuro-Reachability* method. The key innovation is to integrate the neural ordinary differential equations network (ODE-Net) with reachability analysis and conformance theory to allow for a data-driven formal verification of the NMs dynamics under uncertainties. The contributions of this work are threefold:

- An ODE-Net-enabled model discovery method is devised to construct a nonlinear ODE model for the uncertainty-perturbed NMs, which can best preserve the dynamic behaviours of NMs without assuming a priori any specific dynamic modes. This modeling approach can effectively address the data rich, information poor (DRIP) problem widely existing in today's microgrids.
- A physics-data-integrated (PDI) modelling approach is then introduced to combine both physics-based and data-driven ODE models, which enables reachability analysis to verify the PDI-NMs dynamics incorporating the hierarchical control of DERs and network transients.

Manuscript received September 10, 2020; revised March 5, 2021; accepted May 29, 2021. Date of publication June 1, 2021; date of current version December 23, 2021. This work was supported in part by National Science Foundation under Grant OIA-2040599. Paper no. TPWRS-01552-2020. *(Corresponding author: Peng Zhang.)*

The authors are with the Department of Electrical and Computer Engineering, Stony Brook University, NY 11794-2350 USA (e-mail: yifan.zhou.1@stonybrook.edu; p.zhang@stonybrook.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TPWRS.2021.3085706.

Digital Object Identifier 10.1109/TPWRS.2021.3085706

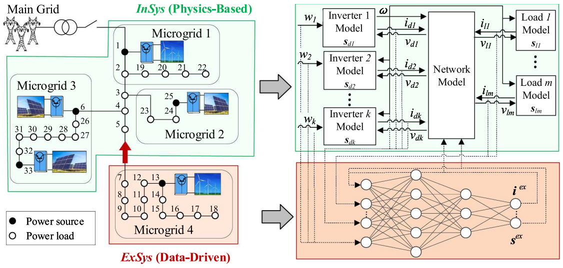


Fig. 1. Illustration of NMs modeling with physics-based *InSys* and data-driven *ExSys*.

- Reachability analysis for NMs is empowered with a conformance theory as a 'feedback' mechanism to further improve the reachset accuracy induced by possible inconformance of the PDI-NMs behaviours compared with the real NMs dynamics.

The remainder of the paper is organized as follows. Section II introduces the ODE-Net-enabled dynamic model discovery. Section III establishes the NMs model integrated by both physics-based and data-driven subsystems. Section IV devises the conformance-empowered reachability analysis for the PDI-NMs dynamics. Section V presents case studies on a typical NMs system to validate the *Neuro-Reachability* method. Finally, Section VI concludes the paper.

II. ODE-NET-ENABLED DYNAMIC MODEL DISCOVERY FOR MICROGRIDS

Pursuant to the attainablility of physics models, the overall NMs system can always be partitioned into an internal subsystem (*InSys*) and an external subsystem (*ExSys*), as illustrated in Fig. 1. *InSys*, where the structure and parameters are precisely known, can be readily formulated by assembling the dynamic models of its components. *ExSys*, in contrast, has to be modeled via a data-driven approach due to the absence of physics models, the unavailability of state measurements, and/or the need to preserve the consumer privacy.

This section devises an ODE-Net-enabled method to discover a state-space model of *ExSys*.

A. ODE-Net-Based State-Space Model Formulation

The functional formulation of *ExSys* is established as

$$\dot{\boldsymbol{x}} = \mathcal{N}(\boldsymbol{x}, \boldsymbol{u}) \tag{1}$$

Here, function $\mathcal{N}$ represents a state-space form of *ExSys* and is to be learned from the measurements; $\boldsymbol{x}$ and $\boldsymbol{u}$ respectively denote the state variables and input variables of *ExSys*. This ODE-governed, learned *ExSys* model can be integrated with the *InSys* model for assessing the overall NMs dynamics. Details of $\boldsymbol{x}$ and $\boldsymbol{u}$ are introduced in Subsection III-B. The rationale behind an ODE-governed *ExSys* model is provided in Appendix C.

Fig. 2 illustrates the dynamic model discovery using ODE-Net. Taking $\boldsymbol{x}$ and $\boldsymbol{u}$ at time t as the inputs, ODE-Net outputs the time derivative of $\boldsymbol{x}$ and therefore explicitly establishes the *ExSys* dynamic model in (1) by the forward propagation in the neural network.

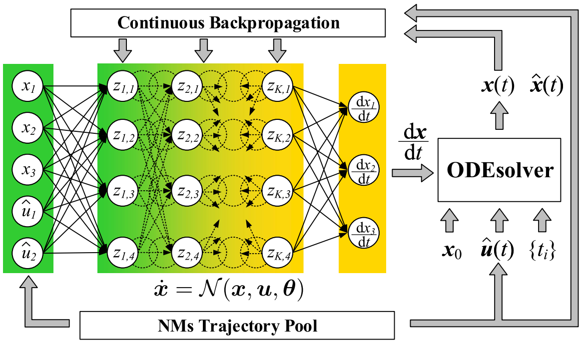


Fig. 2. ODE-Net-enabled dynamic model discovery of *ExSys*.

Given a time series of NMs trajectories as $\{(t_1, t_2, \ldots, t_n), (\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_n), (\hat{\boldsymbol{u}}_1, \hat{\boldsymbol{u}}_2, \ldots, \hat{\boldsymbol{u}}_n)\}$, ODE-Net best matches the *ExSys* dynamics by minimizing the error between the state measurements $\hat{\boldsymbol{x}}$ and the numerical solution of (1):

$$\begin{aligned} &\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} L(\boldsymbol{x}_i) = \sum_{i=1}^{n} \frac{1}{2} \eta_i \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|_2 \\ &s.t. \boldsymbol{x}_i = \hat{\boldsymbol{x}}_1 + \int_{t_1}^{t_i} \mathcal{N}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta}) \mathrm{d}t \end{aligned} \tag{2}$$

where $\boldsymbol{\theta}$ denotes the ODE-Net parameters; η_i denotes the weighting factor at time point i.

B. Continuous Backpropagation Technique

The main difficulty in the optimization of (2) lies in the ODE integration operation in the constraints. In this subsection, the continuous propagation technique [25] is applied to handle the ODE integration in the ODE-Net training.

Lagrange multiplier $\boldsymbol{\lambda}$ is first introduced to (2) to remove the ODE constraints and build the following loss function:

$$\mathcal{L} = \sum_{i=1}^{n} L(\boldsymbol{x}_i) - \int_{t_0}^{t_n} \boldsymbol{\lambda}^T \left(\dot{\boldsymbol{x}} - \mathcal{N}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\theta})\right) \mathrm{d}t \tag{3}$$

Backpropagation computes the gradient of the loss function with respect to the ODE-Net parameters to minimize the loss function [26]. With the loss function (3) involving the integration operator, the partial derivative of $\mathcal{L}$ with respect to $\boldsymbol{\theta}$ is calculated as:

$$\begin{aligned} &\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \\ &= \sum_{i=1}^{n} \left(\frac{\partial L}{\partial \boldsymbol{x}_i} \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{\theta}} - \int_{t_{i-1}}^{t_i} \boldsymbol{\lambda}^T \left(\frac{\partial \dot{\boldsymbol{x}}}{\partial \boldsymbol{\theta}} - \frac{\partial \mathcal{N}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} - \frac{\partial \mathcal{N}}{\partial \boldsymbol{\theta}} \right) \mathrm{d}t \right) \\ &= \sum_{i=1}^{n} \frac{\partial L}{\partial \boldsymbol{x}_i} \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{\theta}} + \sum_{i=1}^{n} \left(\boldsymbol{\lambda}^T(t_i^-) \frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{\theta}} - \boldsymbol{\lambda}^T(t_{i-1}^+) \frac{\partial \boldsymbol{x}_{i-1}}{\partial \boldsymbol{\theta}} \right) \\ &\quad + \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} \left(\frac{\mathrm{d}\boldsymbol{\lambda}^T}{\mathrm{d}t} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} + \boldsymbol{\lambda}^T \frac{\partial \mathcal{N}}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\theta}} + \boldsymbol{\lambda}^T \frac{\partial \mathcal{N}}{\partial \boldsymbol{\theta}} \right) \mathrm{d}t \end{aligned} \tag{4}$$

The Lagrange multiplier variables are given by [27]:

$$\frac{\mathrm{d}\boldsymbol{\lambda}^T}{\mathrm{d}t} = -\boldsymbol{\lambda}^T \frac{\partial \mathcal{N}}{\partial \boldsymbol{x}} \tag{5}$$

where the boundary conditions are set as:

$$\boldsymbol{\lambda}^T(t_n^+) = 0, \boldsymbol{\lambda}^T(t_i^+) = \boldsymbol{\lambda}^T(t_i^-) + \frac{\partial *}{\partial L}\boldsymbol{x}_i \tag{6}$$

Then, (4) can be derived into:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \int_{t_0}^{t_n} \boldsymbol{\lambda}^T \frac{\partial \mathcal{N}}{\partial \boldsymbol{\theta}} \mathrm{d}t \tag{7}$$

Collecting (5) and (7) leads to an ODE integration problem:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix} \boldsymbol{\lambda}^T \\ \frac{\partial *}{\partial \mathcal{L}}\boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\lambda}^T \frac{\partial *}{\partial \mathcal{N}}\boldsymbol{x} \\ \boldsymbol{\lambda}^T \frac{\partial *}{\partial \mathcal{N}}\boldsymbol{\theta} \end{bmatrix} \tag{8}$$

