Neuro-Dynamic State Estimation for Networked Microgrids

Fei Feng, *Graduate Student Member, IEEE*, Yifan Zhou, *Member, IEEE*, and Peng Zhang

***Abstract*—The increasing integration of distributed energy resources (DERs) brings complicated dynamics in networked microgrids (NMs), calling for high-fidelity dynamic state estimation (DSE) of NMs. Traditional DSE, which requires accurate physical models of the entire NMs, is becoming increasingly unattainable. This paper devises neuro-dynamic state estimation (*Neuro-DSE*), a learning-based DSE algorithm to track the dynamics of inverter-interfaced NMs with unknown subsystems. The process and contributions include: 1) a data-driven *Neuro-DSE* algorithm is established for NMs with partially unidentified dynamic models by incorporating the neural-ordinary-differential-equations (ODE-Net) into Kalman filters; 2) a self-refined *Neuro-DSE*$^+$ method is devised to tackle limited and noisy measurements. Specifically, Kalman filters are embedded into ODE-Net training for automatic filtering, augmenting, and correcting effects; 3) a *Neuro-KalmanNet-DSE* algorithm is derived to relieve the model mismatch scenarios by integrating KalmanNet with *Neuro-DSE*. Numerical simulations carried out on typical four-microgrid NMs reveal that *Neuro-DSE* can track the dynamics under various control modes (e.g., droop/secondary controls) and components. Its variants increase the accuracy of *Neuro-DSE* under limited measurement and model mismatch scenarios.**

***Index Terms*—Networked microgrids, neuro-dynamic state estimation, Kalman filter, neural ordinary differential equations, KalmanNet.**

I. Introduction

In SUPPORT of power sector's decarbonization and energy resiliency, networked microgrids (NMs) are being increasingly developed as they can collaboratively serve critical communities and host distributed energy resources (DERs) [1]. Today's NMs are undergoing an increasing integration of inverter-interfaced renewable resources. The ubiquitous uncertainties [2], deteriorated inertia [3], frequent plug-and-play [4], [5], as well as unforeseen failures [6], [7], may jointly trigger complicated dynamic processes in NMs operations, creating the high need for accurate evaluations of the dynamic states of NMs.

Dynamic state estimation (DSE) is an indispensable foundation for power system operation as it provides the most-likely dynamic states of the system to perform online monitoring and control [8], [9]. However, the accurate physics models of each inverter-based DER and the entire NMs may not always be attainable to support existing DSE methods to precisely track the fast dynamics of NMs, especially the internal states of inverter controllers.

Physics-based DSE algorithms, represented by Kalman filter and its variants [10], [11], [12], [13], strongly rely on accurate dynamic models of the whole system to estimate the system states [14], [15]. Ref. [16] used a model-based equivalence method to reduce the parameters of the detailed model. Then, it integrates the equivalent physics model into the Kalman filter to estimate the required dynamic states for the modified controller. However, the complete physical models in NMs are often unattainable due to unavailable parameters of distributed inverter controllers, frequently changing control modes and plug-and-play of DERs, data privacy needs, etc. The complications of physical models may lead to subsystems with unidentified dynamic models in NMs, which unavoidably make the classical, physics-based DSE algorithms impractical [17], [18].

Recent progress in learning dynamic models from data shed light on developing data-driven DSE without requiring explicit physics of the entire system. Ref. [19] employs a long short-term memory (LSTM) networks to realize the DC microgrid state estimation. An improved residual network (ResNet) is applied for the state estimation of the distribution system by collecting historical data [20]. Meanwhile, a deep neural network (DNN) based hybrid DSE is established for multi-machine power systems [21]. Recently, neural-ordinary-differential-equations (ODE-Net) [22] emerges to become an efficacious paradigm for learning underlying dynamic models of power systems [23], which also ignites new hopes for data-driven DSE because it can best preserve the continuous-time dynamic characteristics. Nevertheless, two fundamental obstacles still hinder the application of existing data-driven approaches to DSE for real-world NMs: I) Data-driven dynamic models learned from limited and noisy measurements may not satisfy the accuracy needs of DSE; and II) Mismatches between data-driven models and real dynamic measurements unavoidably bias the state estimator.

To bridge the gap, this paper devises neuro-dynamic state estimation (*Neuro-DSE*), which integrates ODE-Net with both physics-based and neural-network-based Kalman filter theories

Manuscript received 22 August 2023; revised 22 December 2023; accepted 4 January 2024. Date of publication 1 March 2024; date of current version 31 January 2025. Paper 2023-KDSEM-1062.R1, approved for publication in the IEEE Transactions on Industry Applications by the Knowledge-and Data-Driven Smart Energy Management in Distribution Networks of the IEEE Industry Applications Society. This work was supported in part by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number 38456 and in part by the National Science Foundation under Grant OIA-2040599 and Grant OIA-2134840. *(Corresponding author: Yifan Zhou.)*

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

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIA.2024.3371956.

Digital Object Identifier 10.1109/TIA.2024.3371956

to allow for learning-based DSE of NMs under unidentified subsystems, limited measurements, and potential model mismatches. Our contributions are threefold:

- An ODE-Net-enabled dynamic state estimation (*Neuro-DSE*) method is established to address DSE in NMs with unidentified subsystem models. *Neuro-DSE* integrates hybrid physics/ODE-Net-based dynamic models with Kalman filters for NMs' awareness, in which ODE-Net well preserves the continuous-time dynamics characteristics of the unknown subsystems. In addition, this feature enables ODE-Net more resilient to the noise in the measurement values.
- A self-refined Neuro-DSE (*Neuro-DSE*$^+$) algorithm is then devised to enable efficacious data-driven DSE under noisy and limited measurement scenarios. The self-refined training framework embeds the Kalman filters into ODE-Net training which enables full states of identified subsystems for ODE-Net training. Thus, the accuracy and expressibility of ODE-Net are enhanced.
- A KalmanNet-enhanced Neuro-DSE (*Neuro-KalmanNet-DSE*) is established to address possible model mismatch scenarios induced by data-driven dynamic models. In this variant, KalmanNet which integrates a dedicated recurrent neural network (RNN) into Kalman filters to adaptively track NMs' states under imprecise knowledge of dynamic models.

The remainder of this paper is organized as follows: Section II devises the ODE-Net-enabled *Neuro-DSE* algorithm. Section III develops the self-refined *Neuro-DSE*$^+$. Section IV establishes *Neuro-KalmanNet-DSE*. Section V presents case studies on a typical NMs system to verify the effectiveness of our methods, followed by the Conclusion in Section VI.

II. Neuro-Dynamic State Estimation

This section devises neuro-dynamic state estimation (*Neuro-DSE*), a data-driven DSE algorithm for estimating the dynamic states of NMs with partially known physics models.

A. Preliminaries of Physics-Based DSE

DSE targets tracking the dynamic states of a discrete-time nonlinear system [24], which can be described as:

$$\begin{cases} \boldsymbol{x}_k = \boldsymbol{f}(\boldsymbol{x}_{k-1}) + \boldsymbol{w}_k \\ \boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k) + \boldsymbol{r}_k \end{cases} \tag{1}$$

where, $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ denote the state variables and measurement variables at time step k, respectively; $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$ denote the discrete-time process and measurement functions, respectively; $\boldsymbol{w}_k$ and $\boldsymbol{r}_k$ are process and measurement noises, respectively.

Kalman filter is a mainstream algorithm for DSE [10], [11], [12]. This subsection takes the extended Kalman filter (EKF) [10], i.e., a prominent Kalman filter variant, as a representative to introduce the basis of physics-based DSE. EKF consists of two kernel steps, i.e., prediction and correction:

- *Prediction:* The current predicted states $\boldsymbol{x}_{k|k-1}$ are calculated via the estimated states $\boldsymbol{x}_{k-1|k-1}$ at the previous step:

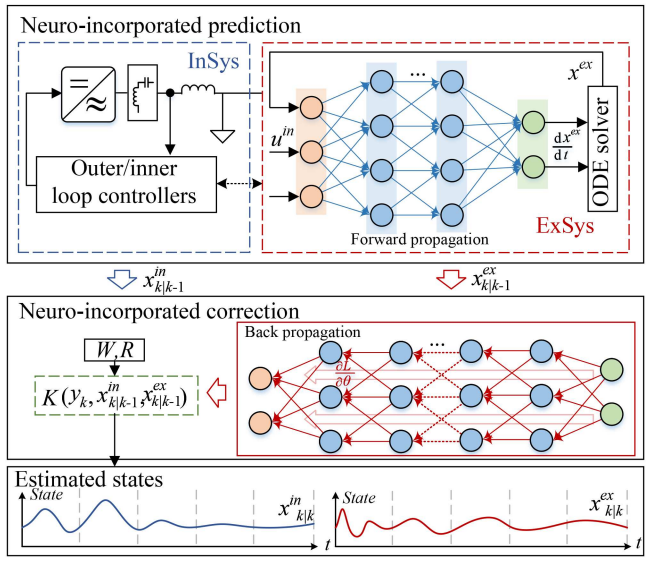

Fig. 1. Architecture of the *Neuro-DSE* algorithm.

$$\boldsymbol{x}_{k|k-1} = \boldsymbol{f}(\boldsymbol{x}_{k-1|k-1}) \tag{2}$$

- *Correction:* The predicted states are corrected based on the noisy measurements $\tilde{\boldsymbol{y}}_k$ and the estimated states $\boldsymbol{x}_{k|k}$ at the current step are generated:

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1} + \boldsymbol{K}_k \cdot (\tilde{\boldsymbol{y}}_k - \boldsymbol{h}(\boldsymbol{x}_{k|k-1})) \tag{3}$$

Here, $\boldsymbol{K}_k$ denotes the Kalman gain, which is derived from $\boldsymbol{f}(\cdot)$ and $\boldsymbol{h}(\cdot)$, i.e., the entire physics model of the system.

