

# A Neural Lyapunov Approach to Transient Stability Assessment of Power Electronics-Interfaced Networked Microgrids

Tong Huang<sup>id</sup>, Member, IEEE, Sicun Gao, and Le Xie<sup>id</sup>, Senior Member, IEEE

**Abstract**—This paper proposes a novel Neural Lyapunov method-based transient stability assessment framework for power electronics-interfaced networked microgrids. The assessment framework aims to determine the large-signal stability of the networked microgrids and to characterize the disturbances that can be tolerated by the networked microgrids. The challenge of such assessment is how to construct a behavior-summary function for the nonlinear networked microgrids. By leveraging strong representation power of neural network, the behavior-summary function, i.e., a Neural Lyapunov function, is learned in the state space. A stability region is estimated based on the learned Neural Lyapunov function, and it is used for characterizing disturbances that the networked microgrids can tolerate. The proposed method is tested and validated in a grid-connected microgrid, three networked microgrids with mixed interface dynamics, and the IEEE 123-node feeder. Case studies suggest that the proposed method can address networked microgrids with heterogeneous interface dynamics, and in comparison with conventional methods that are based on quadratic Lyapunov functions, it can characterize the stability regions with much less conservativeness.

**Index Terms**—Networked microgrids, transient stability assessment, neural Lyapunov method, energy management system, machine learning, resilient grid.

## I. INTRODUCTION

THE PAST decade has witnessed increasing deployment of distributed energy resources (DERs) in the electric distribution grid. DERs play a crucial role of decarbonizing the energy sector and enhancing the resilience of the grid [1]. However, deepening penetration of DERs leads to unprecedented complexity for distribution system operation in monitoring, control, and protection. One promising architecture

Manuscript received March 12, 2021; revised June 25, 2021 and August 23, 2021; accepted September 23, 2021. Date of publication October 5, 2021; date of current version December 23, 2021. This work was supported in part by the U.S. Department of Energy’s Office of Energy Efficiency and Renewable Energy (EERE) through the Solar Energy Technologies Office (SETO) under Grant DE-EE0009031; in part by the National Science Foundation under Grant OAC-1934675, Grant ECCS-2035688, Grant ECCS-1611301, Grant CCF 2047034; and in part by the Air Force Office of Scientific Research Young Investigator Program (AFOSR YIP) under Grant FA9550-19-1-0041. Paper no. TSG-00386-2021. (Corresponding author: Le Xie.)

Tong Huang and Le Xie are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: tonghuang@tamu.edu; le.xie@tamu.edu).

Sicun Gao is with the Department of Computer Science and Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: sicung@ucsd.edu).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSG.2021.3117889>.

Digital Object Identifier 10.1109/TSG.2021.3117889

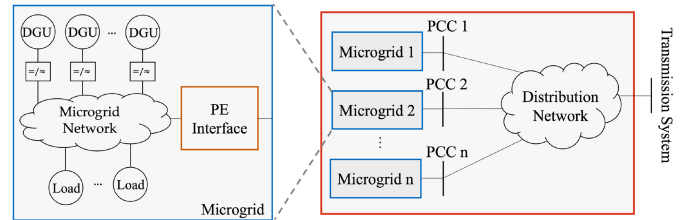


Fig. 1. A microgrid-based distribution system: inside the left blue box shows the physical structure of a microgrid.

to manage the massive integration of DERs is to reconfigure the distribution system as *power electronics-interfaced networked microgrids*<sup>1</sup> shown in Figure 1. A microgrid packages interconnected distributed generation units (DGUs) and loads which are regulated locally by the microgrid central controller (MGCC) [2]. The microgrid has a power-electronic (PE) interface [2] that physically connects to its host distribution system via a point of common coupling (PCC). Microgrids are networked with each other through PCCs and distribution lines. With such a configuration, instead of managing massive DGUs at grid edges, a distribution system operator (DSO) only needs to coordinate a few PE interfaces of microgrids [2], by which the system management complexity at the DSO level is significantly reduced.

Given the microgrid-based distribution system, a key function of its distribution management system (DMS) is to assess the physical security of networked microgrids. Functionally speaking, this task should be comprised of both static security assessment (SSA) and transient stability assessment (TSA). The SSA scrutinizes if physical variables of networked microgrids in the steady-state time scale are within predefined normal operating ranges. It is typically considered as optimization constraints when researchers develop coordination strategies of networked microgrids [3] for grid resilience enhancement and economical efficiency maximization. The TSA examines the dynamic behaviors of networked microgrids in a faster time scale. The TSA tool aims to characterize (large) disturbances that the networked microgrids can tolerate. Such characterization allows for efficient design and planning of the microgrid-based distribution

<sup>1</sup>In the sequel, “power electronics-interfaced networked microgrids” and “networked microgrids” are used interchangeably for the sake of brevity.

systems [4], and it also enables a DSO to maintain situational awareness in real-time operation [4]. This paper focuses on assessing transient stability of *networked* microgrids. Such a topic concerns the DSO, because excessive energy transactions among microgrids may lead to stability issues, even though each individual microgrid is stabilized by its local MGCC [2].