Subsequently, $\frac{\partial *}{\partial \mathcal{L}}\boldsymbol{\theta}$ can be obtained from (8) by any ODE solver, e.g., Trapezoidal integration. Given the final value of $\lambda(t)$, i.e., $\boldsymbol{\lambda}^T(t_n)$ in (6), rather than the initial value, (8) requires solving the ODEs backwards in time, which leads to a reverse-mode integration [25]:

$$\left.\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}\right|_{t_1} = \left.\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}}\right|_{t_n} + \int_{t_n}^{t_1} \boldsymbol{\lambda}^T \frac{\partial \mathcal{N}}{\partial \boldsymbol{\theta}} \mathrm{d}t = \sum_{i=2}^{n} \int_{t_i}^{t_{i-1}} \boldsymbol{\lambda}^T \frac{\partial \mathcal{N}}{\partial \boldsymbol{\theta}} \mathrm{d}t \tag{9}$$

with $\boldsymbol{\lambda}(t)$ also solved by the reverse-mode integration:

$$\boldsymbol{\lambda}^T(t_{i-1}^+) = \boldsymbol{\lambda}^T(t_i^-) + \int_{t_i}^{t_{i-1}} \boldsymbol{\lambda}^T \frac{\partial \mathcal{N}}{\partial \boldsymbol{x}} \mathrm{d}t \tag{10}$$

Further, consider *a set of time series* of NMs trajectories as $\{(\boldsymbol{t}^{(1)}, \hat{\boldsymbol{x}}^{(1)}, \hat{\boldsymbol{u}}^{(1)}), \ldots, (\boldsymbol{t}^{(m)}, \hat{\boldsymbol{x}}^{(m)}, \hat{\boldsymbol{u}}^{(m)})\}$. For the j^{th} measurement, let $\frac{\partial *}{\partial \mathcal{L}^{(j)}}\boldsymbol{\theta}$ be the gradient of the loss function computed by (9). The overall gradient is obtained:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} = \sum\nolimits_{j=1}^{m} \frac{\partial \mathcal{L}^{(j)}}{\partial \boldsymbol{\theta}} \tag{11}$$

Consequently, the ODE-Net parameters are updated using gradient descent so that $\mathcal{L}$ can be decreased during training:

$$\boldsymbol{\theta} \longleftarrow \boldsymbol{\theta} - r\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}} \tag{12}$$

where r denotes the learning rate.

The continuous backpropagation incorporates the "ODE solver" in the gradient descent for the ODE-Net parameter optimization, and hence effectively retains the intrinsic continuous differential structure of the dynamical NMs.

C. Neural Network Structure Design

The keystone of ODE-Net is to model the continuous-time nonlinear dynamics via the continuous backpropagation. As for realization, ODE-Net can be built upon any well-designed DNN architectures (e.g. multi-layer perceptron, residual neural network). However, a deep ODE-Net involves a large number of parameters and prohibitively high cost for computing $\frac{\partial *}{\partial \mathcal{N}}\boldsymbol{x}$ and $\frac{\partial *}{\partial \mathcal{N}}\boldsymbol{\theta}$ in the continuous backpropagation. To resolve this problem, [25] devises a *continuous-depth* network, which enables a linear memory cost with network depth and controllable numerical error.

In a classical neural network, the hidden layers follow a discrete structure (i.e., layers 1, 2, $\cdots$, K as illustrated by the solid-line neurons in Fig. 2), and therefore formulates a set of difference equations for forward propagation:

$$\boldsymbol{z}_{k+1} = \boldsymbol{z}_k + \boldsymbol{h}(\boldsymbol{z}_k, \boldsymbol{\theta}_k) \tag{13}$$

Here, $k \in \mathbb{N}^+$ denotes the discrete layer; $\boldsymbol{z}_k$ and $\boldsymbol{\theta}_k$ respectively denote the output states and parameters of the k^{th} hidden layer.

The continuous-depth network regards the forward propagation of the discrete layers in (13) as an Euler discretization of a set of continuous differential equations, i.e., it continuously propagates the states from the input layer to the output layer (see the dotted-line neurons in Fig. 2). This idea leads to a continuous "layer dynamics":

$$\frac{\mathrm{d}\boldsymbol{z}(k)}{\mathrm{d}k} = \boldsymbol{h}(\boldsymbol{z}(k), \boldsymbol{\theta}_h) \Longrightarrow \boldsymbol{z}_K = \boldsymbol{z}_1 + \int_1^K \boldsymbol{h}(\boldsymbol{z}, \boldsymbol{\theta}_h)\mathrm{d}k \tag{14}$$

Here, $k \in \mathbb{R}^+$ denotes the continuous layer; $\boldsymbol{z}(k)$ and $\boldsymbol{\theta}_h$ respectively denote the output states and parameters of the continuous hidden layer. As can be seen from (14), $\boldsymbol{z}_K$, i.e., the final output of the hidden layer, can be directly obtained by integrating over the continuous layers.

Based on the continuous-depth network, the chain rule is applied to compute $\frac{\partial *}{\partial \mathcal{N}}\boldsymbol{x}$ and $\frac{\partial *}{\partial \mathcal{N}}\boldsymbol{x}$ for (8):

$$\frac{\partial \mathcal{N}}{\partial \boldsymbol{x}} = \boldsymbol{\theta}_K^T \frac{\partial \boldsymbol{z}_K}{\partial \boldsymbol{z}_1}\frac{\partial \boldsymbol{z}_1}{\partial \boldsymbol{x}}, \frac{\partial \mathcal{N}}{\partial \boldsymbol{\theta}_h} = \boldsymbol{\theta}_K^T \frac{\partial \boldsymbol{z}_K}{\partial \boldsymbol{\theta}_h} \tag{15}$$

To obtain $\frac{\partial *}{\partial \boldsymbol{z}_K}\boldsymbol{z}_1$ and $\frac{\partial *}{\partial \boldsymbol{z}_K}\boldsymbol{\theta}_h$ in the equation above, the reverse-mode ODE integration discussed in Subsection II-B is again applied. Specifically, in analogy to (3), the following ODEs are formulated [25] for the gradient of $\boldsymbol{z}_K$ with respect to $\boldsymbol{z}(k)$:

$$\frac{\mathrm{d}}{\mathrm{d}k}\begin{bmatrix} \frac{\partial *}{\partial \boldsymbol{z}_K}\boldsymbol{z} \\ \frac{\partial *}{\partial \boldsymbol{z}_K}\boldsymbol{\theta}_h \end{bmatrix} = \begin{bmatrix} -(\frac{\partial *}{\partial \boldsymbol{z}_K}\boldsymbol{z})^T \frac{\partial *}{\partial \boldsymbol{h}}\boldsymbol{z} \\ (\frac{\partial *}{\partial \boldsymbol{z}_K}\boldsymbol{z})^T \frac{\partial *}{\partial \boldsymbol{h}}\boldsymbol{\theta}_h \end{bmatrix} \tag{16}$$

Then, in analogy to (11) and (12), the numerical solution of (16), i.e., $\frac{\partial *}{\partial \boldsymbol{z}_K}\boldsymbol{z}_1$ and $\frac{\partial *}{\partial \boldsymbol{z}_K}\boldsymbol{\theta}_h$, is obtained by the ODE integration.

III. PHYSICS-DATA-INTEGRATED ODE MODELING FOR NMS DYNAMICS

Based on the ODE-Net-enabled *ExSys* formulation, this section establishes the overall dynamic model of NMs by combining the physics-based *InSys* and data-driven *ExSys*, as illustrated in Fig. 1.

A. Physics-Based Formulation of InSys

Given the model of each component (e.g., a DER, a power load, a branch), *InSys* can be explicitly formulated as a system of differential algebraic equations (DAEs):

$$\dot{\boldsymbol{s}}^{in} = \boldsymbol{g}^{in}(\boldsymbol{s}^{in}, \boldsymbol{i}^{in}, \boldsymbol{w}^{in}, \boldsymbol{s}^{ex}) \tag{17a}$$

$$\dot{\boldsymbol{i}}^{in} = \boldsymbol{f}^{in}(\boldsymbol{s}^{in}, \boldsymbol{i}^{in}) + \boldsymbol{m}^{in}\boldsymbol{v} \tag{17b}$$

$$\boldsymbol{n}^{in}\boldsymbol{i}^{in} + \boldsymbol{n}^{ex}\boldsymbol{i}^{ex} = \boldsymbol{0} \tag{17c}$$

Here, (17a) formulates the dynamics of each component in *InSys*, where $\boldsymbol{s}^{in}$ denotes the state variables of each component; function $\boldsymbol{g}^{in}$ is formulated according to the dynamics of each component; $\boldsymbol{i}^{in}$ denotes the current injections from each component and is normally formulated under the DQ coordinates for the inverter-dominated NMs; $\boldsymbol{w}^{in}$ denotes the uncertain inputs caused by DERs in *InSys*; $\boldsymbol{s}^{ex}$ represents the global control signals sent by *ExSys*, i.e., $\boldsymbol{\Omega}$ and $\boldsymbol{e}$ as detailed in Appendix. B. Equation (17b) formulates the current injections from each component, where $\boldsymbol{v}$ denotes the DQ-axis voltages at each bus of *InSys* and boundary buses of *ExSys*; $\boldsymbol{m}^{in}$ denotes the DQ-axis incidence matrix between components and buses. Equation (17c) formulates the Kirchhoff's Current Law at the buses of *InSys* and boundary buses of *ExSys*, where $\boldsymbol{i}^{ex}$ denotes the current injections from *ExSys*; matrices $\boldsymbol{n}^{in}$ and $\boldsymbol{n}^{ex}$ denotes the directed component-bus incidence matrix.

Since *InSys* represents the identified subsystem in NMs, all the state variables of *InSys* can be acquired by either measurements or dynamic state estimation.

B. Data-Driven Formulation of ExSys

Following (17), *ExSys* interacts with *InSys* through its current injections $\boldsymbol{i}_{ex}$ and global control signals $\boldsymbol{s}_{ex}$. Thus, $\boldsymbol{i}_{ex}$ and $\boldsymbol{s}_{ex}$ are retained for dynamic simulations of NMs and are supposed to be measurable, which represent the fundamental electrical- and cyber- interactions between *ExSys* and *InSys*.