Obviously, conventional DSE relies on physics models to estimate the system states, which hinders its application in NMs with unknown subsystem models. Thus, we devise *Neuro-DSE*, which integrates a learning-based ODE-Net dynamic model into Kalman filter to enable data-driven DSE.

B. Neuro-DSE Algorithm of NMs

Fig. 1 demonstrates the outline of the *Neuro-DSE* algorithm. The core idea of *Neuro-DSE* is to establish a data-driven dynamic model of the unidentified subsystems so that the dynamic states of the rest of the NMs can still be estimated via the Kalman filter. As illustrated on the top of Fig. 1, without loss of generality, the NMs system is divided into an external subsystem (ExSys), whose dynamic model is unidentified, and an internal subsystem (InSys), whose dynamic model is well defined by its physics natures.[1] Our target is to construct a data-driven dynamic model of ExSys, thereby performing a neural-network-incorporated DSE of InSys.

In the following, we successively establish the ODE-Net-enabled, data-based formulation for ExSys, the physics-based formulation for InSys, and finally the *Neuro-DSE* algorithm based on the physics-neural-integrated NMs formulation.

[1]The "well-defined" InSys means the physics model of InSys is attainable. However, measurements of InSys can be partial (i.e., not fully observable).

1) ODE-Net-Enabled ExSys Modeling: ODE-Net is capable of learning continuous-time dynamic models from discrete-time measurements, which exhibits superior noise-resilience over conventional deep neural networks (DNNs) [22], [23].

In this subsection, ODE-Net is employed to formulate ExSys based on available measurements:

$$\frac{d\boldsymbol{x}^{ex}}{dt} = \boldsymbol{F}(\boldsymbol{x}^{ex}, \boldsymbol{u}^{in}) \tag{4}$$

In (4), $\boldsymbol{x}^{ex}$ denotes the states measured from ExSys, which can consist of both physics quantities and control signals.[2] $\boldsymbol{u}^{in}$ represents the measurable state variables of InSys, which reflects the interactions between InSys and ExSys. Function $\boldsymbol{F}$ denotes the state-space model of ExSys governed by the forward propagation of ODE-Net:

$$\boldsymbol{F} = \boldsymbol{f}_l(\boldsymbol{f}_{l-1}(\cdots \boldsymbol{f}_1(\boldsymbol{x}^{ex}, \boldsymbol{u}^{in}, \boldsymbol{\theta}_1)\cdots, \boldsymbol{\theta}_{l-1}), \boldsymbol{\theta}_l) \tag{5}$$

where $\boldsymbol{f}_l(\cdot)$ and $\boldsymbol{\theta}_l$ respectively denote the function and the trainable parameters of the lth layer of ODE-Net.

To make ODE-Net best match ExSys' dynamics, the loss function is set as the error between the numerical integration results of (4) and the time-series measurements of ExSys:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}} \quad L(\boldsymbol{\theta}) = \sum_{k=1}^{n} (\boldsymbol{x}_k^{ex} - \tilde{\boldsymbol{x}}_k^{ex})^2 + \gamma \cdot \boldsymbol{\theta}^2 \tag{6a}$$

$$\text{s.t.} \quad \boldsymbol{x}_k^{ex} = \tilde{\boldsymbol{x}}_1^{ex} + \int_{t_1}^{t_k} \boldsymbol{F}(\boldsymbol{x}^{ex}, \tilde{\boldsymbol{u}}^{in}, \boldsymbol{\theta}) dt \tag{6b}$$

where, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1\} \cup \cdots \cup \{\boldsymbol{\theta}_L\}$ denotes all the trainable parameters of ODE-Net; γ denotes the regularization coefficient; n is the total number of time slides; $\tilde{\boldsymbol{x}}_k^{ex}$ denotes the measurements of ExSys (e.g., current injections to InSys, global control signals sent to InSys) at time point k.

Because (6) incorporates numerical integration as constraints, gradient descent based on a continuous backpropagation technique is applied to train the ODE-Net until the loss function converges:

$$\boldsymbol{\theta} - \eta \frac{\partial L}{\partial \boldsymbol{\theta}} \to \boldsymbol{\theta} \quad , \quad \frac{\partial L}{\partial \boldsymbol{\theta}}|_{t_1} = \int_{t_n}^{t_1} \boldsymbol{\lambda}^T \frac{\partial F}{\partial \boldsymbol{\theta}} \tag{7}$$

where, η denotes the learning rate; λ denotes the Lagrangian multiplier corresponding to constraints (6b).

2) Physics-Based InSys Modeling: InSys is formulated by its physics natures. Considering the fast dynamics of all the DERs (i.e., droop/secondary controls), branch lines and power loads, it can be proved that the differential algebraic equation-based InSys model can be rigorously converted into a system of ordinary differential equations [25]. Hence, InSys is functionally formulated as:

$$\frac{d\boldsymbol{x}^{in}}{dt} = \boldsymbol{G}(\boldsymbol{x}^{in}, \boldsymbol{x}^{ex}) \tag{8}$$

[2]Specifically, in this work, we assume that, for ExSys, only the boundary dynamic behaviours (i.e., current injections to InSys) and control signals sent to InSys (i.e., secondary control signals of grid-forming inverters in ExSys) can be measured, which represents very limited measurements. However, the method is adaptive to arbitrary measurements from ExSys.

where, $\boldsymbol{x}^{ex}$ denotes ExSys states (see (4)); $\boldsymbol{x}^{in}$ denotes InSys states (e.g., state variables of each DER, load, branch).

3) Neuro-DSE Algorithm: By integrating and discretizing the ODE-Net-enabled ExSys and the physics-enabled InSys, (9) constructs the model basis of *Neuro-DSE*, which is a discrete-time, physics-neural-integrated NMs model:

$$\begin{cases} \boldsymbol{x}_k^{ex} = \mathcal{D}\boldsymbol{F}(\boldsymbol{x}_{k-1}^{ex}, \boldsymbol{u}_{k-1}^{in}) + \boldsymbol{w}_k^{ex} & \text{(9a)} \\ \boldsymbol{x}_k^{in} = \mathcal{D}\boldsymbol{G}(\boldsymbol{x}_{k-1}^{in}, \boldsymbol{x}_{k-1}^{ex}) + \boldsymbol{w}_k^{in} & \text{(9b)} \\ \boldsymbol{y}_k = \boldsymbol{M}(\boldsymbol{x}_k^{ex}, \boldsymbol{x}_k^{in}) + \boldsymbol{r}_k & \text{(9c)} \end{cases}$$

where, $\mathcal{D}$ denotes a discretization operator, which discretizes the neural/physics dynamics $\boldsymbol{F}(\cdot)$ and $\boldsymbol{G}(\cdot)$ presented in (4) and (8); $\boldsymbol{M}(\cdot)$ denotes the measurement function of NMs; $\boldsymbol{y}_k$ denotes the measurement variables; $\boldsymbol{w}_k^{ex}$ and $\boldsymbol{w}_k^{in}$ denote the Gaussian processing noises which follow Gaussian noise sequences $\mathcal{N}(0, W)$; $\boldsymbol{r}_k$ is the Gaussian measurement noise following $\mathcal{N}(0, R)$, where W and R are the corresponding covariance matrices.

Without loss of generality, we derive the *Neuro-DSE* algorithm based on the EKF method. Yet, the algorithm is readily compatible with arbitrary Kalman-type filters. *Neuro-DSE* is also composed of a predictor and a corrector (see Fig. 1). However, because of the incorporation of ODE-Net-based modeling, both the prediction and the correction will be involved with neural network operations:

- *Neuro-incorporated prediction:* The prediction step predicts the InSys and ExSys states based on the estimation at the previous step. While the prediction of $\boldsymbol{x}^{in}$ is trivial, the prediction of $\boldsymbol{x}^{ex}$ involves the forward propagation of ODE-Net according to (9a) and (5):

$$\begin{aligned} \boldsymbol{x}_{k|k-1}^{ex} = \mathcal{D}\boldsymbol{f}_L(\boldsymbol{f}_{L-1}(\cdots \boldsymbol{f}_1(\boldsymbol{x}_{k-1|k-1}^{ex}, \boldsymbol{u}_{k-1|k-1}^{in}, \boldsymbol{\theta}_1) \\ \cdots, \boldsymbol{\theta}_{L-1}), \boldsymbol{\theta}_L) \end{aligned} \tag{10}$$

  where, k denotes the current time step.
- *Neuro-incorporated correction:* The correction step generates the estimated states by correcting the predictions:

$$\begin{bmatrix} \boldsymbol{x}_{k|k}^{ex} \\ \boldsymbol{x}_{k|k}^{in} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{k|k-1}^{ex} \\ \boldsymbol{x}_{k|k-1}^{in} \end{bmatrix} + \boldsymbol{K}_k \cdot (\tilde{\boldsymbol{y}}_k - \boldsymbol{M}(\boldsymbol{x}_{k|k-1}^{ex}, \boldsymbol{x}_{k|k-1}^{in})) \tag{11}$$

  where, $\tilde{\boldsymbol{y}}_k$ denotes the noisy measurements of NMs. Specifically, the Kalman gain $\boldsymbol{K}_k$ is given by (12), which requires the backward gradients of ODE-Net w.r.t. $\boldsymbol{x}^{ex}$ and $\boldsymbol{u}^{in}$:

$$\begin{aligned} \boldsymbol{K}_k = (\boldsymbol{J}_k \boldsymbol{\Sigma}_{k-1} \boldsymbol{J}_k^T + \boldsymbol{W}) \cdot (\boldsymbol{J}_k^M)^T \cdot \\ (\boldsymbol{J}_k^M \cdot (\boldsymbol{J}_k \boldsymbol{\Sigma}_{k-1} \boldsymbol{J}_k^T + \boldsymbol{W}) \cdot (\boldsymbol{J}_k^M)^T + \boldsymbol{R})^{-1} \end{aligned} \tag{12}$$

  where $\boldsymbol{J}_k = \begin{bmatrix} \frac{\partial(\mathcal{D}\boldsymbol{F})}{\partial \boldsymbol{x}^{ex}} & \frac{\partial(\mathcal{D}\boldsymbol{F})}{\partial \boldsymbol{x}^{in}} \\ \frac{\partial(\mathcal{D}\boldsymbol{G})}{\partial \boldsymbol{x}^{ex}} & \frac{\partial(\mathcal{D}\boldsymbol{G})}{\partial \boldsymbol{x}^{in}} \end{bmatrix}$ and $\boldsymbol{J}_k^M = \begin{bmatrix} \frac{\partial \boldsymbol{M}}{\partial \boldsymbol{x}^{ex}} & \frac{\partial \boldsymbol{M}}{\partial \boldsymbol{x}^{in}} \end{bmatrix}$ respectively denote the Jacobian matrices; $\boldsymbol{W}$ and $\boldsymbol{R}$ respectively denote the noise and measurement covariance matrix; $\boldsymbol{\Sigma}_k$ denotes the covariance matrices iteratively calculated by $\boldsymbol{\Sigma}_k = (\boldsymbol{J}_k \boldsymbol{\Sigma}_{k-1} \boldsymbol{J}_k^T + \boldsymbol{W})(\boldsymbol{I} - (\boldsymbol{J}_k^M)^T \cdot \boldsymbol{K}_k^T)$.

Consequently, by integrating ODE-Net with the process functions and covariance evolution of Kalman filters, *Neuro-DSE* enables state estimation of the accessible subsystem of the NMs

(i.e., InSys) even without the physics model of the inaccessible subsystems (i.e., ExSys).

III. *Neuro-DSE*$^+$: Self-Refined Neuro-Dynamic State Estimation

Neuro-DSE relies on measurements to learn dynamic models of unidentified subsystems and perform data-driven DSE. Its efficacy might be jeopardized when the excessively limited and noisy measurements fail to generate a qualified ODE-Net. This section enhances *Neuro-DSE* by devising a self-refined neuro-dynamic state estimation (*Neuro-DSE*$^+$). *Neuro-DSE*$^+$ is able to proactively augment and filter the measurements, and therefore it provides a more efficacious data-driven DSE especially under limited measurements.

A. Self-Refined Training of ODE-Net

ODE-Net plays an important role in *Neuro-DSE*, as it learns the dynamic model of unidentified subsystems. The core idea of *Neuro-DSE*$^+$ is to enhance the quality and quantity of the data used for ODE-Net training, therefore obtaining a sufficiently accurate data-driven model even under noisy and limited measurements.

We reformulate the ODE-Net-enabled dynamic model of ExSys with an augmented input:

$$\frac{d\boldsymbol{x}^{ex}}{dt} = \boldsymbol{F}(\boldsymbol{x}^{ex}, \boldsymbol{x}^{in}) \tag{13}$$

Comparing (13) with (4), an obvious distinction is that (13) incorporates the complete set of the InSys states $\boldsymbol{x}^{in}$ (rather than merely the measurable states $\boldsymbol{u}^{in}$) to enrich the input information into ODE-Net.

However, because of the limited measurement, $\boldsymbol{x}^{in}$ may not be fully accessible, meaning that it can not be directly used for training as Section II-B. Therefore, we establish a self-refined training procedure for ODE-Net:

$$\min_{\boldsymbol{\theta}\in\mathbb{R}} L^+(\boldsymbol{\theta}) = \sum_{k=1}^{n} (\boldsymbol{x}_k^{ex} - \boldsymbol{x}_{k|k}^{ex})^2 + \gamma \cdot \boldsymbol{\theta}^2 \tag{14a}$$

$$\text{s.t. } \boldsymbol{x}_k^{ex} = \tilde{\boldsymbol{x}}_1^{ex} + \int_{t_1}^{t_k} \boldsymbol{F}(\boldsymbol{x}^{ex}, \boldsymbol{x}_{i|i}^{in}, \boldsymbol{\theta}) dt \tag{14b}$$

$$\begin{bmatrix} \boldsymbol{x}_{k|k}^{ex} \\ \boldsymbol{x}_{k|k}^{in} \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{k|k-1}^{ex} \\ \boldsymbol{x}_{k|k-1}^{in} \end{bmatrix} + \boldsymbol{K}_k \cdot (\tilde{\boldsymbol{y}}_k - \boldsymbol{M}(\boldsymbol{x}_{k|k-1}^{ex}, \boldsymbol{x}_{k|k-1}^{in})) \tag{14c}$$

where, $\boldsymbol{x}_{k|k}^{ex}$ denotes the estimated states of ExSys at time point k; $\boldsymbol{x}_{i|i}^{ex}$ denotes the estimated states of InSys at time point i (i.e., corresponding to time t_i); other notations are the same as defined in (11).

As illustrated in Fig. 2, a salient feature of (14) is that it embeds the Kalman filter process (14c) into ODE-Net training, which enables: (i) constructing a loss function between ODE-Net's predictions and the filtered ExSys states and therefore mitigating the impact of noisy measurements; (ii) constructing ExSys dynamics (14b) using the full states of InSys and therefore greatly enriching the expressibility of ODE-Net.

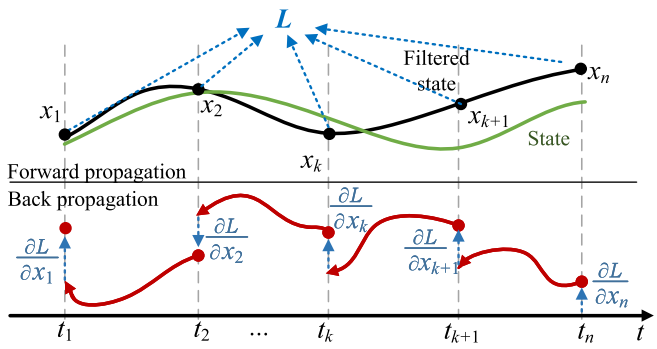

Fig. 2. Self-refined training of ODE-Net.

Algorithm 1: *Neuro-DSE* $^+$.

Initialize: $\boldsymbol{\theta}$, $\tilde{\boldsymbol{x}}_k^{ex}$, $\tilde{\boldsymbol{u}}_k^{in}$, $\boldsymbol{F}(\cdot)$, $\boldsymbol{G}(\cdot)$, $\boldsymbol{M}(\cdot)$, k ;
if *Pre-training* **then**
  Input: $\boldsymbol{\theta}$, $\tilde{\boldsymbol{x}}_k^{ex}$, $\tilde{\boldsymbol{u}}_k^{in}$, $\boldsymbol{F}(\cdot)$, $\boldsymbol{G}(\cdot)$, $\boldsymbol{M}(\cdot)$;
  Execute $\tilde{\boldsymbol{x}}_1^{ex} \xrightarrow{\boldsymbol{F}(\cdot),\tilde{\boldsymbol{u}}_k^{in}} \boldsymbol{x}_k^{ex}$ **Eq.**(4);
  Execute $\min_{\boldsymbol{\theta}\in\mathbb{R}} \sum_{k=1}^{n} L(\boldsymbol{x}_k^{ex}, \tilde{\boldsymbol{x}}_k^{ex})$ **Eq.**(6,7);
  Estimate and output $\boldsymbol{x}_{k|k}^{ex}$, $\boldsymbol{x}_{k|k}^{in}$ **Eq.**(5-12);
else
  repeat
    Input: $\boldsymbol{\theta}$, $\tilde{\boldsymbol{x}}_k^{ex}$, $\boldsymbol{x}_{k|k}^{ex}$, $\boldsymbol{x}_{k|k}^{in}$, $\boldsymbol{F}(\cdot)$, $\boldsymbol{G}(\cdot)$, $\boldsymbol{M}(\cdot)$;
    Execute $\tilde{\boldsymbol{x}}_1^{ex} \xrightarrow{\boldsymbol{F}(\cdot),\tilde{\boldsymbol{x}}_{k|k}^{in}} \boldsymbol{x}_k^{ex}$ **Eq.**(13);
    Execute $\min_{\boldsymbol{\theta}\in\mathbb{R}} \sum_{k=1}^{n} L(\boldsymbol{x}_k^{ex}, \boldsymbol{x}_{k|k}^{ex})$ **Eq.**(14,7);
    Output neural function $\boldsymbol{F}(\cdot)$ **Eq.**(13);
    Estimate $\boldsymbol{x}_{k|k}^{ex}$, $\boldsymbol{x}_{k|k}^{in}$ **Eq.**(5-12);
  until $\boldsymbol{x}_{k|k}^{in}$ *remain unchanged*;
end
Result: $\boldsymbol{x}_{k|k}^{ex}$, $\boldsymbol{x}_{k|k}^{in}$, $\boldsymbol{F}(\cdot)$;

Again, continuous backpropagation is applied for optimizing (14). Once the training converges, the corresponding ODE-Net can be integrated into (9) for dynamic state estimation.