There are two categories of approaches to the design of stability assessment tools for networked microgrids. One category of approaches is to tailor the stability assessment development in bulk transmission systems for microgrid application. One well-studied TSA method developed for the transmission systems is the energy function method [5]–[8] which aims to construct an energy function that can certify stability. While an exact energy function can be developed for multi-machine power systems with transmission lines that have uniform R/X ratios [9], [10], it *generally* does not exist for power grids with lossy lines [5]. In distribution systems, large, uneven R/X ratios of distribution lines result in non-existence of the exact energy function in networked microgrids. It is plausible to use a numerical energy function for estimating critical clearing time [7], [8] in transmission systems with lossy lines. The numerical energy function however involves approximations on the path-dependent term [6]–[8], and its reliability and performance lack rigorous theoretical guarantee [5]. The TSA of transmission systems can also be achieved by decision-tree (DT) approaches [11]–[13]. While the DT-based approaches have a transparent decision-making process, how to connect decisions from decision trees to the classic stability concept [14] is an open question, and such a kind of interpretability is required by safety-critical power grids. The good performance of the DT-based approach heavily depends on thorough selection of training scenarios which is hard to be accomplished in practice. Another category of stability assessment approaches is designed specifically for microgrids and distribution systems. References [15] and [16] evaluate transient stability of microgrids by performing time-domain simulations under credible contingencies. The time-domain simulation approach is flexible for including complex dynamics of grid components, but it cannot certify stability rigorously, as by definition, stability [14] requires one to examine system responses under infinite number of disturbances, which is not possible for simulation-based time-domain methods. Reference [17] proposes a distributed framework capable of assessing the small-signal stability of networked microgrids. Reference [18] conducts small-signal stability analysis for DC microgrids. However, the small-signal analysis cannot quantify the magnitudes of large disturbances that distribution systems can tolerate. References [5], [19] and [20] construct a quadratic Lyapunov function which can be used for TSA of a power system with line loss. However, the TSA tool based on quadratic Lyapunov functions may be overly conservative. Besides, references [19] and [20] do not address the nonlinearity of AC microgrid network between multiple grid components. Reference [2] utilizes linear matrix inequalities (LMIs) in order to certify global asymptotic stability of networked microgrids. The framework proposed in [2] requires a special form of interface dynamics and it cannot characterize disturbances that can be tolerated by networked

microgrids when global asymptotic stability is not guaranteed. Reference [21] leverages the sum of square (SOS) technique to construct a Lyapunov function in order to assess the transient stability of networked microgrids. However, the SOS technique may require significant computation power when addressing large systems. Besides, most literature conducting stability analysis for conventional distribution systems, such as [22]–[24], focuses on voltage stability issues, where the analysis relies on steady-state models lying in a slow time scale. The stability issues in a fast time scale, such as angle stability, are rarely addressed in conventional distribution systems. This is because these fast dynamic problems are mainly mitigated by the host transmission systems which provide frequency regulation for the conventional distribution systems. However, networked microgrids may operate independently without connecting to transmission systems. As a result, the frequency may be regulated locally in a fast time scale by the networked microgrids, rather than the transmission systems. This renders the TSA in a fast time scale imperative for the networked microgrids.

In this paper, we develop a theoretically rigorous, machine learning-inspired TSA tool for networked microgrids. Assessing the transient stability of networked microgrids is formulated as a problem of computing the stability region. We leverage neural networks to learn a local Lyapunov function in the state space. The optimal stability region is estimated based on the Lyapunov function learned, and is used for characterizing disturbances that the networked microgrids can tolerate. Case studies suggest that the proposed TSA tool has the following merits: 1) It can assess the transient stability of networked microgrids with heterogeneous interface dynamics; 2) It can provide less conservative characterization of disturbances that can be tolerated by networked microgrids, compared with methods based on quadratic Lyapunov functions; and 3) It requires less computational power when addressing large systems, compared with the SOS-based approach [21]. Building upon our preliminary work [4], this paper substantially expands the scope by the following improvement: 1) We refine the Lyapunov risk, allowing for assessing transient stability of networked microgrids with mixed interface dynamics; 2) We develop a prerequisite checking condition to ensure the existence of Lyapunov functions; 3) We present a tuning procedure for the user-defined parameters in the proposed algorithms; 4) We propose an algorithm for estimating the largest stability region given a Lyapunov function learned; and 5) The proposed algorithm is tested in networked microgrids with mixed interface dynamics and a realistic 123-node feeder.

The rest of this paper is organized as follows: Section II describes the dynamics of networked microgrids; Section III presents the Neural Lyapunov method to TSA; and Section IV tests and validates the tool in three numerical experiments; and Section V concludes the paper and points out future direction.

## II. DYNAMICS OF MICROGRIDS WITH PE INTERFACES

With the physical configuration of the networked microgrids in Figure 1, the dynamics that a microgrid exhibits at the DSO-level control are mainly determined by the control

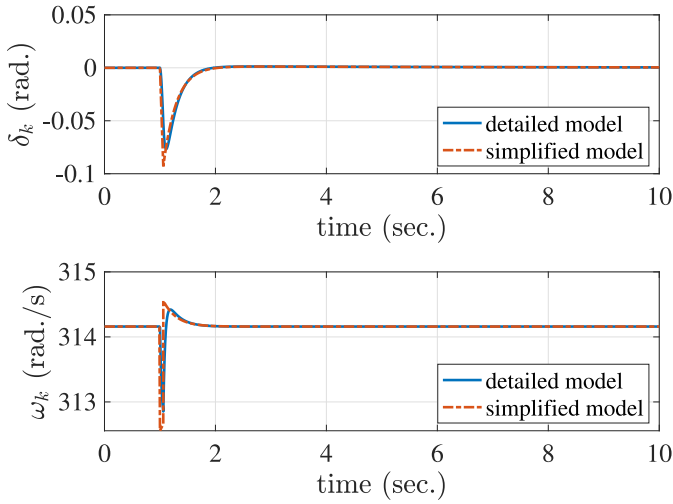


Fig. 2. Comparison between the full and simplified microgrid interface dynamics:  $\delta_k$  and  $\omega_k$ .

strategy deployed at its PE interface [2], [17]. This section characterizes the dynamics of the PE interfaces by presenting typical control schemes deployed at the microgrid interfaces. Based on the interface dynamics, we provide a mathematical description for the dynamics of networked microgrids.