Accordingly, *ExSys* is functionally formulated as:

$$\dot{\boldsymbol{s}}^{ex} = \boldsymbol{g}^{ex}(\boldsymbol{i}^{in}, \boldsymbol{s}^{in}, \boldsymbol{i}^{ex}, \boldsymbol{s}^{ex}, \boldsymbol{w}^{in}, \boldsymbol{w}^{ex}) \tag{18a}$$

$$\dot{\boldsymbol{i}}^{ex} = \boldsymbol{f}^{ex}(\boldsymbol{i}^{in}, \boldsymbol{s}^{in}, \boldsymbol{i}^{ex}, \boldsymbol{s}^{ex}, \boldsymbol{w}^{in}, \boldsymbol{w}^{ex}) \tag{18b}$$

Here, (18a) formulates the dynamics of control variables sending from *ExSys* to *InSys*; (18b) formulates the dynamics of the current injections from *ExSys*; $\boldsymbol{w}^{ex}$ denotes the uncertain factors in *ExSys*. Note that (18) can readily incorporate additional measurable and shareable states of *ExSys*.

ODE-Net is used to establish the ODE model in (18), following the procedures in Section II. The *ExSys* formulation in (1) can now be expanded by: $\mathcal{N} = [\boldsymbol{g}^{ex}; \boldsymbol{f}^{ex}]$ denoting the dynamics of *ExSys*; $\boldsymbol{x} = [\boldsymbol{s}^{ex}; \boldsymbol{i}^{ex}]$ denoting the state variables of *ExSys*; and $\boldsymbol{u} = [\boldsymbol{i}^{in}; \boldsymbol{s}^{in}; \boldsymbol{w}^{in}; \boldsymbol{w}^{ex}]$ assembling the input variables in (18).

C. Physics-Data-Integrated (PDI) NMs Model

The entire NMs model is established by combining the physics-based formulation of *InSys* and data-driven formulation of *ExSys*:

$$\begin{cases} \dot{\boldsymbol{s}} = \boldsymbol{g}(\boldsymbol{i}, \boldsymbol{s}, \boldsymbol{w}) & \text{(19a)} \\ \dot{\boldsymbol{i}} = \boldsymbol{f}(\boldsymbol{i}, \boldsymbol{s}, \boldsymbol{w}) + \boldsymbol{m}\boldsymbol{v} & \text{(19b)} \\ \boldsymbol{0} = \boldsymbol{n}\boldsymbol{i} & \text{(19c)} \end{cases}$$

where $\boldsymbol{s}$, $\boldsymbol{i}$ and $\boldsymbol{w}$ respectively assemble the state variables, current injections and uncertainties of *InSys* and *ExSys*.

The DAE model in (19) can be rigorously converted to a system of nonlinear ODEs [28], as briefly introduced below. Denote the matrix constructed by the maximal linearly independent columns of $\boldsymbol{n}$ as $\boldsymbol{n}_1$, and the matrix constructed by the other columns as $\boldsymbol{n}_0$. Since $\boldsymbol{n}_1$ is non-singular, (19c) leads to the following:

$$\boldsymbol{n}_0\boldsymbol{i}_0 + \boldsymbol{n}_1\boldsymbol{i}_1 = \boldsymbol{0} \Longrightarrow \boldsymbol{i}_1 = -\boldsymbol{n}_1^{-1}\boldsymbol{n}_0\boldsymbol{i}_0 \tag{20}$$

where $\boldsymbol{i}_0$ and $\boldsymbol{i}_1$ respectively denote the sub-vectors of $\boldsymbol{i}$ corresponding to $\boldsymbol{n}_0$ and $\boldsymbol{n}_1$.

Taking derivative of (19c) yields the following:

$$\boldsymbol{0} = \boldsymbol{n}\dot{\boldsymbol{i}} = \boldsymbol{n}\hat{\boldsymbol{f}}(\boldsymbol{i}_0, \boldsymbol{s}) + \boldsymbol{n}\boldsymbol{m}\boldsymbol{v} \Longrightarrow \boldsymbol{v} = -(\boldsymbol{n}\boldsymbol{m})^{-1}\boldsymbol{n}\hat{\boldsymbol{f}} \tag{21}$$

where $\hat{\boldsymbol{f}}(\boldsymbol{i}_0, \boldsymbol{s}, \boldsymbol{w}) = \boldsymbol{f}(\boldsymbol{i}, \boldsymbol{s}, \boldsymbol{w})$ by substituting (20) to $\boldsymbol{f}$.

Therefore, (19) is converted to an ODE model:

$$\begin{cases} \dot{\boldsymbol{s}} = \hat{\boldsymbol{g}}(\boldsymbol{i}_0, \boldsymbol{s}, \boldsymbol{w}) & \text{(22a)} \\ \dot{\boldsymbol{i}}_0 = -\boldsymbol{m}_0(\boldsymbol{n}\boldsymbol{m})^{-1}\boldsymbol{n}\hat{\boldsymbol{f}}(\boldsymbol{i}_0, \boldsymbol{s}) + \hat{\boldsymbol{f}}_0(\boldsymbol{i}_0, \boldsymbol{s}) & \text{(22b)} \end{cases}$$

where $\boldsymbol{m}_0$ and $\hat{\boldsymbol{f}}_0$ respectively extract the components of $\boldsymbol{m}$ and $\hat{\boldsymbol{f}}$ corresponding to $\boldsymbol{i}_0$.

The obtained ODE model in (22) is rigorously equivalent to the original DAE model (19a) without adopting any linearization. Hence, it can be used for the transient analysis under small or large disturbances. The PDI-NMs model in (22) can then be abstracted as:

$$\dot{\boldsymbol{X}} = \boldsymbol{F}(\boldsymbol{X}, \boldsymbol{W}) \tag{23}$$

where $\boldsymbol{X}$ denotes the state variables of NMs integrating states of both *InSys* and *ExSys*; $\boldsymbol{W}$ denotes to the uncertainty inputs.

IV. Neuro-Reachability Analysis of NMs Dynamics

Based on the PDI-NMs model, this section devises a *Neuro-Reachability* method for dynamic verification of NMs.

A. Reachset for NMs

Reachability analysis verifies the NMs dynamics by calculating reachsets, i.e., a provable enclosure of all possible dynamic trajectories under infinitely many uncertain scenarios. Given the set of initial NMs states $\mathcal{X}^0$ and the set of DER uncertainties $\mathcal{W}$, the time-point reachset is defined as the set of all the possible NMs states at time t:

$$\mathcal{R}(t) = \Big\{\boldsymbol{X}(t) = \int_0^t \boldsymbol{F}(\boldsymbol{X}(\tau), \boldsymbol{W}(\tau))\mathrm{d}\tau \;\Big|\; \boldsymbol{X}(0) \in \mathcal{X}^0, \boldsymbol{W} \in \mathcal{W}\Big\} \tag{24}$$

In this research, the uncertainty set $\mathcal{W}$ is formulated by an unknown-but-bounded set; and zonotope bundle is utilized for reachset formulation [29], i.e., an intersection of finite zonotopes which is able to represent arbitrary polytopes with a satisfactory computational efficiency.

Applying Taylor expansion to (23) at $(\boldsymbol{X}^*, \boldsymbol{W}^*)$ gives:

$$\dot{\boldsymbol{X}} \in \boldsymbol{F}^* + \boldsymbol{J}(\boldsymbol{X} - \boldsymbol{X}^*) + \boldsymbol{J}_w(\boldsymbol{W} - \boldsymbol{W}^*) + \boldsymbol{e}_l \tag{25}$$

where $\boldsymbol{F}^* = \boldsymbol{F}(\boldsymbol{X}^*, \boldsymbol{W}^*)$; $\boldsymbol{J} = \frac{\partial *}{\partial \boldsymbol{F}^*}\boldsymbol{X}$ and $\boldsymbol{J}_w = \frac{\partial *}{\partial \boldsymbol{F}^*}\boldsymbol{W}$ respectively denote the Jacobian matrix referring to the first-order Taylor series; $\boldsymbol{e}_l$ is the Lagrange remainder [30]. For computational convenience, $\mathcal{R}(t)$ is shifted to $\mathcal{R}(t) - \boldsymbol{X}^*$ to perform the set computation. Therefore, the time-point reachset of the NMs dynamics can be computed by the evolution of the previous time-point reachset:

$$\mathcal{R}(t) = \left(e^{\boldsymbol{J}\Delta}\mathcal{R}(t-\Delta)\right) \oplus \int_0^{\Delta} e^{\boldsymbol{J}(\Delta-\tau)}\mathcal{W}_F \mathrm{d}\tau \oplus \mathcal{R}_{el} \quad (26)$$

where Δ denotes the time step; $\oplus$ denotes the Minkowski addition. In (26), the first term computes the reachset propagation resulting from the NMs states, i.e., $\boldsymbol{J}(\boldsymbol{X}-\boldsymbol{X}^*)$. The second term computes the reachset propagation resulting from the inputs $\mathcal{W}_F = \boldsymbol{J}_w(\mathcal{W}-\boldsymbol{W}^*)+\boldsymbol{F}^*$, which composes both the uncertainty impact $\boldsymbol{J}_w(\boldsymbol{W}-\boldsymbol{W}^*)$ and the linearization point impact $\boldsymbol{F}^*$. And the third term in (26) computes the set of linearization error [30] to ensure an provable over-approximation of the nonlinear NMs dynamics.

Further, the reachset during time interval $[t-\Delta, t]$ can be calculated as the union of the time-point reachable sets during the interval as $\mathcal{R}([t-\Delta,t]) = \cup_{\tau\in[t-\Delta,t]}\mathcal{R}(\tau)$.

B. Conformance Reachset for PDI-NMs

The ODE-Net-enabled model learned from a finite set of training samples, although sufficiently precise for the training set, would not perfectly replicate the real dynamics of the system under any circumstances. To address the possible inaccuracy of the PDI-NMs model, this subsection empowers the reachability analysis done in Subsection IV-A with the conformance theory as a 'feedback' mechanism to further improve the *Neuro-Reachability* reliability.