B. Procedure of Neuro-DSE$^+$ Algorithm

Algorithm 1 summarizes the *Neuro-DSE*$^+$ algorithm. Three kernel steps are incorporated:

- *Step 1*: (Pre-training of ODE-Net) *Neuro-DSE*$^+$ initializes an augmented ODE-Net following (13). *Neuro-DSE* pre-trains an ODE-Net based on (4) directly using the measurement data and estimates the InSys states accordingly.
- *Step 2*: (Self-refined training of ODE-Net) *Neuro-DSE*$^+$ performs training based on (14) using the estimated InSys states. Once the ODE-Net converges, go to *Step 3*.
- *Step 3*: (InSys states updating) InSys states are re-estimated using the up-to-date augmented ODE-Net following the neural-incorporated prediction/correction presented in (5) and (11). If InSys states remain unchanged, the algorithm terminates, outputting the ODE-Net and the corresponding state estimation results; otherwise, go to *Step 2*.

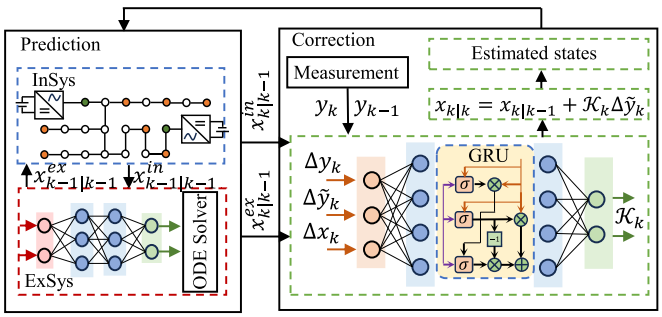

Fig. 3. *Neuro-KalmanNet-DSE* architecture.

Neuro-DSE$^+$ filters the noise-contained measurements and augments the unmeasured states to construct the training data for the ODE-Net model. Such a process is particularly beneficial for data-driven DSE under limited and very noisy observations, as it proactively employs the DSE physics of the NMs to refine the measurements and adjust the neural network, rather than merely relying on the observable data.

IV. *Neuro-KalmanNet-DSE*: KalmanNet-Enhanced Neuro-Dynamic State Estimation

Generally, *Neuro-DSE* and *Neuro-DSE*$^+$ are both integrated with classical Kalman filters. However, possible model mismatches may happen when imperfect neural networks are trained from finite samples or inaccurate system parameters exist. The efficacy of traditional Kalman filters, which depend on precise understanding and modeling of underlying dynamics, may be significantly influenced by the accuracy of domain knowledge and model assumptions [26]. Therefore, in such scenarios, the performance of conventional Kalman filter-integrated Neuro-DSE may be affected. To this end, this section further develops *Neuro-KalmanNet-DSE*, which empowers the *Neuro-DSE* philosophy with KalmanNet to achieve an adaptive state estimation under inaccurate NMs models.

A. KalmanNet Architecture for Neuro-DSE

The Kalman gain matrix $\boldsymbol{K}$ plays an important role in classical Kalman filters (see (3)), which uses system models to compute the impact of the noisy measurements on the estimated states. The keystone of KalmanNet is to replace the traditional model-based Kalman gain with a data-driven Kalman gain to relieve the impact of inaccurate models.

Inspired by (12), KalmanNet takes three input features to describe the mapping between the elements of Kalman gain matrix and system states/measurements: (i) the observation difference $\Delta \boldsymbol{y}_k = \boldsymbol{y}_k - \boldsymbol{y}_{k-1}$; (ii) the correction difference $\Delta \tilde{\boldsymbol{y}}_k = \tilde{\boldsymbol{y}}_k - \boldsymbol{M}(\boldsymbol{x}^{ex}_{k|k-1}, \boldsymbol{x}^{in}_{k|k-1})$ and (iii) the prediction difference $\Delta \boldsymbol{x}_k = \boldsymbol{x}_{k|k} - \boldsymbol{x}_{k|k-1}$, where $\boldsymbol{x}$ consists of $\boldsymbol{x}^{ex}$ and $\boldsymbol{x}^{in}$. Meanwhile, as shown in (12), the computation of Kalman gain follows a recursive nature, which motivates designing KalmanNet as an internal memory element (e.g., RNN) to describe the Kalman gain (See Fig. 3). As a consequence, the KalmanNet for *Neuro-DSE* is formulated as:

$$\mathcal{K}_k = \mathcal{K}_{RNN}(\Delta \boldsymbol{y}_k, \Delta \tilde{\boldsymbol{y}}_k, \Delta \boldsymbol{x}_k, \boldsymbol{\varphi}) \tag{15}$$

where $\mathcal{K}_{RNN}$ represents the function of RNN; $\boldsymbol{\varphi}$ is the trainable parameters in the KalmanNet. A noticeable feature of (15) is that the learning-based Kalman gain is adjusted based on the latent information of the measurement and estimation data to relieve model mismatches during its training process.

Accordingly, *Neuro-DSE* can be performed by replacing the model-driven $\boldsymbol{K}_k$ with the data-driven $\mathcal{K}_k$:

$$\boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1} + \mathcal{K}_k \cdot \Delta \tilde{\boldsymbol{y}}_k \tag{16}$$

B. KalmanNet Training Algorithm

KalmanNet is trained in a supervised fashion [26]. To best correct the dynamic state behavior with the adaptive learning-based Kalman gain, the loss function is set as the minimization error between the train samples $\tilde{\boldsymbol{x}}_k$ and the estimated state $\boldsymbol{x}_{k|k}$ produced by the KalmanNet:

$$\min_{\boldsymbol{\varphi} \in \mathbb{R}} L(\boldsymbol{\varphi}) = \sum_{k=1}^{n} \frac{1}{n} (\boldsymbol{x}_{k|k} - \tilde{\boldsymbol{x}}_k)^2 + \gamma \cdot \boldsymbol{\varphi}^2$$
$$\text{s.t.} \quad \boldsymbol{x}_{k|k} = \boldsymbol{x}_{k|k-1} + \mathcal{K}_k \cdot \Delta \tilde{\boldsymbol{y}}_k \tag{17}$$

where γ presents the regularization coefficient.

The loss gradient with respect to Kalman gain can be devised from the output of KalmanNet as:

$$\frac{\partial L(\boldsymbol{\varphi})}{\partial \mathcal{K}_k} = \frac{\partial ||\mathcal{K}_k \Delta \boldsymbol{y}_k - \Delta \boldsymbol{x}_k||_2}{\partial \mathcal{K}_k}$$
$$= 2\mathcal{K}_k \Delta \tilde{\boldsymbol{y}}_k \cdot (\Delta \tilde{\boldsymbol{y}}_k + \boldsymbol{x}_{k|k-1} - \tilde{\boldsymbol{x}}_k) \tag{18}$$

Algorithm 2 provides the overall procedure of *Neuro-KalmanNet-DSE*.

V. Numerical Tests

This section validates the *Neuro-DSE* method and its variants in a typical islanded networked microgrids system. All codes are implemented in MATLAB 2020a and Python 3.7 on a 2.50 GHz PC.

A. Test System and Algorithm Settings

The test system is a phasor-model-based, 4-microgrid NMs in islanded mode (see Fig. 4). Five inverters are connected to the NMs [3] in Fig. 4(a). System parameters are presented in Appendix Part A, including the parameters of DERs, power loads, and composition information of the microgrid system. Each controller (see Fig. 4(b)[3]) comprises three different parts. The first part is a power controller that adopts droop control or secondary control for power regulation effects. The second and third components of the control system encompass the voltage and current controllers, respectively. These controllers are used

[3]In the controller block, v and i denote voltage and current outputs; P and Q denote active and reactive power generations; superscript * denotes reference value; subscripts $_l$ and $_o$ denote before and after filters.

Algorithm 2: *Neuro-KalmanNet-DSE.*

Initialize: $\boldsymbol{\theta}$, $\tilde{\boldsymbol{x}}_k^{ex}$, $\tilde{\boldsymbol{u}}_k^{in}$, $\boldsymbol{F}(\cdot)$, $\boldsymbol{G}(\cdot)$, $\boldsymbol{M}(\cdot)$, $\mathcal{K}_{RNN}(\cdot)$, k;
if *Pre-training* **then**
  ▷ **ODE-Net pre-training**:
  Execute $\tilde{\boldsymbol{x}}_1^{ex} \xrightarrow{\boldsymbol{F}(\cdot),\tilde{\boldsymbol{u}}_k^{in}} \boldsymbol{x}_k^{ex}$ **Eq.**(4);
  Execute $\min_{\boldsymbol{\theta}\in\mathbb{R}} \sum_{k=1}^{n} L(\boldsymbol{x}_k^{ex})$ **Eq.**(6,7);
  Estimate $\boldsymbol{x}_{k|k-1}$, $\boldsymbol{x}_{k|k}$ **Eq.**(5-12);
  ▷ **KalmanNet pre-training**:
  Execute $\Delta\boldsymbol{y}_k, \Delta\hat{\boldsymbol{y}}_k, \Delta\boldsymbol{x}_k \xrightarrow{\mathcal{K}_{RNN}(\cdot)} \mathcal{K}_k$ **Eq.**(15);
  Execute $\min_{\boldsymbol{\varphi}\in\mathbb{R}} \sum_{k=1}^{n} L(\boldsymbol{x}_{k|k})$ **Eq.**(16,17);
  Estimate $\boldsymbol{x}_{k|k}$ **Eq.**(16);
else
  ▷ **Alternating training of ODE-Net and KalmanNet** :
  repeat
    Execute $\tilde{\boldsymbol{x}}_1^{ex} \xrightarrow{\boldsymbol{F}(\cdot),\boldsymbol{x}_{k|k}^{in}} \boldsymbol{x}_k^{ex}$ **Eq.**(13);
    Execute $\min_{\boldsymbol{\theta}\in\mathbb{R}} \sum_{k=1}^{n} L(\boldsymbol{x}_k^{ex})$ **Eq.**(14,7);
    Estimate $\boldsymbol{x}_{k|k-1}$, $\boldsymbol{x}_{k|k}$ **Eq.**(5-12);
    Execute $\Delta\boldsymbol{y}_k, \Delta\hat{\boldsymbol{y}}_k, \Delta\boldsymbol{x}_k \xrightarrow{\mathcal{K}_{RNN}(\cdot)} \mathcal{K}_k$ **Eq.**(15);
    Execute $\min_{\boldsymbol{\varphi}\in\mathbb{R}} \sum_{k=1}^{n} L(\boldsymbol{x}_{k|k})$ **Eq.**(16,17);
    Output $\boldsymbol{x}_{k|k}$, $\boldsymbol{F}(\cdot)$,$\mathcal{K}_{RNN}(\cdot)$ **Eq.**(13);
  until $\boldsymbol{x}_{k|k}$ *remain unchanged*;
end