For the  $n$  networked microgrids in Figure 1, without loss of generality, suppose that the angle droop control is deployed in the  $k$ -th microgrid's PE interface, where  $k = 1, 2, \dots, n$ . The interface dynamics of the  $k$ -th microgrid are [2], [17]

$$M_{ak}\dot{\delta}'_k + \delta'_k = D_{ak}(P_k^* - P_k) \quad (1a)$$

$$M_{vk}\dot{E}'_k + E'_k = D_{vk}(Q_k^* - Q_k) \quad (1b)$$

where  $\delta'_k$  and  $E'_k$  are *deviations* of voltage phase angle  $\delta_k$  and voltage magnitude  $E_k$  from their steady state values  $\delta_k^*$  and  $E_k^*$  at the  $k$ -th PCC, respectively, i.e.,  $\delta'_k := \delta_k - \delta_k^*$  and  $E'_k := E_k - E_k^*$ ;  $M_{ak}$  and  $M_{vk}$  are tracking time constants;  $D_{ak}$  and  $D_{vk}$  are droop gains;  $P_k$  and  $Q_k$  denote real and reactive power injections to PCC  $k$ ; and  $P_k^*$  and  $Q_k^*$  are the steady-state injections of real and reactive power at the  $k$ -th PCC [2].

Equation (1) is a simplified model of the PE interface dynamics. Appendix A describes a detailed model of a typical PE interfaces including a power calculator, a droop controller, a voltage and current controller and an output LC filter. The simplified model assumes that the dynamics of the power calculator, the voltage and current controller and the output filter are stabilized fast. Here, we demonstrate that the simplified dynamics can approximate the behavior of the PE interface modeled with details. Consider a microgrid interface connecting to its host distribution system via a tie line. At the tie line, suppose that a three-phase-to-ground fault occurs at the 1st second and it is cleared 3 cycles later. Figure 2 shows the simulation of the microgrid interface response to the event based on the detailed model and the simplified model. The simulation details are presented in Appendix B. It can be observed in Figure 2 that the simplified model can reflect the general trend of the response of the interface modeled with details. Therefore, we use the simplified dynamics to model the behaviors of the PE interfaces.

The  $n$  microgrids are networked via distribution network which introduces constrains

$$P_k - G_{kk}E_k^2 - \sum_{i \neq k} E_k E_i Y_{ki} \cos(\delta_{ki} - \sigma_{ki}) = 0 \quad (2a)$$

$$Q_k + B_{kk}E_k^2 - \sum_{i \neq k} E_k E_i Y_{ki} \sin(\delta_{ki} - \sigma_{ki}) = 0, \quad \forall k, \quad (2b)$$

where  $\delta_{ki} = \delta_k - \delta_i$ ;  $G_{kk} + jB_{kk}$  is the  $k$ -th diagonal entry in the admittance matrix of the distribution network; and  $Y_{ki} \angle \sigma_{ki}$  is the  $(k, i)$ -th entry of the admittance matrix. The steady-state values  $\delta_k^*$ ,  $E_k^*$ ,  $P_k^*$  and  $Q_k^*$  are designed based on economic dispatch and they satisfy the following equality constrains:

$$P_k^* - G_{kk}E_k^{*2} - \sum_{i \neq k} E_k^* E_i^* Y_{ki} \cos(\delta_{ki}^* - \sigma_{ki}) = 0 \quad (3a)$$

$$Q_k^* + B_{kk}E_k^{*2} - \sum_{i \neq k} E_k^* E_i^* Y_{ki} \sin(\delta_{ki}^* - \sigma_{ki}) = 0, \quad \forall k, \quad (3b)$$

where  $\delta_{ki}^* = \delta_k^* - \delta_i^*$ . Differential equations (1) with algebraic equations (2) characterize the dynamics of the  $n$  networked microgrids, and their compact form is

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \quad (4)$$

where  $\mathbf{x} = [\delta'_1, \delta'_2, \dots, \delta'_n, E'_1, E'_2, \dots, E'_n]$ ; and  $\mathbf{f}(\cdot)$  is determined by (1) and (2). Note that the equilibrium point  $\mathbf{o}$  of the dynamic system (4) is the origin of the state space.

If  $M_{vk} \gg M_{ak}$ , the *time-scale separation* can be assumed [2], [4]. In such a case, the voltage deviation  $E'_k$  evolves much slower than the phase angle deviation  $\delta'_k$  and, therefore,  $E'_k$  is assumed to be constant [2]. Furthermore, if only angular stability is of interest, the dynamics of the  $n$  networked microgrids can be described by

$$M_{ak}\dot{\delta}'_k + \delta'_k = D_{ak}(P_k^* - P_k), \quad \forall k, \quad (5)$$

where  $P_k = G_{kk}E_k^{*2} + \sum_{i \neq k} E_k^* E_i^* Y_{ki} \cos(\delta_{ki}^* + \delta_{ki}^* - \sigma_{ki})$ . The compact form of (5) can be also expressed as (4) where  $\mathbf{x}$  and  $\mathbf{f}(\cdot)$  should be revised accordingly. Besides, with the time-scale separation assumption, if the frequency droop control is deployed in the  $j$ -th microgrid, the  $j$ -th differential equation in (5) is replaced by

$$\dot{\delta}'_j = \omega'_j, \quad M_{tj}\dot{\omega}'_j + D_{tj}\omega'_j = P_j^* - P_j$$

where  $\omega'_j = \omega_j - \omega_n$ .

With the networked microgrids (4) and its equilibrium point  $\mathbf{o}$ , a DSO may have the following two questions [4]: 1) Is  $\mathbf{o}$  asymptotically stable? 2) How ‘‘large’’ are the disturbances that the networked microgrids can tolerate? Suppose that the length of the state vector  $\mathbf{x}$  is  $m$ , and recall that the equilibrium point is at the origin, i.e.,  $\mathbf{o} = \mathbf{0}$ , the definition of the asymptotic stability is introduced as follows:

*Definition 1 (Asymptotic Stability and Stability Region [14]):* An equilibrium point  $\mathbf{0}$  is asymptotically stable, if

$$\forall \epsilon > 0, \exists \zeta > 0, \|\mathbf{x}(0)\| < \zeta \implies \forall t \geq 0, \|\mathbf{x}(t)\| < \epsilon$$

$$\text{and } \lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{0}. \quad (6)$$

Moreover, when the system is asymptotically stable at  $\mathbf{0}$ , its stability region, or region of attraction, is defined as  $\mathcal{D} = \{x_0 \in \mathbb{R}^m | \mathbf{x}(0) = x_0 \implies \lim_{t \rightarrow \infty} \mathbf{x}(t) = \mathbf{0}\}$ .