1) Conformance-Empowered Reachset Formulation: Incorporating the model inaccuracy into the Taylor expansion of NMs dynamics leads to the following:

$$\dot{\boldsymbol{X}} \in \boldsymbol{F}^* + \boldsymbol{J}(\boldsymbol{X}-\boldsymbol{X}^*) + \boldsymbol{J}_w(\boldsymbol{W}-\boldsymbol{W}^*) + \boldsymbol{e}_l + \boldsymbol{e}_m \quad (27)$$

Compared with (25), an additional term $\boldsymbol{e}_m$ is introduced to address the impact of the discrepancy between the PDI-NMs model and the real NMs trajectories. In this research, the PDI-NMs model inaccuracy is formulated as multidimensional intervals $\mathcal{E}_m = [\underline{\boldsymbol{e}}_m, \overline{\boldsymbol{e}}_m]$, where $\underline{\boldsymbol{e}}_m$ and $\overline{\boldsymbol{e}}_m$ respectively denotes the infimum and supremum of the PDI-NMs model inaccuracy. Correspondingly, the reachset computation can be modified as:

$$\begin{aligned}\mathcal{R}_c(t) = &\left(e^{\boldsymbol{J}\Delta}\mathcal{R}_c(t-\Delta)\right) \oplus \mathcal{R}_{el} \oplus \\ &\int_0^{\Delta} e^{\boldsymbol{J}(\Delta-\tau)}(\mathcal{W}_F + \mathcal{E}_m)\mathrm{d}\tau\end{aligned} \quad (28)$$

Here, $\mathcal{R}_c$ denotes a conformance reachset incorporating the PDI-NMs model error $\mathcal{E}_m$.

Given a specific $\mathcal{E}_m$, $\mathcal{R}_c$ can be readily computed following Subsection IV-A. However, theoretically, $\mathcal{E}_m$ is a posteriori error depending on specific NMs states and DER uncertainties, and is supposed to be computed by comparing the PDI-NMs dynamics with the real NMs trajectories. To tackle this difficulty, an optimization approach to estimating $\mathcal{E}_m$ is introduced by constructing a minimal-volume conformance reachset while ensuring a provable enclosure of the time-series measurements of the real NMs dynamics.

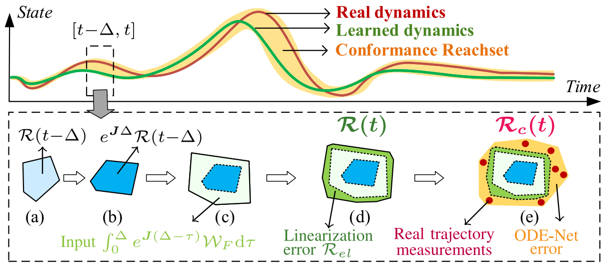


Fig. 3. Illustration of conformance reachset.

2) $\mathcal{E}_m$-Optimization Model: Given a time series of NMs trajectories $\{\boldsymbol{t}, \hat{\boldsymbol{X}}, \hat{\boldsymbol{W}}\} = \{(t_1, t_2, \ldots, t_n), (\hat{\boldsymbol{X}}_1, \hat{\boldsymbol{X}}_2, \ldots, \hat{\boldsymbol{X}}_n), (\hat{\boldsymbol{W}}_1, \hat{\boldsymbol{W}}_2, \ldots, \hat{\boldsymbol{W}}_n)\}$, $\mathcal{R}(t)$ can be computed following Subsection IV-A with the NMs initial state set as $\mathcal{X}^0 = \{\hat{\boldsymbol{X}}_1\}$ and the DER uncertainty set as $\mathcal{W} = \{\hat{\boldsymbol{W}}\}$. As discussed above, since the PDI-NMs model may not perfectly conform with the real NMs dynamics, the reachset $\mathcal{R}(t)$ from (26), computed with $\mathcal{W}$ and $\mathcal{X}^0$, possibly does not enclose $\hat{\boldsymbol{X}}$. Inspired by the conformance theory in [31], [32], $\mathcal{E}_m$ is optimized to ensure that the conformance reachset $\mathcal{R}_c(t)$ encloses the time-series NMs states $\hat{\boldsymbol{X}}$:

$$\begin{aligned}&\mathcal{E}_m\textbf{-opt model}\text{: } \min_{\mathcal{E}_m} \sum\nolimits_{t\in\boldsymbol{t}} \mathrm{Vol}(\mathcal{R}_c(t)) \\ &\qquad s.t. \hat{\boldsymbol{X}}(t) \in \mathcal{R}_c(t), \forall t \in \boldsymbol{t}\end{aligned} \quad (29)$$

Here, $\mathcal{R}_c$ is computed as (28) with aforementioned $\mathcal{X}^0$ and $\mathcal{W}$; $\mathrm{Vol}(\cdot)$ computes the volume of the reachset. Optimization in (29) solves an $\mathcal{E}_m$ such that $\mathcal{R}_c$ encloses $\hat{\boldsymbol{X}}$ with a minimal modification on the reachset. As a special case, if the PDI-NMs model perfectly replicates the NMs trajectories $\hat{\boldsymbol{X}}$, (29) gives $\mathcal{E}_m = \emptyset$.

Fig. 3 illustrates the basic idea of this optimization-based conformance reachset method. For better visualization, the trajectories generated by the PDI-NMs model (denoted by the green line) differs largely from the real NMs trajectories (denoted by the red line), which further leads to a quite loose conformance reachset (denoted by the yellow area). Fortunately, case studies in Section V will show that the ODE-Net-based dynamic model is quite accurate and the obtained neuro-reachsets is rather tight. Taking time period $[t-\Delta, t]$ as an example, following (26), the conventional reachset $\mathcal{R}(t)$ (Fig. 3(d)) is computed incorporating the evolution of NMs states (Fig. 3(b)), DER uncertainty inputs (Fig. 3(c)) and linearization error (Fig. 3(d)). Then, as illustrated in Fig. 3(e), the conformance reachset $\mathcal{R}_c(t)$, which amends $\mathcal{R}(t)$ with the impact of the model inaccuracy set $\mathcal{E}_m$, will enclose the real NMs states at time t with a minimized set volume. The above process is successively computed over the time horizon to optimize a $\mathcal{E}_m$ such that $\mathcal{R}_c$ encloses the real NMs trajectories.

The $\mathcal{E}_m$-opt model is intractable to optimize due to the complicated set calculations in the objective and constraints. For efficient volume computation, $\mathcal{R}_c$ is over-approximated by hyperrectangles. Denote $\mathcal{D} = (e^{\boldsymbol{J}\Delta}\mathcal{R}^c(t-\Delta)) \oplus \int_0^{\Delta} e^{\boldsymbol{J}(\Delta-\tau)}\mathcal{W}_F \mathrm{d}\tau \oplus \mathcal{R}_{el}$, which can be readily computed as the conventional reachset by (26). Denote $\boldsymbol{E} =$

$\int_0^\Delta e^{\boldsymbol{J}(\Delta-\tau)}\mathrm{d}\tau$. Accordingly, the volume of the conformance reachset is over-approximated as:

$$\begin{aligned} \mathrm{Vol}(\mathcal{R}_c(t)) &= \mathrm{Vol}(\mathcal{D} \oplus \boldsymbol{E}\mathcal{E}_m) \\ &\subseteq \mathrm{Vol}(\mathrm{box}(\mathcal{D}) \oplus \boldsymbol{E}[\underline{\boldsymbol{e}}_m, \overline{\boldsymbol{e}}_m)]) \triangleq \mathrm{Vol}(\overline{\mathcal{R}}_c(t)) \end{aligned} \tag{30}$$

Consequently, the objective of $\mathcal{E}_m$-opt, i.e., minimizing the conformance reachset volume, is simplified into minimizing the summation of edge lengths of the hyperrectangle $\overline{\mathcal{R}}_c(t)$. The new objective becomes:

$$\min_{\overline{\boldsymbol{e}}_m, \underline{\boldsymbol{e}}_m} \mathbf{1}^T (\sup(D) - \inf(D) + |\boldsymbol{E}|(\overline{\boldsymbol{e}}_m - \underline{\boldsymbol{e}}_m)) \tag{31}$$

On the other hand, the constraints of $\mathcal{E}_m$-opt require that the NMs trajectories are enclosed by the conformance reachset $\mathcal{R}_c$, which are also handled by hyperrectangles:

$$\begin{cases} \hat{\boldsymbol{X}}(t) \leq \sup(\mathcal{D} \oplus \boldsymbol{E}\mathcal{E}_m) = \sup(\mathcal{D}) + |\boldsymbol{E}|\overline{\boldsymbol{e}}_m \\ \hat{\boldsymbol{X}}(t) \geq \inf(\mathcal{D} \oplus \boldsymbol{E}\mathcal{E}_m) = \inf(\mathcal{D}) + |\boldsymbol{E}|\underline{\boldsymbol{e}}_m \end{cases} \tag{32}$$

Combining (31) and (32) leads to a reformulated $\mathcal{E}_m$-opt model in the form of linear programming that can be tractably solved.

Further, with *a set of time series* of NMs trajectories, $\mathcal{E}_m$-opt results from each trajectory are joint to obtain the overall $\mathcal{E}_m$:

$$\mathcal{E}_m = \bigcup\nolimits_{j=1}^{m} \mathcal{E}_m^{(j)} \tag{33}$$

where $\mathcal{E}_m^{(j)}$ denotes the optimization result from (31) and (32) for the j^{th} measurements.

V. CASE STUDIES

This section demonstrates the technical merit and efficacy of the *Neuro-Reachability* method for NMs. The algorithm is implemented in MATLAB R2019b.

A. Case Design

Case studies are conducted on the 4-microgrid NMs in Fig. 1, with the DER controller parameters modified from [33]. The NMs parameters are provided in Appendix. A. The DERs can be equipped with droop controllers and/or secondary controllers. Five cases are designed to verify the *Neuro-Reachability* method. Each case shares the same settings of NMs topology and load condition, with different control strategies of DERs as well as power source mixes.