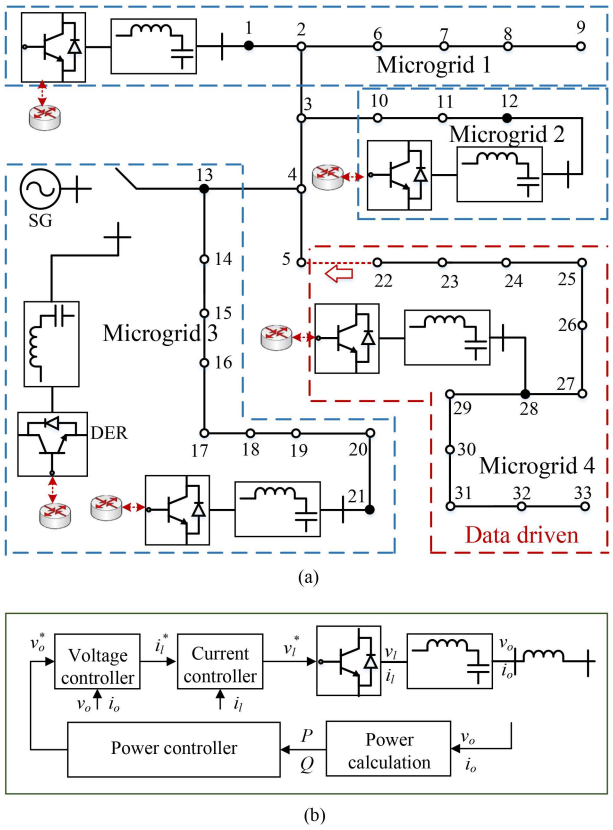

Fig. 4. Test system: 4-microgrid networked microgrids with 5 grid-forming DERs. (a) Topology of the test system; (b) Controller of DERs.

to effectively mitigate high-frequency disturbances and ensure adequate damping for the output filter [27], [28], [29]. The majority of dynamic models of the test system are presented in Appendix Part B, including the power controller, power loads, and branches. The detailed control diagrams and dynamic models for the microgrid system in Fig. 4 can be found in [25], [27]. In addition, bus 13 can be connected to different types of power sources, including droop/secondary-control-based DER, virtual synchronous generator (VSG) [30] and synchronous generator (SG) [23], to study the performance of *Neuro-DSE* under different power mixes.

We assume microgrid 4 is the ExSys without explicit physics knowledge and will be formulated via a learning-based fashion. The corresponding ODE-Net adopts a two-layer perceptron architecture, with 40 neurons in each layer. The regularization coefficient is set as 0.5. Training data for ODE-Net is generated by time-domain simulations under 20% uncertainties of the renewable energy inputs. In this work, branch current measurements are used for *Neuro-DSE*, while the internal signals of inverter controllers are assumed inaccessible. However, the method is compatible with other types of measurements. Specifically, for secondary control, the communication signals between DERs (i.e., the frequency/voltage secondary control signals $\boldsymbol{\Omega}$ and $\boldsymbol{e}$) are also attainable as they are explicitly measured by each DER for the control purpose.

The default case is that all the DERs adopt droop controls (including bus 13) and all the branch currents in InSys are measurable. It is assumed that both process and measurement noises of the NMs follow Gaussian distributions $N(\mu, \sigma^2)$ [10]. In the following, various cases are studied, including different DER control strategies, power source mixes, noise levels, and measurement availability, to thoroughly verify the efficacy of *Neuro-DSE* and its variants.

B. Validity of Neuro-DSE

This subsection validates the effectiveness of the devised *Neuro-DSE* under various circumstances.

1) Neuro-DSE Under Different Noise Levels: We study the performance of *Neuro-DSE* under different noise levels. Two scenarios are considered: (a) measurement noise as $N(0, e^{-6})$ and process noise as $N(0, e^{-6})$; (b) measurement noise increased to $N(0, e^{-4})$ and process noise as $N(0, e^{-6})$[10]. Fig. 5 presents the simulation results. The following insights can be obtained:

- *Neuro-DSE* can track both measurable states (e.g., currents in Fig. 5(a-1) and (b-1)) and unmearsurable states (e.g., internal control signals of inverters in Fig. 5(a-2) and (b-2)) of NMs.
- Under both large noises (Fig. 5(b)) and small noises (Fig. 5(a)), the dynamic states estimated from *Neuro-DSE* are close to the true values.
- *Neuro-DSE* exhibits powerful compatibility to different Kalman filters. The estimation results from both EKF-based and UKF-based *Neuro-DSE* are close to the true states under different noise levels as shown in Fig. 5.

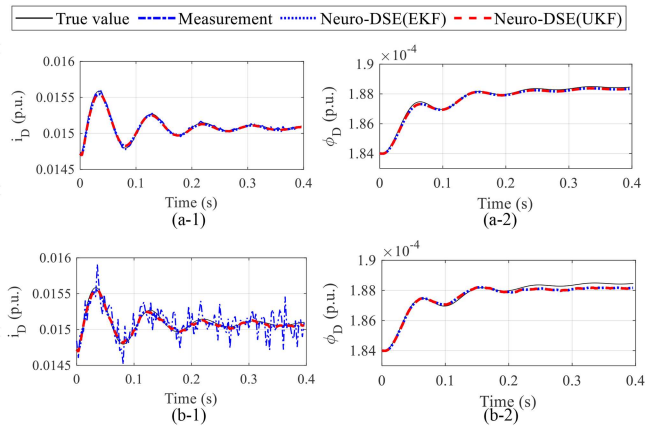

Fig. 5. Selected states at D-axis of *Neuro-DSE* under different noise levels. (a-1) Current under $N(0, e^{-6})$, (a-2) Voltage control signal under $N(0, e^{-6})$; (b-1) Current under $N(0, e^{-4})$, and (b-2) Voltage control signal under $N(0, e^{-4})$.

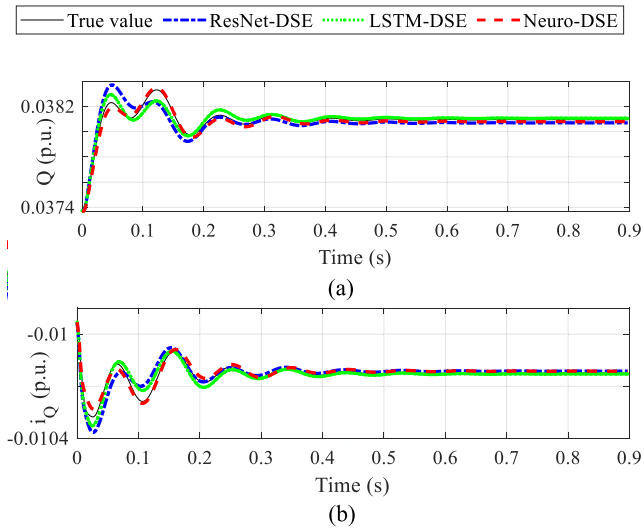

Fig. 6. State trajectories of DER 1 under different neural networks. (a) Reactive power generation; (b) Current output at D-axis.

- Although the results are generally satisfactory, we still observe a small bias in the estimated states under large noises (for example, Fig. 5(b-2)). This is exactly the motivation for developing *Neuro-DSE*$^{+}$, which will be discussed in Part C.

2) Comparison With Conventional DNN-Based DSE: We then compare *Neuro-DSE* with conventional DNN-based DSE to illustrate the superiority of the devised method. Two representative DNNs are studied: (a) a residual neural network (ResNet) comprised of 8 hidden layers with double-layer skips and 100 hidden units in each layer; (b) a long short-term memory (LSTM) network with 100 hidden units.