In Definition 1,  $\mathbf{x}(0)$  is resulting from the microgrid interconnection-level events, say, topology changes of distribution system network, and one of the microgrids enters an islanded/grid-connected mode. The transient stability assessment framework proposed in this paper aims to answer the two DSO questions.

### III. NEURAL LYAPUNOV METHODS

This section addresses two important questions from a DSO's perspective. We first point out the asymptotic stability of networked microgrids can be certified by the Lyapunov linearization method [14] and formulate the second DSO's question as the one of estimating a stability region of networked microgrids. Then an optimal stability region is estimated via learning a Lyapunov function. Finally, how to empirically tune the parameters of proposed algorithms is discussed.

#### A. Asymptotic Stability Check

Given the networked microgrids (4) and its equilibrium  $\mathbf{o}$ , the Lyapunov linearization method [14] suggests the asymptotic stability of  $\mathbf{o}$  can be determined by examining the linearized version of (4), i.e.,

$$\dot{\mathbf{x}} = A\mathbf{x}. \quad (7)$$

In (7),  $A \in \mathbb{R}^{m \times m}$  is a system matrix, where  $m$  is the length of the state vector  $\mathbf{x}$ . The system matrix  $A$  is obtained by linearizing (4) around its equilibrium point  $\mathbf{o}$  based on the linearization technique [25] which has been extensively used for characterizing small-signal phenomena of power grids [26]–[30]. Suppose that matrix  $A$  has  $m$  eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_m$ . The equilibrium point  $\mathbf{o}$  of (4) is asymptotically stable [14], if

$$\text{Re}(\lambda_i) < 0 \quad \forall i = 1, 2, \dots, m. \quad (8)$$

Condition (8) answers the first question raised in Section II.

For the second DSO's question, the stability region in Definition 1 can be leveraged to characterize the disturbances that the networked microgrids (4) operating at  $\mathbf{o}$  are able to tolerate. Definition 1 essentially says that the system trajectory starting in the stability region  $\mathcal{D}$  tends to the equilibrium point  $\mathbf{o}$ . The second DSO's question can be answered if such a stability region is obtained.

A stability region  $\mathcal{D}$  can be estimated based on a system behavior-summary function, i.e., a Lyapunov function, in conjunction with the Local Invariant Set Theorem [14]. The Lyapunov function is given by the following definition [4]:

*Definition 2:* A continuous differentiable scalar function  $V(\mathbf{x})$  is a Lyapunov function for an autonomous system (4) with equilibrium point  $\mathbf{0}$ , if, in a ball  $\mathcal{B}_u := \{\mathbf{x} \in \mathbb{R}^m | u > 0, \|\mathbf{x}\|_2^2 < u^2\}$ , 1)  $V$  is positive definite in  $\mathcal{B}_u$ , and 2)  $\dot{V}$  is negative definite in  $\mathcal{B}_u$ .

Once a legitimate Lyapunov function  $V(\mathbf{x})$  becomes available, a stability region can be estimated by a region  $\mathcal{S}_d$  that is defined by any sublevel set of  $V(\mathbf{x})$  completely contained in the ball  $\mathcal{B}_u$ , i.e., for  $d > 0$ ,

$$\mathcal{S}_d = \{\mathbf{x} \in \mathbb{R}^m | V(\mathbf{x}) < d\} \subseteq \mathcal{B}_u. \quad (9)$$

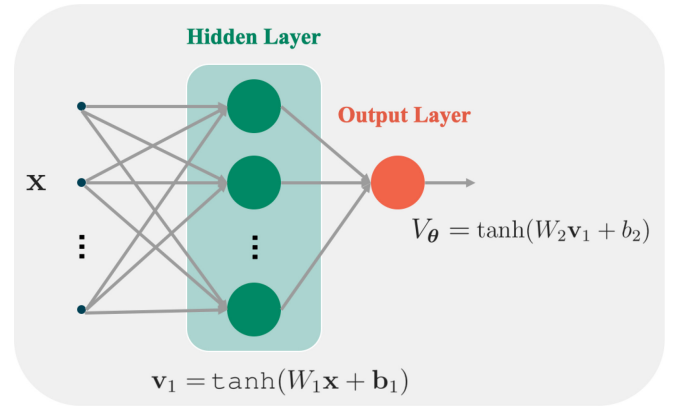


Fig. 3. A neural network with one hidden layer.

The region  $\mathcal{S}_d$  is an invariant set due to the decreasing nature of the Lyapunov function  $V(\mathbf{x})$ . Besides, the Invariant Set Theorem [14] suggests that with the Lyapunov function  $V(\mathbf{x})$ , a system trajectory  $\mathbf{x}(t)$  starting in  $\mathcal{S}_d$  converges to the origin of the state space. Therefore, the region  $\mathcal{S}_d$  is a stability region. In order to characterize the disturbances that the networked microgrids can tolerate, the remaining questions are: 1) How to find a legitimate Lyapunov function in a valid region  $\mathcal{B}_u$ ; and 2) with a Lyapunov function valid in  $\mathcal{B}_u$ , how to make the stability region  $\mathcal{S}_d$  as large as possible by tuning  $d$  in (9). These two questions are addressed in Sections III-B and III-C.