Case 1: All the DERs are equipped with droop control; microgrid 4 is supposed to be model-free and data-driven, while other microgrids are physics-based with formulations detailed in Appendix. B;

Case 2: All the DERs, both in *InSys* and *ExSys*, are equipped with hierarchical control (i.e., both droop and secondary controls) under an all-to-all communication among DERs, and microgrid 4 is data-driven similar as Case 1;

Case 3: All settings are the same with Case 1, except that microgrid 3 (which comprises 2 DERs) is data-driven while microgrids 1, 2, 4 are physics-based;

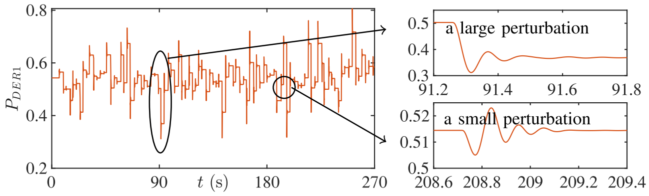
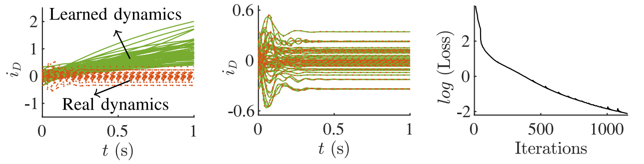


Fig. 4. Illustration of training set: time-series measurements of NMs dynamics under perturbations.



(a) ODE-Net performance at the starting stage (b) ODE-Net performance at the final stage (c) Loss function evolution process

Fig. 5. Training process of ODE-Net for NMs dynamic model discovery.

Case 4: All settings are the same with Case 3, except that the DER at bus 6 in microgrid 3 is replaced by a synchronous generator (SG);

Case 5: All settings are the same with Case 3, except that the DER at bus 6 in microgrid 3 is replaced by an energy storage unit (ESU);

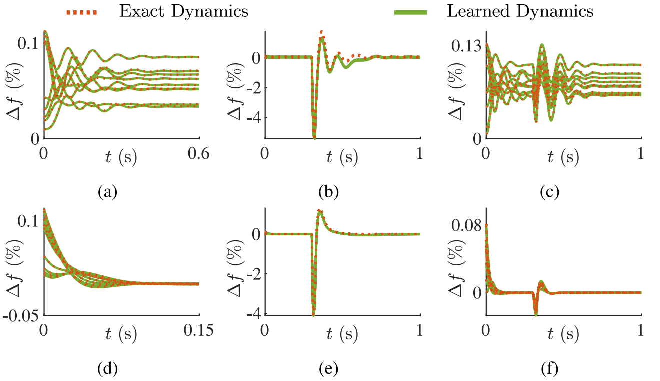
Uncertainty settings: By default, the uncertainty of each DER is set as 20%, which means that the reference active power of DER controllers can randomly deviate from the prediction to address the impact of uncertain renewables. Mathematically, uncertainty is modelled as an unknown-but-bounded set [29] in reachability analysis.

Dataset settings: NMs undergo frequent transients due to the fluctuating of DERs. Thereby, measurements are acquired under uncertain perturbations to construct the training set for ODE-Net. In this paper, the training data contains time-series measurements lasting for 270 seconds. Fig. 4 shows the training data, taking a single dimension (i.e., power output from DER1) for illustration purpose.

B. Validity of ODE-Net-Based NMs Model Discovery

1) Performance of ODE-Net: This subsection demonstrates the performance of ODE-Net in learning the state-space model of microgrids. A two-layer perceptron architecture is used for ODE-Net, with 40 neurons in each layer. The trapezoidal rule is employed for NMs dynamic integration. The Adaptive Moment Estimation (Adam) [34] algorithm is applied to enable an adaptive learning rate during the ODE-Net training.

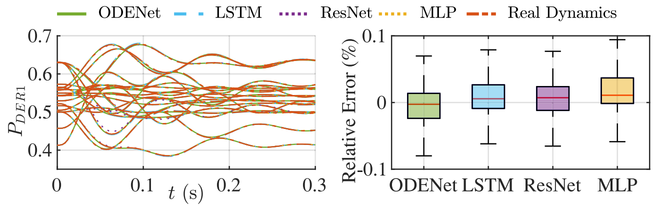
Fig. 5 presents the ODE-Net training process for Case 1. At the starting stage, ODE-Net is randomly initialized and largely deviates from the real NMs trajectories, as illustrated in Fig. 5(b). Then, after the neural network training via continuous backpropagation, ODE-Net converges to a perfect match of the NMs trajectories on the training set, as illustrated in Fig. 5(c). Additionally, Fig. 5(d) presents the evolution of loss function at the logarithmic scale. As a rule of thumb, 1500 iterations lead to convergence of the ODE-Net.



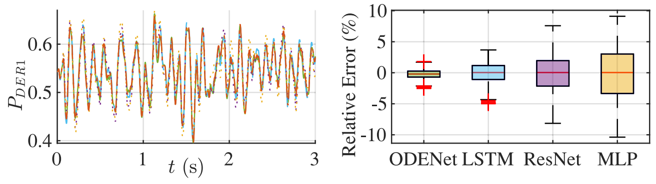
(a) (b) (c)

(d) (e) (f)

Fig. 6. ODE-Net performance on the test set of NMs dynamics (Δf: frequency deviation). (a)-(c), droop control under no fault, short-circuit fault and open-circuit fault respectively. (d)-(f), hierarchical control under no fault, short-circuit fault and open-circuit fault respectively.



(a) Comparison on training set (270s measurement)



(b) Comparison on test set (50s predictions)

Fig. 7. Comparison between ODE-Net and conventional DNN methods for NMs dynamic analysis.

Further, Fig. 6 illustrates the ODE-Net performance on the test set, which verifies its ability to generalize beyond the training set. Three types of scenarios are studied, i.e., no-fault, a short-circuit fault at bus 19, and an open-circuit fault of branch 2-3. Each fault occurs at 0.3 s and is cleared at 0.32 s. An interesting finding is that the ODE-Net-enabled NMs formulation accurately captures the uncertain NMs transients not only under the frequently fluctuating DER uncertainties, but also under large disturbances, although the latter scenarios never appear in the training set. This shows the robustness of the ODE-Net-enabled NMs formulation.

2) Comparison Between ODE-Net and Conventional DNN Techniques: To further illustrate the superiority of ODE-Net in modelling continuous-time dynamics of NMs, ODE-Net is compared with three representative DNN techniques, i.e., multi-layer perceptron (MLP), residual neural network (ResNet), and long short-term memory (LSTM). The configuration of each DNN is set as:

- An MLP comprised of 3 hidden layers with 200 hidden units in each layer;
- A ResNet comprised of 10 hidden layers with double-layer skips and 200 hidden units in each layer;
- An LSTM network with 200 hidden units.

Conventional DNN learns a nonlinear function between inputs and outputs. Therefore, discretization on differential equations in (1) is inevitable [35], [36]:

$$\frac{\boldsymbol{x}(t)-\boldsymbol{x}(t-\Delta)}{\Delta} = \frac{1}{2}(\mathcal{N}(\boldsymbol{x}(t),\boldsymbol{u}(t)) + \mathcal{N}(\boldsymbol{x}(t-\Delta),\boldsymbol{u}(t-\Delta))) \tag{34}$$

Here, we adopt the trapezoidal rule where Δ denotes the time step. Consequently, loss function for conventional DNN is constructed by using the distance between the real derivatives $\boldsymbol{y} = \frac{\boldsymbol{x}(t)-\boldsymbol{x}(t-\Delta)}{\Delta}$ and the trained derivatives $\hat{\boldsymbol{y}} = \frac{1}{2}(\mathcal{N}(\boldsymbol{x}(t),\boldsymbol{u}(t)) + \mathcal{N}(\boldsymbol{x}(t-\Delta)\boldsymbol{u}(t-\Delta)))$, :

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} L_{DNN} = \sum_{i=1}^{n} \frac{1}{2}\eta_i \|\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i\|_2 \tag{35}$$

Comparing (35) with (2), an obvious distinction is that ODE-Net is capable of directly minimizing the difference between real dynamic states and trained dynamic states.

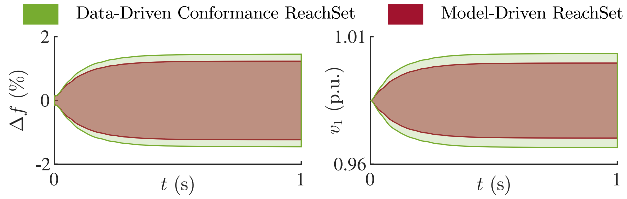
Fig. 7 presents the simulation results. Although both ODE-Net and DNN methods exhibit satisfactory accuracy on the training set (see Fig. 7(a) for time-domain trajectories and boxplot statistics), Fig. 7(b) clearly illustrates that their performance differs largely on the test set containing frequently perturbed NMs dynamics. Among the selected conventional DNN techniques, LSTM exhibits the best performance on the test set. Still, ODE-Net has a 50% higher precision than LSTM despite its simplest network architecture (i.e., a two-layer MLP). In addition, ODE-Net is able to take irregularly or sparsely sampled measurements since it directly handles the differential equations. In contrast, conventional DNN usually requires the measurements to be well aligned or synchronized to ensure a proper discreteziation.

Another advantage of ODE-Net is that with the discovered state-space model, various power system analytics (e,g., transient simulation, dynamic verification, stability analysis) can be conducted by regarding the ODE-Net-based *ExSys* as a special type of "dynamic component" and incorporating the data-driven model into the existing system model.

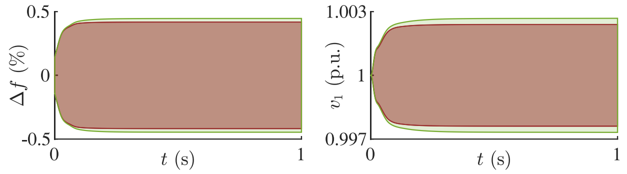
C. NMs Dynamic Verification via Neuro-Reachability

This subsection studies the NMs dynamics with both uncertain perturbations and fault disturbances via the *Neuro-Reachability* analysis.