Fig. 6 clearly illustrates that the devised ODE-Net-based *Neuro-DSE* outperforms the conventional DNN-based DSE methods. As shown in the figure, ResNet-based DSE shows large differences at the starting stage; LSTM-based DSE tends to have

TABLE I
MSE OF DSE WITH DIFFERENT NEURAL NETWORKS

Noise	Type	$\boldsymbol{P}$	$\boldsymbol{Q}$	$\boldsymbol{i}_D$	$\boldsymbol{i}_Q$	$\boldsymbol{i}_D^{ex}$	$\boldsymbol{i}_Q^{ex}$
$1e^{-6}$	ODENet	$2.70e^{-5}$	$2.67e^{-5}$	$7.44e^{-6}$	$6.75e^{-6}$	$1.13e^{-4}$	$1.83e^{-5}$
	ResNet	$1.78e^{-4}$	$4.10e^{-5}$	$2.34e^{-5}$	$7.19e^{-6}$	$3.32e^{-4}$	$8.68e^{-5}$
	LSTM	$6.71e^{-5}$	$4.53e^{-5}$	$1.24e^{-5}$	$7.19e^{-6}$	$1.63e^{-4}$	$1.01e^{-5}$
$1e^{-4}$	ODENet	$4.23e^{-5}$	$1.46e^{-4}$	$8.72e^{-6}$	$7.92e^{-6}$	$1.52e^{-4}$	$1.25e^{-4}$
	ResNet	$3.78e^{-4}$	$1.17e^{-3}$	$8.12e^{-5}$	$1.68e^{-4}$	$1.15e^{-3}$	$2.78e^{-3}$
	LSTM	$3.67e^{-4}$	$2.39e^{-4}$	$5.58e^{-5}$	$2.90e^{-5}$	$1.10e^{-3}$	$4.78e^{-4}$

* NDSE:*Neuro-DSE*

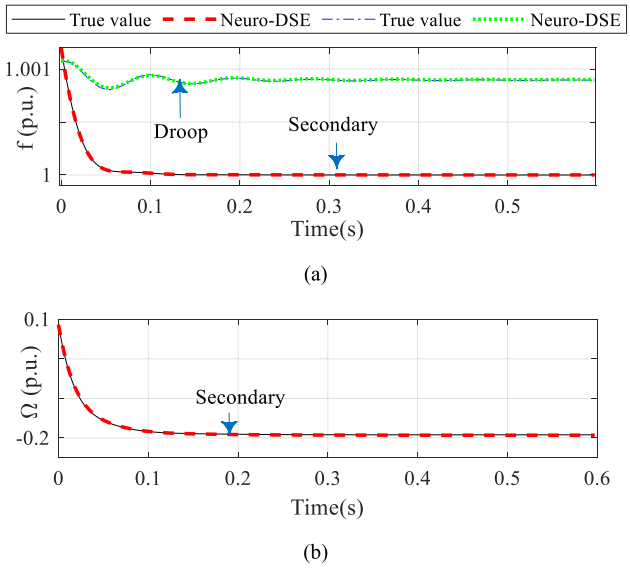

Fig. 7. State trajectories of DER 1 under droop/secondary controls. (a) Frequency; (b) Frequency control signal.

slight biases for the steady-state; and only *Neuro-DSE* provides accurate estimation during the whole time period.

Table I further presents the maximum square errors (MSE) of different methods under different noise levels. It is again observed that *Neuro-DSE* presents the best performance compared with other DNN-based DSE methods. More importantly, with the increase of the noise level, the MSE of *Neuro-DSE* does not increase much while the performance of ResNet and LSTM sharply deteriorates.

3) Neuro-DSE Under Different DER Control Modes: Fig. 7 further validates the effectiveness of *Neuro-DSE* under different control modes of inverters. Both droop control and secondary control are studied. Fig. 7(a) shows that compared with droop control, secondary control can recover the system frequency to the nominal value, and *Neuro-DSE* is capable of tracking the system dynamics under different control modes. Furthermore, the devised *Neuro-DSE* method also shows the satisfactory ability for tracking communication signals, such as the frequency control signal Ω among DERs presented in Fig. 7(b).

4) Neuro-DSE Under Different NMs Compositions: Finally, we present *Neuro-DSE*'s powerful universality under different NMs compositions. Besides the droop/secondary-controlled DERs, two additional power sources are studied, i.e., VSG, and SG. Fig. 8 presents the simulation results. It can be observed that

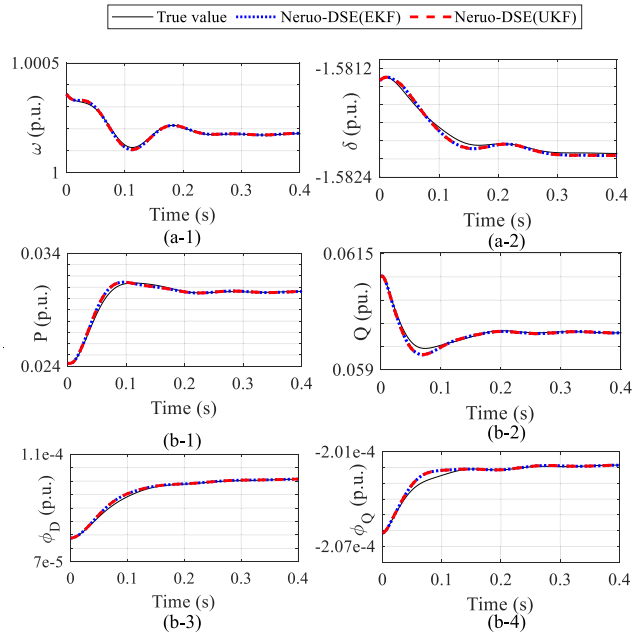

Fig. 8. Selected states of SG and VSG under *Neuro-DSE*. (a-1) Speed of SG, (a-2) Rotor angle of SG; (b-1) Active power of VSG, (b-2) Reactive power of VSG, (b-3) Voltage controller signals at D axis, (b-4) Voltage controller signals at Q axis.

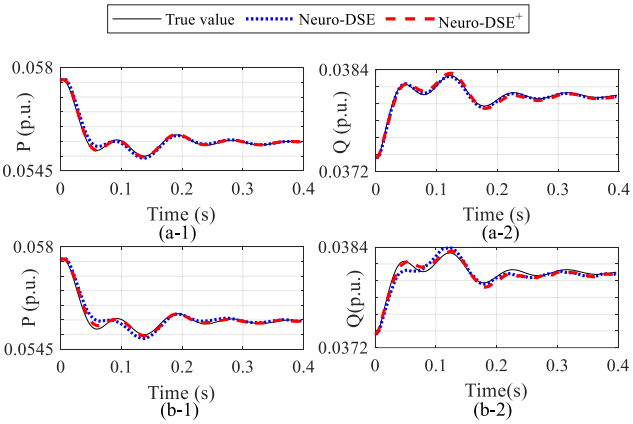

Fig. 9. Power states of DER 1 under different noise levels. (a-1) Active power under $N(0, e^{-6})$, (a-2) Reactive power under $N(0, e^{-6})$; (b-1) Active power under $N(0, e^{-4})$, (b-2) Reactive power under $N(0, e^{-4})$.

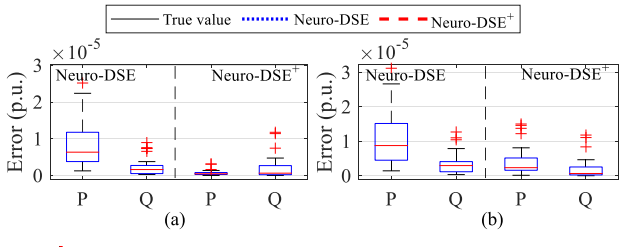

Fig. 10. Differences of power states of DER 1 under different noise levels. (a) States under $N(0, e^{-6})$; (b) States under $N(0, e^{-4})$.

Neuro-DSE maintains high accuracy for both traditional synchronous generators and inverter-interfaced virtual synchronous generators, and again exhibits satisfactory estimation performance for tracking the internal controller signals of VSG (see Fig. 8(b-3) and (b-4)).

C. Efficacy of Neuro-DSE$^+$

This subsection verifies the effectiveness of *Neuro-DSE*$^+$ under poor measurements. As we introduced, excessively noisy and limited measurements may cause biases of *Neuro-DSE*, which motivates us to devise a self-refined *Neuro-DSE*$^+$ to improve the estimation performance.

1) Neuro-DSE$^+$ *Under Different Noise Levels:* First, we demonstrate the efficacy of *Neuro-DSE*$^+$ under different noise levels. Figs. 9 and 10 compare the performance of *Neuro-DSE* and *Neuro-DSE*$^+$ under two measurement noises N(0,e^{-6}) and N(0,e^{-4}) to illustrate the superiority of *Neuro-DSE*$^+$. It can be observed that:

- As shown in Fig. 9, *Neuro-DSE*$^+$ obtains more accurate state estimation results than *Neuro-DSE* under noisy measurement, evidenced by the fact that the estimated states of *Neuro-DSE*$^+$ under different noise levels are always closer to the real states than that of *Neuro-DSE*. For example, in Fig. 9(b), when the noise has a distribution of N(0,e^{-4}), the estimated $\boldsymbol{P}$ from *Neuro-DSE* has larger deviations to the true value, whereas the result from *Neuro-DSE*$^+$ remains accurate. This verifies the powerful tracking ability of *Neuro-DSE*$^+$ under high noise level.
- Fig. 10 further quantitatively studies the performance of *Neuro-DSE*$^+$. Box-plots of the estimation error under 40 random noisy scenarios are provided. It is obvious that *Neuro-DSE*$^+$ outperforms *Neuro-DSE* in terms of robustness against noises, as the interquartile range of the estimation error of *Neuro-DSE*$^+$ is significantly smaller that that of *Neuro-DSE*.