#### B. Learning Lyapunov Function From the State Space

##### 1) Lyapunov Function With Neural Network Structure:

We assume that a Lyapunov function candidate is a neural network. The neural network has a hidden layer and an output layer [31], [32]. The input of the hidden layer is the state vector  $\mathbf{x} \in \mathbb{R}^m$  and the output is a vector  $\mathbf{v}_1 \in \mathbb{R}^p$  where  $p$  is the number of neurons in the hidden layer, as shown in Figure 3. Function  $g_1 : \mathbb{R}^m \rightarrow \mathbb{R}^p$  describes the relationship between  $\mathbf{x}$  and  $\mathbf{v}_1$  and its definition is

$$\mathbf{v}_1 = g_1(\mathbf{x}) := \tanh(W_1 \mathbf{x} + \mathbf{b}_1) \quad (10)$$

where  $W_1 \in \mathbb{R}^{p \times m}$ ,  $\mathbf{b}_1 \in \mathbb{R}^p$ ; and  $\tanh(\cdot)^2$  is an entry-wised hyperbolic tangent function [4]. Furthermore, we define an intermediate vector  $\mathbf{c}_1 = [c_{1,1}, c_{1,2}, \dots, c_{1,p}]^T$  for the hidden layer by  $\mathbf{c}_1 = W_1 \mathbf{x} + \mathbf{b}_1$ . For the output layer, its input is vector  $\mathbf{v}_1$  and its output is  $V_\theta \in \mathbb{R}$  which is interpreted as the Lyapunov candidate evaluated at vector  $\mathbf{x}$ .  $V_\theta$  is related with  $\mathbf{v}_1$  via function  $g_2 : \mathbb{R}^p \rightarrow \mathbb{R}$  defined by

$$V_\theta = g_2(\mathbf{v}_1) := \tanh(W_2 \mathbf{v}_1 + b_2) \quad (11)$$

where  $W_2 \in \mathbb{R}^{1 \times p}$ ; and  $b_2 \in \mathbb{R}$ . The intermediate variable  $c_2$  associated with the output layer is defined by  $c_2 = W_2 \mathbf{v}_1 + b_2$ . In sum, the Lyapunov function candidate is

$$V_\theta(\mathbf{x}) = g_2(g_1(\mathbf{x})). \quad (12)$$

Denote by  $\theta$  the vector that consists of all unknown entries in  $W_1$ ,  $\mathbf{b}_1$ ,  $W_2$ , and  $b_2$ . The subscript of  $V_\theta$  indicates that the Lyapunov function candidate depends on  $\theta$ .

<sup>2</sup>In this paper, “ $\tanh(\cdot)$ ” and “ $\tanh(\cdot)$ ” are different. The input argument of “ $\tanh(\cdot)$ ” can be a vector, while the input argument of “ $\tanh(\cdot)$ ” can only be a scalar.

**Algorithm 1** Lyapunov Risk Minimization

---

```

1: function MinRisk( $\theta_0, \mathcal{X}, \mathbf{f}, p, r, \eta, \alpha, \beta, \gamma, \tau$ )
2:    $\theta \leftarrow \theta_0$ 
3:   while  $i \leq r$  do
4:     Update  $V_\theta$  and  $\dot{V}_\theta$  by (12), (14) with  $\theta$ 
5:     Compute  $R_{|\mathcal{X}|, \rho}(\theta)$  via (13) over  $\mathcal{X}$ 
6:      $\theta \leftarrow \theta - \eta \nabla_{\theta} R_{|\mathcal{X}|, \rho}(\theta)$ ;  $i \leftarrow i + 1$ 
7:   end while
8:   return  $\theta$ 
9: end function

```

---

2) *Lyapunov Risk Minimization*: We proceed to tune  $\theta$  such that  $V_\theta(\mathbf{x})$  in (12) meets the two conditions in Definition 2. Suppose that there are  $q$  state vectors  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_q$ . Let set  $\mathcal{X}$  collect these  $q$  vector samples. To tune  $\theta$ , we introduce a cost function called (empirical) Lyapunov risk, i.e.,

$$R_q(\theta) = \frac{\alpha}{q} \sum_{i=1}^q (\text{ReLU}(-V_\theta(\mathbf{x}_i))) + \frac{\beta}{q} \sum_{i=1}^q (\text{ReLU}(\dot{V}_\theta(\mathbf{x}_i) + \tau)) + \gamma V_\theta^2(\mathbf{0}) \quad (13)$$

where the tunable parameters  $\alpha, \beta, \gamma$  and  $\tau$  are positive scalars;  $\text{ReLU}(\cdot)$  denotes the rectified linear unit defined by

$$\text{ReLU}(y) = \begin{cases} y, & y \geq 0 \\ 0, & y < 0; \end{cases}$$

and  $\dot{V}_\theta$  is given by [4]

$$\dot{V}_\theta = \frac{\partial V_\theta}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) = \frac{\partial V_\theta}{\partial c_2} \frac{\partial c_2}{\partial \mathbf{v}_1} \frac{\partial \mathbf{v}_1}{\partial \mathbf{c}_1} \frac{\partial \mathbf{c}_1}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}). \quad (14)$$

In (14), the dynamics  $\mathbf{f}(\mathbf{x})$  is provided in (4);

$$\frac{\partial V_\theta}{\partial c_2} = 1 - V_\theta^2; \quad \frac{\partial c_2}{\partial \mathbf{v}_1} = W_2; \quad \frac{\partial \mathbf{c}_1}{\partial \mathbf{x}} = W_1; \quad \text{and} \\ \frac{\partial \mathbf{v}_1}{\partial \mathbf{c}_1} = \text{diag}(1 - \tanh^2(c_{1,1}), \dots, 1 - \tanh^2(c_{1,p})).$$

The interpretation of the Lyapunov risk (13) is presented as follows. In (13), The first ‘‘ReLU’’ term incurs positive penalty if  $V_\theta(\mathbf{x}_i)$  is negative. The second ‘‘ReLU’’ term results to positive penalty if  $\dot{V}_\theta(\mathbf{x}_i)$  is greater than  $-\tau$ . If the evaluation of  $V_\theta$  at the origin of the state space is not zero, the Lyapunov risk also increases according to (13). Parameters  $\alpha, \beta, \gamma$  and  $\tau$  determine the importance of the three terms of (13) and their tuning procedure is discussed in Section III-D.