Fig. 8 studies the neuro-reachsets under the quasi-static scenario, where the NMs is only perturbed by the DER uncertainties. The simulation shows that the neuro-reachsets tightly encloses the model-driven reachsets, which verifies both the accuracy and conservativeness of the method. In particular, in Case 2 where the DERs are equipped with both droop and secondary control, the *Neuro-Reachability* is supposed to learn not only the interactive currents between *ExSys* and *InSys*, but also the control signals from *ExSys*. Fig. 8(b) still shows a tight and perfect over-approximation of the real reachsets, which exhibits the potential of the *Neuro-Reachability* method in learning the

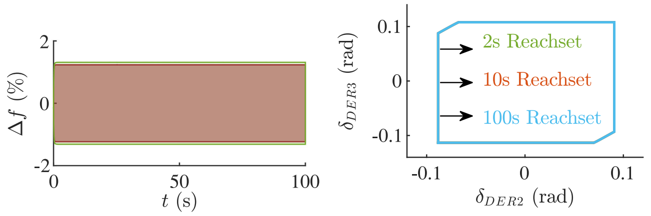


(a) Case 1: DERs with droop control

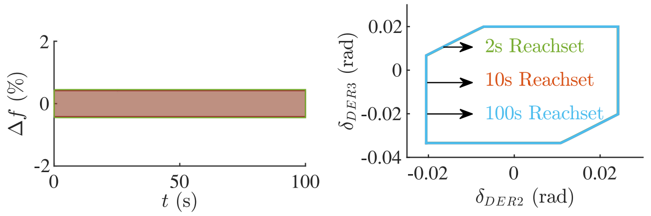


(b) Case 2: DERs with hierarchical control

Fig. 8. Neuro-reachsets and its comparison with the model-driven reachsets (Δf: frequency deviation; v_1: voltage amplitude at DER1).



(a) Case 1: DERs with droop control



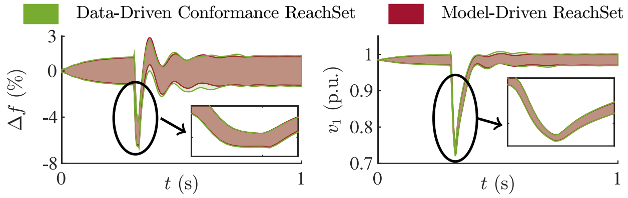
(b) Case 2: DERs with hierarchical control

Fig. 9. Illustration of numerical stability of neuro-reachsets.

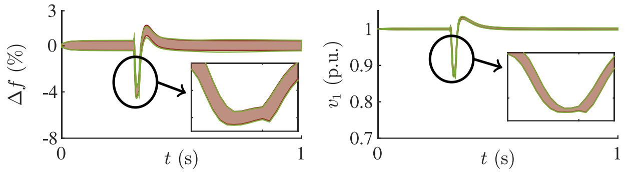
NMs dynamic model with complicated hierarchical control. *Neuro-Reachability* shows the superiority of hierarchical control from two aspects: i) reachsets in Case 2 are narrower than the those in Case 1, which reflects strengthened robustness of the NMs against uncertainties; ii) reachsets in Case 2 get stable faster than those in Case 1, which reflects a speedy power sharing between the DERs during the NMs dynamics induced by uncertain perturbations.

Fig. 9 further illustrates that for a duration of 100 s, neuro-reachsets remain numerically stable. Meanwhile, the 2D projection of reachsets at 2 s, 10 s and 100 s shows that reachsets do not diverge over the time. In fact, *Neuro-Reachability* inherits the numerical stability of conventional reachability methods [30], since the data-driven model error is formulated as a constant set $\mathcal{E}_m$ as presented in (28).

Further, Fig. 10 studies the neuro-reachsets under a momentary short-circuit fault occurred at 0.3 s and cleared at 0.32 s. Even though the ODE-Net-based NMs formulation is learned from the dynamics of small disturbances induced by the DERs' uncertainties, simulation results show that the *Neuro-Reachability* method accurately and tightly captures the fast NMs dynamics under heterogeneous uncertainties during disturbances. Comparing the neuro-reachsets in Case 1 and Case 2, it is obvious that the hierarchical control exhibits restrained frequency/voltage dips and enhanced damping on the oscillations of the NMs states. Specifically, the reachable sets during the periods from the fault occurrence through the fault clearance are magnified by the subplots. *Neuro-Reachability* thus effectively mimics the NMs transients initiated by large disturbances.

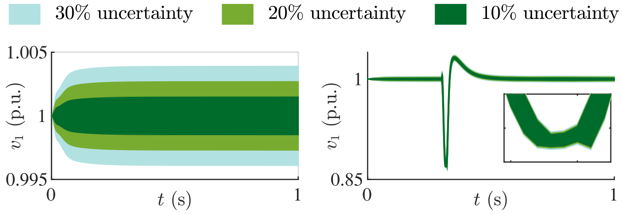


(a) Case 1: DERs with droop control



(b) Case 2: DERs with hierarchical control

Fig. 10. Neuro-reachsets under a short-circuit fault.



(a) No fault (b) Short-circuit fault

Fig. 11. Neuro-reachsets under different uncertainty levels of Case 2.

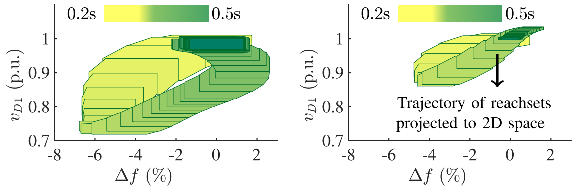
Neuro-Reachability inherits the efficiency and scalability of conventional reachability methods. Therefore, the worst time consumption is of polynomial complexity [30].

D. *Efficacy and Versatility of Neuro-Reachability*

This subsection studies the impact factors of the NMs reachsets via the *Neuro-Reachability* method.

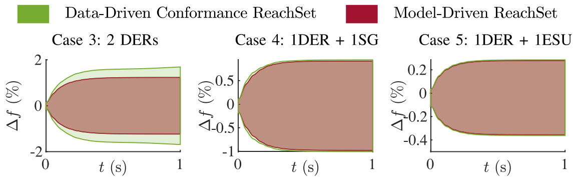
1) Neuro-Reachsets Under Different Uncertainty Levels: First, the impact of uncertainties on NMs dynamics is investigated. Taking DER1's output-voltage in Case 2 as an example, Fig. 11 shows that the reachsets expand with the increasing uncertainties under both no-fault and faulted scenarios. The propagation of uncertainties in the NMs dynamics is therefore distinctly demonstrated by the neuro-reachsets at different uncertainty levels.

2) Neuro-Reachsets Under Different Control Strategies: Second, the impact of the DERs' control strategies on NMs reachsets is investigated. Fig. 8 and Fig. 10 exhibit the efficacy of the hierarchical control to stabilize the system under uncertainties from the time-domain perspective. Further, Fig. 12 presents the neuro-reachsets from a state-space observation. The

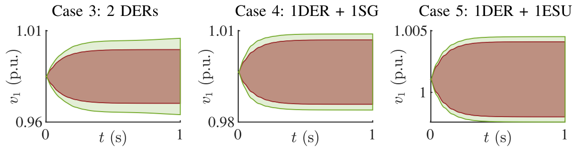


(a) Case 1: DERs with droop control (b) Case 2: DERs with hierarchical control

Fig. 12. State space neuro-reachsets under different control strategies (The colored zonotopes represent the trajectory of reachsets at different timesteps. Here, the high-dimensional reachsets are projected to 2-dimensional space for visualization purpose.). (a) Case 1: DERs with droop control. (b) Case 2: DERs with hierarchical control.



(a) Frequency reachset



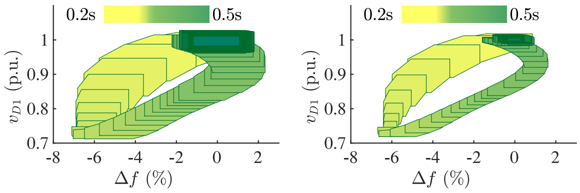
(b) Voltage reachset

Fig. 13. Neuro-reachsets under a short-circuit fault under different mixes of power sources in NMs.

efficacy of the hierarchical control for restraining the uncertainty impacts, damping the frequency/voltage overshoot and recovering the NMs states after faults are distinctly verified via the neuro-reachsets. Hence, *Neuro-Reachability* is promisingly helpful for verifying the controller's performance under the infinite many uncertain scenarios in the absence of the microgrid model.

3) Neuro-Reachsets Under Different Power Source Mixes: Finally, the impact of power source mixes is investigated via neuro-reachsets. Different power sources influence the NMs transients by their diverse dynamic features.

Fig. 13 studies the quasi-static reachsets (i.e., with no fault but only DER uncertainties perturbing the NMs) for Case 3, Case 4 and Case 5. The neuro-reachsets tightly encloses the model-driven reachsets in all cases, which again verifies the correctness of the *Neuro-Reachability* method. An interesting finding is that the neuro-reachsets of Case 3 in Fig. 13 are nearly identical to those of Case 1 in Fig. 8. This is because case 1 and Case 3 describe the identical NMs only with different microgrids being data-driven. Results of Case 4, compared with those of Case 3, show that the neuro-reachsets shrinks when the NMs are equipped with a SG, benefiting from the inertia and regulating ability of SG as well as its full dispachability compared with DERs. The neuro-reachsets is further improved when the NMs are equipped with an ESU as presented by the neuro-reachsets of Case 5, indicating enhanced robustness against the uncertainties with ESU. Meanwhile, it is noteworthy that both the SG and ESU provide voltage support to boost the NMs voltage, as illustrated in Fig. 13(b). Additionally, Fig. 14 investigates the neuro-reachsets of NMs during a short-circuit fault for Case 4 (i.e., equipped with a SG) and Case 5 (i.e., equipped with an ESU). The *Neuro-Reachability* method, as a data-driven approach, successfully captures the transient characteristics of the controllable power sources in restraining the uncertainties and damping the frequency/voltage dip/rise during the large disturbances compared with Fig. 12(a).