2) Neuro-DSE$^+$ *Under Different Measurement Availability:* Second, we demonstrate the performance of *Neuro-DSE*$^+$ under different availability levels of the measurement data. Fig. 11 presents the DSE results when 100%, 80% and 70% of branch currents are respectively measured. Numerical experiments show that *Neuro-DSE*$^+$ is capable of providing more accurate estimation results than *Neuro-DSE* especially under limited measurements. For example, under 70% measurements, the estimated reactive power state $\boldsymbol{Q}$ of *Neuro-DSE* has obvious deviations after 0.25 s while *Neuro-DSE*$^+$ consistently tracks the true values of the NMs. This is because *Neuro-DSE*$^+$ automatically supplements the unmeasurable states and uses them as inputs to the ODE-Net model, which significantly enrich the training information.

Table II presents the MSE performance of *Neuro-DSE* and *Neuro-DSE*$^+$ under 40 random noisy scenarios. It can be seen that *Neuro-DSE*$^+$ exhibits satisfatory performance in estimating the internal controller signals. For example, the MSE of voltage controller signal $\boldsymbol{\phi}_Q$ of *Neuro-DSE*$^+$ under 70% measurement

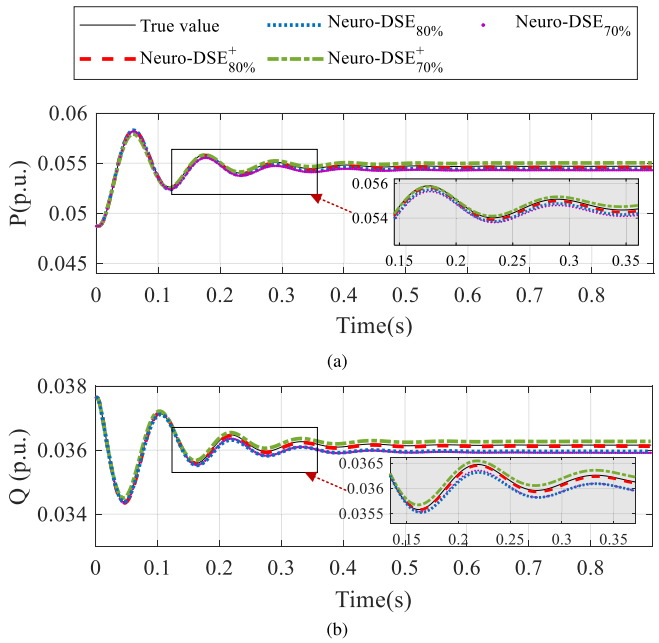

Fig. 11. State trajectories of DER 2 under different measurement levels. (a) Active power generation; (b) Reactive power generation.

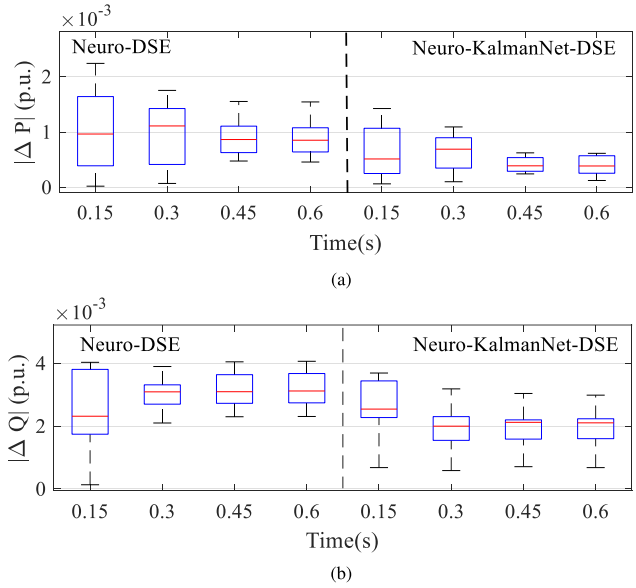

Fig. 12. State differences of DER1 with different *Neuro-DSE*. (a) Differences of active power at different time slots; (b) Differences of reactive power at different time slots.

TABLE II
MSE OF NEURO-DSEs WITH DIFFERENT MEASUREMENT LEVELS

meas	Type	$\boldsymbol{P}$	$\boldsymbol{Q}$	$\boldsymbol{\phi}_D$	$\boldsymbol{\phi}_Q$	$\boldsymbol{i}_D$	$\boldsymbol{i}_Q$
70%	NDSE	$4.13e^{-4}$	$1.42e^{-4}$	$2.24e^{-9}$	$9.01e^{-10}$	$5.53e^{-5}$	$1.80e^{-5}$
	NDSE$^+$	$2.04e^{-4}$	$5.55e^{-5}$	$1.86e^{-9}$	$4.29e^{-10}$	$2.36e^{-5}$	$5.83e^{-6}$
80%	NDSE	$3.38e^{-4}$	$6.89e^{-5}$	$2.09e^{-9}$	$4.21e^{-10}$	$3.93e^{-5}$	$1.06e^{-5}$
	NDSE$^+$	$1.18e^{-4}$	$2.13e^{-5}$	$3.90e^{-10}$	$2.35e^{-10}$	$4.22e^{-6}$	$1.98e^{-6}$
100%	NDSE	$3.01e^{-4}$	$6.35e^{-5}$	$2.02e^{-9}$	$4.01e^{-10}$	$2.45e^{-5}$	$8.62e^{-5}$
	NDSE$^+$	$5.56e^{-5}$	$2.03e^{-5}$	$3.06e^{-10}$	$2.65e^{-10}$	$3.56e^{-6}$	$3.33e^{-6}$

* Note: NDSE: *Neuro-DSE*; NDSE$^+$:*Neuro-DSE*$^+$.

level is $4.29e^{-10}$ which is 52.4% lower than that of *Neuro-DSE* which is $9.01e^{-10}$.

D. Efficacy of Neuro-KalmanNet-DSE Addressing Model Mismatch

This subsection validates the effectiveness of *Neuro-KalmanNet-DSE* to relieve the impacts of model mismatches. Neural models can induce potential model mismatches as it is learned from a finite set of training samples. The physics models may also be inaccurate, because of the inexact system parameters, etc. To examine such effect, we generate measurement data by increasing the branch resistance of the test system by 50%, but using the original parameters for DSE algorithms. Specifically, the KalmanNet is constructed as an RNN with a fully connected input layer, a single GRU layer [26], and a fully connected output layer, to learn the Kalman gain.

Fig. 12 compares the errors of *Neuro-DSE* and *Neuro-KalmanNet-DSE* by examining 40 scenarios under randomly-generated noisy measurements. The results illustrate that *Neuro-KalmanNet-DSE* is more immune to model mismatches, as compared to *Neuro-DSE*. As shown in Fig. 12(a), the active power errors of *Neuro-KalmanNet-DSE* at 0.3 s are much smaller than those of *Neuro-DSE*. By examine the errors in reactive power estimates in Fig. 12(b), one can lead to the same conclusion.

VI. CONCLUSION

This paper devises *Neuro-DSE*, an innovative approach for data-driven Dynamic State Estimation in Networked Microgrids under undefined dynamic models of inverters and subsystems. Two variants (i.e., *Neuro-DSE*$^+$ and *Neuro-KalmanNet-DSE*) are established to enhance the performance of *Neuro-DSE* under noisy limited measurement and model mismatch scenarios. Case studies on standard 4-microgrid NMs demonstrate the accuracy and noise resilience of *Neuro-DSE* under varying noise levels. It illustrates *Neuro-DSE*$^+$ achieves up to a 50% increase in state estimation accuracy compared to the base *Neuro-DSE* under limited measurement conditions. Meanwhile, *Neuro-KalmanNet-DSE* improves deviations of *Neuro-DSE* under the model mismatch scenario. The devised method offers an inspiring tool for energy management and real-time operation of NMs.

APPENDIX

A. Parameter Setting of the Test System

This appendix offers the parameters of the test system as shown in Fig. 4, including the parameters of DERs (see Table III), and the parameters of power loads (see Table IV). The composition information of the test system can be found in [25]. Without loss of generality, the impedance of each branch is assumed homogeneous with $r_b = 7.79 \times 10^{-4} p.u.$, $L_b = 6.24 \times 10^{-3} p.u.$.

TABLE III
PARAMETERS OF DERS

DER ID.	1	2	3	4	5
Bus No.	1	12	13	21	28
Control parameters	$m_p(p.u.)$	$n_q(p.u.)$	$r_o(p.u.)$	$L_o(p.u.)$	F
	41.22	1.5	0.2	3.2×10^{-3}	0.75
	k_{pv}	k_{iv}	k_{pc}	k_{ic}	
	2	20	105	1600	
Secondary /VSG control	α	A	β_{ld}/β_{fl}	B_{ld}/B_{fl}	H
	50	5	100/400	0/1	28.15

* Note: r_o: coupling resistance; L_o: coupling inductance; k_{pv}: voltage proportional gain; k_{iv}: voltage integral gain; k_{pc}: current proportional gain; k_{ic}: current integral gain; F: current feedforward gain; subscripts $_{ld}$ and $_{fl}$: secondary control parameters for leader DER and follower DERs [3].