Given the training set  $\mathcal{X}$ , in order to find a Lyapunov function valid in  $\mathcal{B}_u$ , unknown parameters  $\theta$  should be chosen such that the Lyapunov risk  $R_q(\theta)$  is minimized, viz.

$$\min_{\theta} R_q(\theta). \quad (16)$$

The gradient decent algorithm can be leveraged to solve (16). Algorithm 1 presents a procedure to update  $\theta$ , where  $\theta_0$  is the initial guess of  $\theta$ ;  $r \in \mathbb{Z}_+$  denotes the times of updating  $\theta$ ; and the positive scalar  $\eta$  is the learning rate. Note that merely using Algorithm 1 to update  $\theta$  is not sufficient even with a large  $r$ . One reason is that  $\mathcal{X}$  solely covers a finite number of training samples in  $\mathcal{B}_u$ . With the  $\theta$  obtained by Algorithm 1 based on  $\mathcal{X}$ , it is possible that one or both of the two conditions in Definition 2 are violated in some part of  $\mathcal{B}_u$  that is not included in  $\mathcal{X}$ . This issue is addressed in Section III-B3.

**Algorithm 2** Learning Lyapunov Function

---

```

1: function AddSample( $\mathcal{X}, V_\theta, \mathbf{f}, u$ )
2:    $\kappa \leftarrow 1$ 
3:   Check (17) in  $\mathcal{B}_u$  and find  $\mathcal{C}$  by dReal
4:   if  $\mathcal{C} = \emptyset$  then  $\kappa \leftarrow 0$             $\triangleright$  No counterexamples found
5:   else  $\mathcal{X} \leftarrow \mathcal{C} \cup \mathcal{X}$             $\triangleright$  Add counterexamples to  $\mathcal{X}$ 
6:   end if
7:   return  $\mathcal{X}, \kappa$ 
8: end function
9: function LearnFunc( $\mathcal{X}, \theta_0, \mathbf{f}, u, p, r, \eta, \alpha, \beta, \gamma, \tau, n_i$ )
10:   $\kappa \leftarrow 1$ ;  $j \leftarrow 0$ 
11:  while  $(\kappa = 1) \wedge (j \leq n_i)$  do
12:     $\theta \leftarrow \text{MinRisk}(\theta_0, \mathcal{X}, \mathbf{f}, p, r, \eta, \alpha, \beta, \gamma, \tau)$ 
13:     $\theta_0 \leftarrow \theta$ ;  $j \leftarrow j + r$ 
14:     $\mathcal{X}, \kappa \leftarrow \text{AddSample}(\mathcal{X}, V_\theta, \mathbf{f}, u)$ 
15:  end while
16:  if  $\kappa = 0$  then  $V_{\theta^*} \leftarrow V_\theta - V_\theta(0)$ 
17:  else  $V_{\theta^*} \leftarrow \emptyset$ 
18:  end if
19:  return  $V_{\theta^*}$ 
20: end function

```

---

3) *Augment of Training Samples*: Here, we utilize the satisfiability modulo theories (SMT) solver [33] to analytically check if the function learned by MinRisk is a legitimate Lyapunov function. This is equivalent to searching for state vectors  $\mathbf{x} \in \mathcal{B}_u$  that satisfy

$$(V_\theta(\mathbf{x}) \leq 0 \vee \dot{V}_\theta \geq 0) \wedge (\|\mathbf{x}\|_2^2 \geq l^2) \quad (17)$$

where  $l$  is a small scalar; and  $\|\mathbf{x}\|_2^2 \geq l^2$  is added for avoiding numerical issues of the SMT solver [34]. The state vectors  $\mathbf{x} \in \mathcal{B}_u$  satisfy condition (17) are termed *counterexamples* which can be found by the SMT solver, such as dReal [33]. Denote by  $\mathcal{C}$  the set that consists of the counterexamples found by the SMT solver. If  $\mathcal{C}$  is not an empty set, the learned function is not a Lyapunov function and the richness of the training set  $\mathcal{X}$  is enhanced by adding counterexamples in  $\mathcal{C}$  to  $\mathcal{X}$ . If dReal cannot find such a counterexample, i.e.,  $\mathcal{C}$  is an empty set, thanks to the delta-completeness property [33] of dReal, it is guaranteed that there is no such a counterexample, thereby suggesting that the function learned is a legitimate Lyapunov function. The procedure of augmenting the training samples is presented in the AddSample function of Algorithm 2.

The function LearnFunc of Algorithm 2 summarizes the overall procedure of updating the unknown parameter  $\theta$  and augmenting the training set  $\mathcal{X}$ . In LearnFunc,  $n_i$  is the maximum iteration times defined by users.

*Remark*: The proposed method requires the availability of dynamics (4). The neural network (12) in this paper is merely for the purpose of learning a Lyapunov function for (4), instead of identifying the networked microgrids dynamics (4).

**C. Stability Region Estimation Algorithm**

Given a Lyapunov function  $V_{\theta^*}$  with its valid region  $\mathcal{B}_u$ , we proceed to tune  $d$  in (9) so that the estimated stability region is maximized. The optimal  $d^*$  is determined by solving [35]

$$d^* = \min_{\mathbf{x}} V_{\theta^*}(\mathbf{x}) \quad (18a)$$

$$\text{s.t. } \|\mathbf{x}\|_2^2 = u^2. \quad (18b)$$



The state vectors satisfying the equality constrain (18b) constitute the boundary of the valid region  $\mathcal{B}_u$  of  $V_{\theta^*}$ . Equation (18a) essentially says that  $d^*$  is the minimal value of  $V_{\theta^*}(\mathbf{x})$  evaluated along  $\mathcal{B}_u$ 's boundary.