(a) Case 4 (b) Case 5

Fig. 14. Neuro-reachsets under different mixes of power sources in NMs.

VI. CONCLUSION

This paper devises a *Neuro-Reachability* method, a data-driven approach to verify the uncertainty-disturbed, fast-changing and strongly-nonlinear dynamics of the NMs with unidentified subsystems. The ODE-Net-enabled dynamic model discovery, reachability analysis, and conformance theory conjointly enable a flexible and accurate model discovery of real-world microgrids, as well as provide reliable reachsets for verifying the NMs dynamics under heterogeneous uncertainties. Case studies of a typical NMs demonstrate the efficacy and robustness of the devised method. In the future, the *Neuro-Reachability* method will be enhanced for the dynamic verification of fully model-free NMs.

APPENDIX A
TEST SYSTEM PARAMETERS

TABLE I
PARAMETERS OF DERS

DER ID.	1	2	3	4	5
Loc. node	1	6	13	25	33
Controller parameters [1]	m_p	n_q	r_o (p.u.)	L_o (p.u.)	F
	10%	30%	6.24×10^{-4}	3.12×10^{-3}	0.75
	k_{pv}	k_{iv}	k_{pc}	k_{ic}	
	2	20	105	1600	
Secondary control	α	A	β_l β_f [2]	B_l B_f	
	50	5	100 400	0 1	

[1] m_p: active power droop gain; n_q: reactive power droop gain; r_o: coupling resistance; L_o: coupling inductance; F: current feed-forward gain; k_{pv}: voltage proportional gain; k_{iv}: voltage integral gain; k_{pc}: current proportional gain; k_{ic}: current integral gain;

[2] β_l and β_f respectively represent secondary control parameters for leader DER and follower DERs [37]. In this paper, DER1 is chosen as the leader DER.

This appendix provides the parameters of the test system, i.e., the 4-microgrid NMs illustrated in Fig. 1.

TABLE II
PARAMETERS OF POWER LOADS

Load ID.	Loc. node	r_l (p.u.)	L_l (p.u.)	Load ID.	Loc. node	r_l (p.u.)	L_l (p.u.)
1	2	3.32	2.33	15	18	3.68	2.57
2	3	3.67	2.57	16	19	3.69	2.58
3	4	2.75	1.92	17	20	3.66	2.56
4	5	5.49	3.84	18	21	3.66	2.56
5	7	1.64	1.15	19	22	3.65	2.55
6	8	1.64	1.15	20	23	3.66	2.56
7	9	5.50	3.85	21	24	0.78	0.54
8	10	5.52	3.86	22	26	5.48	3.83
9	11	7.38	5.16	23	27	5.47	3.82
10	12	5.54	3.88	24	28	5.42	3.79
11	14	2.74	1.91	25	29	2.70	1.89
12	15	4.89	4.08	26	30	0.24	0.72
13	16	5.55	3.89	27	31	2.17	1.51
14	17	5.53	3.87	28	32	1.55	1.08

TABLE III
COMPOSITION OF MICROGRIDS

MG 1[1]	Node ID.: 1, 2, 19, 20, 21, 22
	DER ID.: 1
	Load ID.: 1, 16, 17, 18, 19
MG 2	Node ID.: 23, 24, 25
	DER ID.: 4
	Load ID.: 20, 21
MG 3	Node ID.: 6, 26, 27, 28, 29, 30, 31, 32, 33
	DER ID.: 2, 5
	Load ID.: 22, 23, 24, 25, 26, 27, 28
MG 4	Node ID.: 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18
	DER ID.: 3
	Load ID.: 5, 6, 7, 8, 9, 10, 11, 12 ,13 ,14, 15

[1] Without loss of generality, impedance of each branch are assumed homogeneous with $r_b = 3.12 \times 10^{-3}$p.u., $L_b = 4.68 \times 10^{-3}$p.u..

APPENDIX B
DETAILS OF PHYSICS-BASED MICROGRID MODEL

This appendix details the physics-based model for each element in *InSys*.

In this paper, a two-layer hierarchical control is employed for DERs. Differential equations formulating the dynamics of inner and outer loops are detailed in our previous study [28]. Specifically, droop control is formulated as:

$$\boldsymbol{\omega} = \omega^* - \boldsymbol{m}_p(\boldsymbol{P} - \boldsymbol{P}^*) + \boldsymbol{\Omega} \tag{36a}$$

$$\boldsymbol{v} = \boldsymbol{v}^* - \boldsymbol{n}_q(\boldsymbol{Q} - \boldsymbol{Q}^*) + \boldsymbol{e} \tag{36b}$$

where $\boldsymbol{\omega}$ denotes DERs' angular speeds; $\boldsymbol{v}$ denotes DERs' voltages; ω^* and $\boldsymbol{v}^*$ respectively denote the reference values of speed and voltage; $\boldsymbol{P}^*$ and $\boldsymbol{Q}^*$ respectively denote the desired active/reactive power from DERs; $\boldsymbol{m}_p$ and $\boldsymbol{n}_p$ respectively denotes the active/reactive power droop gains of DERs. Specifically, $\boldsymbol{P}^*$ is impacted by available power from renewable energies, which introduces uncertainty to NMs dynamics.

Power sharing and voltages/frequency restoration in NMs are achieved by secondary control, which is formulated as:

$$\frac{\mathrm{d}\boldsymbol{\Omega}}{\mathrm{d}t} = -\boldsymbol{\alpha}(\boldsymbol{\omega} - \omega^*) - \boldsymbol{A}\boldsymbol{\Omega} \tag{37a}$$

$$\frac{\mathrm{d}\boldsymbol{e}}{\mathrm{d}t} = -\boldsymbol{\beta}(\boldsymbol{v} - \boldsymbol{v}^*) - \boldsymbol{B}(\boldsymbol{Q} \oslash \boldsymbol{Q}_n) \tag{37b}$$

Here, $\boldsymbol{\Omega}$ and $\boldsymbol{e}$ denote the secondary control signals following the distributed-averaging proportional-integral logic [37]; $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ are parameters for secondary control, where $\boldsymbol{A}$ and $\boldsymbol{B}$ represents the communication between different DERs; $\boldsymbol{Q}_n$ denotes the reactive power rating of DERs.

Dynamics of power loads and branches are modelled by impedance dynamics [5]:

$$\boldsymbol{L}_l \frac{\mathrm{d}\boldsymbol{i}_l}{\mathrm{d}t} = -\boldsymbol{r}_l\boldsymbol{i}_l + \omega_s\boldsymbol{I}_s\boldsymbol{L}_l\boldsymbol{i}_l + \boldsymbol{M}_l\boldsymbol{v} \tag{38}$$

$$\boldsymbol{L}_b \frac{\mathrm{d}\boldsymbol{i}_b}{\mathrm{d}t} = -\boldsymbol{r}_b\boldsymbol{i}_b + \omega_s\boldsymbol{I}_s\boldsymbol{L}_b\boldsymbol{i}_b + \boldsymbol{M}_b\boldsymbol{v} \tag{39}$$

where $\boldsymbol{i}_l$ and $\boldsymbol{i}_b$ respectively denote the DQ-axis currents of loads and branches. $\boldsymbol{r}_l$ and $\boldsymbol{L}_l$ respectively denote the matrices of load resistances and inductance; $\boldsymbol{M}_l$ is the DQ-axis incidence matrix between power loads and buses; $\boldsymbol{r}_b$, $\boldsymbol{L}_b$ and $\boldsymbol{M}_b$ are similarly defined for branches. Formulation of arbitrary types of power loads is further provided in [5].

APPENDIX C
ODE-GOVERNED *EXSYS* FORMULATION

This appendix briefly explains the rationale behind the use of ODE-Net to learn an ODE-governed *ExSys* model. Without loss of generality, we take droop control as an example. To make a distinction from the main text, this appendix uses subscript for variables.

Based on the differential equations of DERs, power loads and branches in Appendix. B, *ExSys* model can be functionally abstracted as:

$$\left\{\begin{aligned} \dot{\boldsymbol{z}}_e &= \boldsymbol{g}_e(\boldsymbol{i}_e, \boldsymbol{z}_e, \boldsymbol{w}_e) && \text{(40a)}\\ \dot{\boldsymbol{i}}_e &= \boldsymbol{f}_e(\boldsymbol{i}_e, \boldsymbol{z}_e) + \boldsymbol{m}_e\boldsymbol{v}_e + \boldsymbol{m}_{bd}\boldsymbol{v}_{bd} && \text{(40b)}\\ \boldsymbol{0} &= \boldsymbol{n}_e\boldsymbol{i}_e + \boldsymbol{n}_{bd}\boldsymbol{i}_{bd} && \text{(40c)} \end{aligned}\right.$$

Here, (40a), (40b) and (40c) respectively formulate the dynamics of DER states ($\boldsymbol{z}_e$), component current ($\boldsymbol{i}_e$) and KCL. $\boldsymbol{v}_e$ and $\boldsymbol{v}_{bd}$ separately denote voltages at *ExSys* buses and boundary buses; $\boldsymbol{i}_{bd}$ denotes current inflows from boundary buses; $\boldsymbol{w}_e$ denotes uncertainty from renewable energy. Since *ExSys* is unidentified, expressions of $\boldsymbol{g}_e$, $\boldsymbol{f}_e$, $\boldsymbol{m}_e$, $\boldsymbol{m}_{bd}$, $\boldsymbol{n}_e$, $\boldsymbol{n}_{bd}$ are unknown. Neither are the detailed constituents of $\boldsymbol{z}_e$ and $\boldsymbol{i}_e$.