TABLE IV
PARAMETERS OF POWER LOADS

Load ID.	Bus No.	r_l (p.u.)	L_l (p.u.)	Load ID.	Bus No.	r_l (p.u.)	L_l (p.u.)
1	2	3.32	2.33	15	33	3.68	2.57
2	3	3.67	2.57	16	6	3.69	2.58
3	4	2.75	1.92	17	7	3.66	2.56
4	5	5.49	3.84	18	8	3.66	2.56
5	22	1.64	1.15	19	9	3.65	2.55
6	23	1.64	1.15	20	10	3.66	2.56
7	24	5.50	3.85	21	11	0.78	0.54
8	25	5.52	3.86	22	14	5.48	3.83
9	26	7.38	5.16	23	15	5.47	3.82
10	27	5.54	3.88	24	16	5.42	3.79
11	29	2.74	1.91	25	17	2.70	1.89
12	30	4.89	4.08	26	18	0.24	0.72
13	31	5.55	3.89	27	19	2.17	1.51
14	32	5.53	3.87	28	20	1.55	1.08

B. Dynamic Model of the Test System

This appendix introduces the dynamic model of the test system including DERs, loads and branches. Dynamic formulations of voltage and current controllers of DERs are detailed in [25]. Specifically, different control strategies in the power controller are considered in this paper:

1) Droop Control: This control is used to realize power-sharing effects. The dynamic model is established as:

$$\begin{cases} \boldsymbol{\omega} = \omega^* - \boldsymbol{m}_p(\boldsymbol{P} - \boldsymbol{P}^*) \\ \boldsymbol{E} = \boldsymbol{E}^* - \boldsymbol{n}_q(\boldsymbol{Q} - \boldsymbol{Q}^*) \end{cases} \tag{19}$$

where, $\boldsymbol{\omega}, \boldsymbol{E}, \boldsymbol{P}$ and $\boldsymbol{Q}$ represent the angular speeds, voltage magnitudes, active and reactive power outputs of DERs, respectively; ω^*, $\boldsymbol{E}^*$, $\boldsymbol{P}^*$ and $\boldsymbol{Q}^*$ denote the corresponding nominal values; $\boldsymbol{m}_p$ and $\boldsymbol{n}_q$ denotes the active/reactive power droop coefficients.

2) Secondary Control: The target is to achieve voltage regulation and frequency recovery based on distributed averaging, the dynamic model [3] is expressed as:

$$\begin{cases} \dfrac{d\boldsymbol{\Omega}}{dt} = -\boldsymbol{\alpha}(\boldsymbol{\omega} - \omega^*) - \boldsymbol{A}\boldsymbol{\Omega} \\ \dfrac{d\boldsymbol{e}}{dt} = -\boldsymbol{\beta}(\boldsymbol{E} - \boldsymbol{E}^*) - \boldsymbol{B}\boldsymbol{Q} \end{cases} \tag{20}$$

where, $\boldsymbol{\Omega}$ and $\boldsymbol{e}$ respectively denote the secondary control signals of all DERs corresponding to frequency and voltage regulations; $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{A}$ and $\boldsymbol{B}$ denote control parameters [25].

3) VSG Control: It aims to improve the inertia. The dynamic model [30] is established as follows:

$$\frac{d\boldsymbol{\omega}}{dt} = \frac{1}{2\boldsymbol{H}\boldsymbol{\omega}}\left(\boldsymbol{P}^* - \boldsymbol{P} + \frac{1}{\boldsymbol{m}_p}(\omega^* - \boldsymbol{\omega})\right) \tag{21}$$

where, $\boldsymbol{H}$ denotes the inertia constant.

Dynamic models of power loads and branches are established as [25]:

$$\boldsymbol{L}_l\frac{d\boldsymbol{i}_l}{dt} = -\boldsymbol{r}_l\boldsymbol{i}_l + \omega_s\boldsymbol{I}_s\boldsymbol{L}_l\boldsymbol{i}_l + \boldsymbol{M}_l\boldsymbol{E} \tag{22}$$

$$\boldsymbol{L}_b\frac{d\boldsymbol{i}_b}{dt} = -\boldsymbol{r}_b\boldsymbol{i}_b + \omega_s\boldsymbol{I}_s\boldsymbol{L}_b\boldsymbol{i}_b + \boldsymbol{M}_b\boldsymbol{E} \tag{23}$$

where, $\boldsymbol{i}_l$ and $\boldsymbol{i}_b$ denote the currents of power loads and branches at DQ-axis, respectively; $\boldsymbol{r}_l$ and $\boldsymbol{L}_l$ respectively denote the resistance and inductance matrices of loads; $\boldsymbol{M}_l$ is the incidence matrix between buses and loads; $\boldsymbol{r}_b$, $\boldsymbol{L}_b$ and $\boldsymbol{M}_b$ respectively present corresponding definitions for branches similarly. The detailed system model can be found in [25].

REFERENCES

[1] M. N. Alam, S. Chakrabarti, and A. Ghosh, "Networked microgrids: State-of-the-art and future perspectives," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1238–1250, Mar. 2019.

[2] Y. Zhou and P. Zhang, "Reachable power flow: Theory to practice," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2532–2541, May 2021.

[3] 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.

[4] F. Feng, P. Zhang, Y. Zhou, and L. Wang, "Distributed networked microgrids power flow," *IEEE Trans. Power Syst.*, vol. 38, no. 2, pp. 1405–1419, Mar. 2022.

[5] L. Wang, Y. Qin, Z. Tang, and P. Zhang, "Software-defined microgrid control: The genesis of decoupled cyber-physical microgrids," *IEEE Open Access J. Power Energy*, vol. 7, pp. 173–182, 2020.

[6] W. Wan, P. Zhang, M. A. Bragin, and P. B. Luh, "Cooperative fault management for resilient integration of renewable energy," *Electric Power Syst. Res.*, vol. 211, 2022, Art. no. 108147.

[7] Z. Bie, P. Zhang, G. Li, B. Hua, M. Meehan, and X. Wang, "Reliability evaluation of active distribution systems including microgrids," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2342–2350, Nov. 2012.

[8] G. Valverde and V. Terzija, "Unscented kalman filter for power system dynamic state estimation," *IET Gener., Transmiss. Distrib.*, vol. 5, no. 1, pp. 29–37, 2011.

[9] F. Feng, P. Zhang, and Y. Zhou, "Authentic microgrid state estimation," *IEEE Trans. Power Syst.*, vol. 37, no. 2, pp. 1657–1660, Mar. 2022.

[10] E. Ghahremani and I. Kamwa, "Dynamic state estimation in power system by applying the extended Kalman filter with unknown inputs to phasor measurements," *IEEE Trans. Power Syst.*, vol. 26, no. 4, pp. 2556–2566, Nov. 2011.

[11] H. Karimipour and V. Dinavahi, "Extended Kalman filter-based parallel dynamic state estimation," *IEEE Trans. Smart Grid*, vol. 6, no. 3, pp. 1539–1549, May 2015.

[12] M. Huang, W. Li, and W. Yan, "Estimating parameters of synchronous generators using square-root unscented Kalman filter," *Electric Power Syst. Res.*, vol. 80, no. 9, pp. 1137–1144, 2010.

[13] A. K. Singh and B. C. Pal, "Decentralized dynamic state estimation in power systems using unscented transformation," *IEEE Trans. Power Syst.*, vol. 29, no. 2, pp. 794–804, Mar. 2014.

[14] C. Carquex, C. Rosenberg, and K. Bhattacharya, "State estimation in power distribution systems based on ensemble Kalman filtering," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6600–6610, Nov. 2018.

[15] N. Zhou, D. Meng, Z. Huang, and G. Welch, "Dynamic state estimation of a synchronous machine using PMU data: A comparative study," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 450–460, Jan. 2015.

[16] G. P. Prajapat, N. Senroy, and I. Kar, "Estimation based enhanced maximum energy extraction scheme for DFIG-wind turbine systems," *Sustain. Energy, Grids Netw.*, vol. 26, 2021, Art. no. 100419. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352467720303507

[17] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Neural network aided Kalman filtering for partially known dynamics," *IEEE Trans. Signal Process.*, vol. 70, pp. 1532–1547, 2022.

[18] S. Akhlaghi, N. Zhou, and Z. Huang, "A multi-step adaptive interpolation approach to mitigating the impact of nonlinearity on dynamic state estimation," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3102–3111, Jul. 2018.

[19] F. S. Adi, Y. J. Lee, and H. Song, "State estimation for DC microgrids using modified long short-term memory networks," *Appl. Sci.*, vol. 10, no. 9, 2020, Art. no. 3028.

[20] H. Yi, Z. Zhang, and P. Wang, "State estimation of distribution network with the improved deep residual neural network," in *Proc. IEEE/IAS Ind. Commercial Power Syst. Asia*, 2022, pp. 972–977.

[21] G. Tian, Q. Zhou, R. Birari, J. Qi, and Z. Qu, "A hybrid-learning algorithm for online dynamic state estimation in multimachine power systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 12, pp. 5497–5508, Dec. 2020.

[22] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. P1–P13.

[23] Y. Zhou and P. Zhang, "Neuro-reachability of networked microgrids," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 142–152, Jan. 2022.

[24] J. Qi, K. Sun, J. Wang, and H. Liu, "Dynamic state estimation for multi-machine power system by unscented Kalman filter with enhanced numerical stability," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 1184–1196, Mar. 2018.

[25] 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 2021.

[26] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. G. van Sloun, and Y. C. Eldar, "KalmanNet: Neural network aided Kalman filtering for partially known dynamics," *IEEE Trans. Signal Process.*, vol. 70, pp. 1532–1547, 2022.

[27] 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.

[28] M. Marwali and A. Keyhani, "Control of distributed generation systems–Part I: Voltages and currents control," *IEEE Trans. Power Electron.*, vol. 19, no. 6, pp. 1541–1550, Nov. 2004.

[29] M. Prodanović, "Power quality and control aspects of parallel connected inverters in distributed generation," 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:63284059

[30] J. Liu, Y. Miura, and T. Ise, "Comparison of dynamic characteristics between virtual synchronous generator and droop control in inverter-based distributed generators," *IEEE Trans. Power Electron.*, vol. 31, no. 5, pp. 3600–3611, May 2016.