The optimization (18) can be solved by finding *critical points* defined as follows. The Lagrangian  $L(\mathbf{x}, \phi)$  of (18) is

$$L(\mathbf{x}, \phi) = \phi \left( \|\mathbf{x}\|_2^2 - u^2 \right) + V_{\theta^*}(\mathbf{x}). \quad (19)$$

where  $\phi \in \mathbb{R}$ . Define a set  $\mathcal{P}$  by

$$\mathcal{P} := \left\{ \mathbf{x} \in \mathbb{R}^m \mid \frac{\partial L(\mathbf{x}, \phi)}{\partial \mathbf{x}} = \mathbf{0}, \|\mathbf{x}\|_2^2 - u^2 = 0 \right\}. \quad (20)$$

Each element of the set  $\mathcal{P}$  is a critical point. The global minimum of  $V_{\theta^*}$  over  $\mathcal{B}_u$ 's boundary occurs at one of the critical points. Finding  $\mathcal{P}$  is equivalent to obtaining all solutions to

$$2\phi\mathbf{x} + \frac{\partial V_{\theta^*}}{\partial \mathbf{x}} = \mathbf{0}; \quad \|\mathbf{x}\|_2^2 - u^2 = 0. \quad (21)$$

Unknown parameters  $W_1, W_2, \mathbf{b}_1$ , and  $\mathbf{b}_2$  in (10) and (11) can be updated by the  $\theta^*$  returned by Algorithm 2. Denote by  $W_1^*, W_2^*, \mathbf{b}_1^*$ , and  $\mathbf{b}_2^*$  the updated version of  $W_1, W_2, \mathbf{b}_1$ , and  $\mathbf{b}_2$ , respectively. In (21),

$$\frac{\partial V_{\theta^*}}{\partial \mathbf{x}} = \left( 1 - V_{\theta^*}(\mathbf{x})^2 \right) W_2^* W_1^* \Lambda \quad (22)$$

where  $\Lambda = \text{diag}(1 - \tanh^2(c_{1,1}^*), \dots, 1 - \tanh^2(c_{1,p}^*))$ , whence  $[c_{1,1}^*, \dots, c_{1,p}^*]^\top = W_1^* \mathbf{x} + \mathbf{b}_1^*$ . With (22), (21) becomes algebraic equations whose compact form is

$$\mathbf{h}(\mathbf{x}, \phi) = \mathbf{0}. \quad (23)$$

The Newton-Krylov (NK) method [36] can solve (23) for  $\mathbf{x}$  and  $\phi$  with the initial guesses  $\mathbf{x}_0$  and  $\phi_0$  on solutions. If set  $\mathcal{P}$  is available,

$$d^* = \min_{\mathbf{x} \in \mathcal{P}} V_{\theta^*}(\mathbf{x}). \quad (24)$$

Then, the corresponding stability region is

$$\mathcal{S}_{d^*} = \{ \mathbf{x} \in \mathbb{R}^m \mid V_{\theta^*}(\mathbf{x}) < d^* \}. \quad (25)$$

With the Lyapunov function learned by LearnFunc, the procedure to estimating a stability region is provided by the SREst function of Algorithm 3, where  $\text{NK}(\mathbf{h}, \mathbf{x}_0, \phi_0)$  denotes the procedure of solving  $\mathbf{h}(\mathbf{x}, \phi) = \mathbf{0}$  with the initial guesses  $\mathbf{x}_0$  and  $\phi_0$  using the NK method; and the NK procedure returns  $\mathbf{x}^*$  and  $\phi^*$  which constitute a solution to  $\mathbf{h}(\mathbf{x}, \phi) = \mathbf{0}$ . The solution found by the NK procedure depends on the initial guesses  $\mathbf{x}_0$  and  $\phi_0$ . To find all critical points, the SREst function repetitively solves (23) for  $n_{\text{sr}}$  times. For each time of solving (23),  $\mathbf{x}_0$  and  $\phi_0$  are randomly realized by Lines 4 and 5 of Algorithm 3, instead of being chosen by users. The Main function of Algorithm 3 summarizes the procedure described in Sections III-A, III-B, and III-C. Line 24 of Algorithm 3 requires a parameter tuning procedure to obtain a system matrix  $A$  with all eigenvalues on the left-half plane. Such a procedure can be designed based on the eigenvalue sensitivity analysis [37] or  $\mathcal{H}_\infty$  optimization [38]. Note that checking asymptotic stability of the given equilibrium (Lines 14-16 of Algorithm 3) is a prerequisite for learning a Lyapunov function and estimating an optimal stability region.