Because (40c) is linear, (40) can be reformulated to a rigorously equivalent ODE system [28]. Denoting $\boldsymbol{n}_{e1}$ as the non-singular matrix constructed by the maximal linearly independent columns of $\boldsymbol{n}_e$, (40c) yields:

$$\boldsymbol{i}_{e1} = -\boldsymbol{n}_{e1}^{-1}(\boldsymbol{n}_{e0}\boldsymbol{i}_{e0} + \boldsymbol{n}_{bd}\boldsymbol{i}_{bd}) \tag{41}$$

where $\boldsymbol{n}_{e0}$ denotes the rest columns of $\boldsymbol{n}_e$; $\boldsymbol{i}_{e1}$ and $\boldsymbol{i}_{e0}$ are sub-vectors of $\boldsymbol{i}_e$ respectively corresponding to $\boldsymbol{n}_{e1}$ and $\boldsymbol{n}_{e0}$. Substituting (41) into (40) yields:

$$\left\{\begin{aligned} \dot{\boldsymbol{z}}_e &= \hat{\boldsymbol{g}}_e(\boldsymbol{i}_{e0}, \boldsymbol{z}_e, \boldsymbol{w}_e) && \text{(42a)}\\ \dot{\boldsymbol{i}}_{e0} &= \hat{\boldsymbol{f}}_{e0}(\boldsymbol{i}_{e0}, \boldsymbol{z}_e) + \boldsymbol{m}_{bd0}\boldsymbol{v}_{bd} - \boldsymbol{m}_{e0}(\boldsymbol{n}_e\boldsymbol{m}_e)^{-1} \\ &\quad (\boldsymbol{n}_e\hat{\boldsymbol{f}}_e(\boldsymbol{i}_{e0}, \boldsymbol{z}_e) + \boldsymbol{n}_e\boldsymbol{m}_{bd}\boldsymbol{v}_{bd} + \boldsymbol{n}_{bd}\dot{\boldsymbol{i}}_{bd}) && \text{(42b)} \end{aligned}\right.$$

where $\hat{\boldsymbol{g}}_e(\boldsymbol{i}_{e0}, \boldsymbol{z}_e, \boldsymbol{w}_e) = \boldsymbol{g}_e(\boldsymbol{i}_e, \boldsymbol{z}_e, \boldsymbol{w}_e)$ and $\hat{\boldsymbol{f}}_e(\boldsymbol{i}_{e0}, \boldsymbol{z}_e) = \boldsymbol{f}_e(\boldsymbol{i}_e, \boldsymbol{z}_e)$ by substituting (41); $\hat{\boldsymbol{f}}_{e0}$, $\boldsymbol{m}_{bd0}$ and $\boldsymbol{m}_{e0}$ are respectively sub-vectors/sub-matrices corresponding to $\boldsymbol{n}_{e0}$.

As indicated by (41), *ExSys* is reformulated in a pure ODE form, which enables the application of ODE-Net for dynamic model discovery.

REFERENCES

[1] P. Zhang, *Networked Microgrids*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[2] F. Feng and P. Zhang, "Enhanced microgrid power flow incorporating hierarchical control," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 2463–2466, May 2020.

[3] W. Wan *et al.*, "Distributed and asynchronous active fault management for networked microgrids," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 3857–3868, Sep. 2020.

[4] N. Soni, S. Doolla, and M. C. Chandorkar, "Improvement of transient response in microgrids using virtual inertia," *IEEE Trans. Power Del.*, vol. 28, no. 3, pp. 1830–1838, Jul. 2013.

[5] Y. Zhou, P. Zhang, and M. Yue, "Reachable dynamics of networked microgrids with large disturbances," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2416–2427, May 2020.

[6] T. A. Wilding *et al.*, "Turning off the drip ('data-rich, information-poor')-rationalising monitoring with a focus on marine renewable energy developments and the benthos," *Renewable Sustain. Energy Rev.*, vol. 74, pp. 848–859, 2017.

[7] Y. Xu, Z. Y. Dong, L. Guan, R. Zhang, K. P. Wong, and F. Luo, "Preventive dynamic security control of power systems based on pattern discovery technique," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1236–1244, Aug. 2012.

[8] Y. Zhou and P. Zhang, "Reachable power flow," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 3290–3293, Jul. 2020.

[9] Y. Zhou and P. Zhang, "Reachable eigenanalysis," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4936–4939, Nov. 2020.

[10] Y. Liu, K. Sun, R. Yao, and B. Wang, "Power system time domain simulation using a differential transformation method," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3739–3748, Sep. 2019.

[11] Z. Yao, "A sufficient and necessary condition of transient stability for multimachine systems," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 683–690, Jan. 2020.

[12] H. Bosetti and S. Khan, "Transient stability in oscillating multi-machine systems using Lyapunov vectors," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2078–2086, Mar. 2018.

[13] M. Korda and I. Mezić, "Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control," *Automatica*, vol. 93, pp. 149–160, 2018.

[14] A. Surana and A. Banaszuk, "Linear observer synthesis for nonlinear systems using Koopman operator framework," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 716–723, 2016.

[15] S. Sinha, B. Huang, and U. Vaidya, "On robust computation of Koopman operator and prediction in random dynamical systems," *J. Nonlinear Sci.*, vol. 30, pp. 2057–2090, 2019.

[16] T. Guo and J. V. Milanović, "Online identification of power system dynamic signature using PMU measurements and data mining," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 1760–1768, May 2016.

[17] L. Zheng *et al.*, "Deep belief network based nonlinear representation learning for transient stability assessment," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2017, pp. 1–5.

[18] A. G. Parlos, K. T. Chong, and A. F. Atiya, "Application of the recurrent multilayer perceptron in modeling complex process dynamics," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 255–266, Mar. 1994.

[19] C.-W. Liu, M.-C. Su, S.-S. Tsay, and Y.-J. Wang, "Application of a novel fuzzy neural network to real-time transient stability swings prediction based on synchronized phasor measurements," *IEEE Trans. Power Syst.*, vol. 14, no. 2, pp. 685–692, May 1999.

[20] S. Wen, Y. Wang, Y. Tang, Y. Xu, P. Li, and T. Zhao, "Real-time identification of power fluctuations based on LSTM recurrent neural network: A case study on singapore power system," *IEEE Trans. Ind. Inform.*, vol. 15, no. 9, pp. 5266–5275, Sep. 2019.

[21] S. K. Azman, Y. J. Isbeih, M. S. El Moursi, and K. Elbassioni, "A unified online deep learning prediction model for small signal and transient stability," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4585–4598, Nov. 2020.

[22] Z. Shi *et al.*, "Convolutional neural network-based power system transient stability assessment and instability mode prediction," *Appl. Energy*, vol. 263, 2020, Art. no. 114586.

[23] A. Gupta, G. Gurrala, and P. Sastry, "An online power system stability monitoring system using convolutional neural networks," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 864–872, Mar. 2019.

[24] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2020, pp. 1–5.

[25] T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6571–6583.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learn*. Cambridge: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[27] Y. Sun, L. Zhang, and H. Schaeffer, "NeuPDE: Neural network based ordinary and partial differential equations for modeling time-dependent data," in *Proc. 1st Math. Scientific Mach. Learn. Conf.*, 2020, pp. 352–372.

[28] Y. Zhou, P. Zhang, and M. Yue, "An ODE-enabled distributed transient stability analysis for networked microgrids," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, 2020, pp. 1–5.

[29] M. Althoff and B. H. Krogh, "Zonotope bundles for the efficient computation of reachable sets," in *Proc. 50th IEEE Conf. Decis. Control*, 2011, pp. 6814–6821.

[30] M. Althoff, O. Stursberg, and M. Buss, "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization," in *Proc. 47th IEEE Conf. Decis. Control*, 2008, pp. 4042–4048.

[31] S. B. Liu and M. Althoff, "Reachset conformance of forward dynamic models for the formal analysis of robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 370–376.

[32] N. Kochdumper, A. Tarraf, M. Rechmal, M. Olbrich, L. Hedrich, and M. Althoff, "Establishing reachset conformance for the formal analysis of analog circuits," in *Proc. 25th Asia South Pacific Des. Automat. Conf.*, 2020, pp. 199–204.

[33] N. Pogaku, M. Prodanovic, and T. C. Green, "Modeling, analysis and testing of autonomous operation of an inverter-based microgrid," *IEEE Trans. Power Electron.*, vol. 22, no. 2, pp. 613–625, Mar. 2007.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2014, pp. 1–15.

[35] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Multistep neural networks for data-driven discovery of nonlinear dynamical systems," 2018, *arXiv:1801.01236*.

[36] H. Shakouri and H. R. Radmanesh, "Identification of a continuous time nonlinear state space model for the external power system dynamic equivalent by neural networks," *Int. J. Elect. Power Energy Syst.*, vol. 31, no. 7/8, pp. 334–344, 2009.

[37] J. W. Simpson-Porco, Q. Shafiee, F. Dörfler, J. C. Vasquez, J. M. Guerrero, and F. Bullo, "Secondary frequency and voltage control of islanded microgrids via distributed averaging," *IEEE Trans. Ind. Electron.*, vol. 62, no. 11, pp. 7025–7038, Nov. 2015.

Yifan Zhou (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2014 and 2019, respectively. She is currently a Postdoctoral Researcher with Stony Brook University, Stony Brook, NY, USA. Her research interests include microgrid stability and control, formal methods and reachability analysis, and quantum computing.

Peng Zhang (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2009. He is currently a SUNY Empire Innovation Professor with Stony Brook University, Stony Brook, NY, USA. He has a joint appointment with Brookhaven National Laboratory, Upton, NY, USA, as a Staff Scientist. Previously, he was a Centennial Associate Professor and a Francis L. Castleman Associate Professor with the University of Connecticut, Storrs, CT, USA. During 2006–2010, he was a System Planning Engineer with BC Hydro and Power Authority, Canada. His research interests include programmable microgrids, networked microgrids, quantum security, quantum computing, power system stability and control, cyber security, formal methods and reachability analysis, and software-defined networking.

Dr. Zhang is an individual Member of CIGRÉ. He is the Editor of the IEEE TRANSACTIONS ON POWER SYSTEMS, IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, and IEEE POWER AND ENERGY SOCIETY LETTERS, and an Associate Editor for the IEEE JOURNAL OF OCEANIC ENGINEERING.