---

### Algorithm 3 Stability Region Estimation

---

```

1: function SREst( $V_{\theta^*}, u, n_{\text{sr}}$ )
2:    $\mathcal{P} \leftarrow \emptyset$ ; construct  $\mathbf{h}$  by (21), (22)
3:   for  $k = 1, 2, \dots, n_{\text{sr}}$  do
4:     Pick a random  $\mathbf{x}_0$  in  $\{\mathbf{x}_0 \in \mathbb{R}^m \mid \|\mathbf{x}_0\|_2^2 = u^2\}$ 
5:     Pick a random  $\phi_0 \in \mathbb{R}$ 
6:      $\mathbf{x}^*, \phi^* \leftarrow \text{NK}(\mathbf{h}, \mathbf{x}_0, \phi_0)$ 
7:     if  $\mathbf{x}^* \notin \mathcal{P}$  then  $\mathcal{P} \leftarrow \mathcal{P} \cup \mathbf{x}^*$ 
8:     end if
9:   end for
10:  Obtain  $\mathcal{S}_{d^*}$  via (24), (25)
11:  return  $\mathcal{S}_{d^*}$ 
12: end function
13: function Main( $\mathbf{f}, u, p, q, \theta_0, r, \eta, \alpha, \beta, \gamma, \tau, n_{\text{sr}}, n_i$ )
14:  Linearize  $\mathbf{f}$  to obtain  $A$  in (7)
15:  Compute eigenvalues  $\lambda_i$  of  $A \forall i = 1, 2, \dots, m$ 
16:  if (8) holds then ▷ Asymptotic stability check
17:    Construct  $\mathcal{X}$  by randomly picking  $q$  vectors in  $\mathcal{B}_u$ 
18:     $V_{\theta^*} \leftarrow \text{LearnFunc}(\mathcal{X}, \theta_0, \mathbf{f}, u, p, r, \eta, \alpha, \beta, \gamma, \tau, n_i)$ 
19:    if  $V_{\theta^*} \neq \emptyset$  then
20:       $\mathcal{S}_{d^*} \leftarrow \text{SREst}(V_{\theta^*}, u, n_{\text{sr}})$ 
21:      return  $V_{\theta^*}, \mathcal{S}_{d^*}$ 
22:    else Request for tuning user-defined parameters
23:    end if
24:  else Request for tuning parameters in (4)
25:  end if
26: end function

```

---

#### D. Parameter Tuning

In Algorithm 3, the empirical settings of  $\theta_0, p, q, r, \eta, n_{\text{sr}}, n_i$  and  $\tau$  are provided as follows: the random initial guess  $\theta_0$  is obtained by the initialization procedure reported in [39];  $p \geq 2m$ ;  $q, n_{\text{sr}}$ , and  $n_i$  are 500, 100, and 5000, respectively; integer  $r \in [10, 30]$ ;  $\tau \in [0.1, 0.5]$ ; and  $\eta \in [0.01, 0.02]$ .

Given a set of user-defined parameters, it is possible that the Main function returns an empty set  $\emptyset$ , meaning that the function fails to find a Lyapunov function valid in  $\mathcal{B}_u$  within  $n_i$  iterations. Solutions to such a situation include 1) decreasing  $u$ ; 2) changing  $\theta_0$ ; and 3) tuning  $\alpha, \beta$ , and  $\gamma$ . Solution 1 works because there may not be a Lyapunov function in a large ball. Solving (16) using gradient-based methods depends on the initial guess on the solution, which justifies Solution 2.

Next we present an empirical procedure to tune  $\alpha, \beta$ , and  $\gamma$ . Denote by  $\theta_{n_i}$  the  $n_i$ -th update of  $\theta$  in LearnFunc. The function  $V_{\theta_{n_i}}$  and its time derivative  $\dot{V}_{\theta_{n_i}}$  can be visualized in subspace of  $\mathcal{B}_u$ . The visualization may suggest which condition(s) in Definition 2 is (are) violated, thereby pointing out the “direction” of tuning  $\alpha, \beta$ , and  $\gamma$ . For example, suppose that one needs to learn a Lyapunov function for a system whose state variables are  $[\delta'_1, \delta'_2, \delta'_3, \omega'_3]$  using LearnFunc. After  $n_i$ -time parameter updates, the function with parameter  $\theta_{n_i}$  and its time derivative can be visualized by numerically evaluating the functions within  $\mathcal{B}_u$ 's projection to the  $\delta'_1$ - $\omega'_3$  plane with  $\delta'_2 = \delta'_3 = 0$ . Suppose that the visualization is given in Figure 4. As shown in Figure 4, the function with parameter  $\theta_{n_i}$  is *not* a Lyapunov function in  $\mathcal{B}_{0.4}$ , because its time derivative is not negative in  $\mathcal{B}_{0.4}$ , although the function is positive. Figure 4 indicates that with other parameters fixed, one may need to increase the penalty resulting from the violation of the second condition of Definition 2, i.e., increasing  $\beta$  in (13).

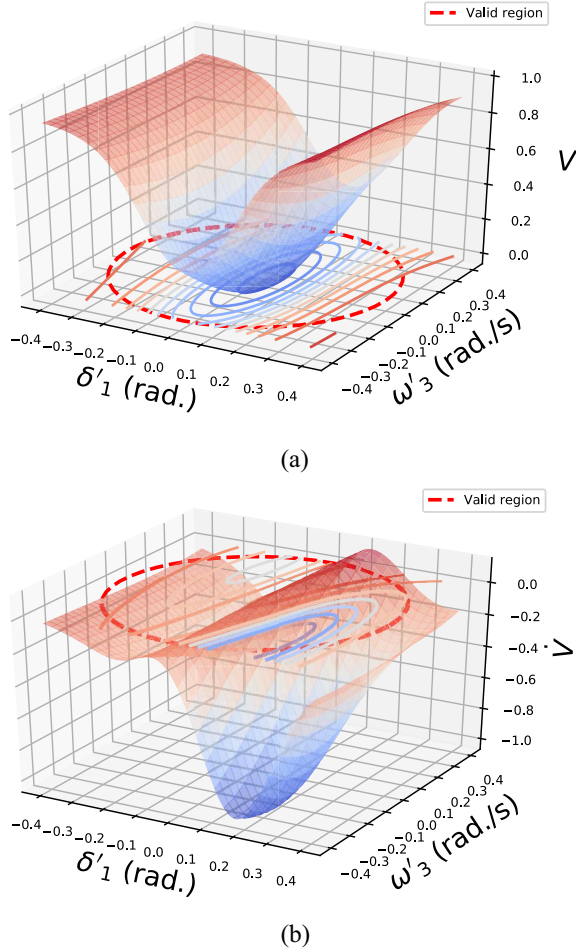


Fig. 4. Visualization of the function (a) and its time derivative (b) after  $n_i$  times of parameter update: the function is NOT a Lyapunov function.

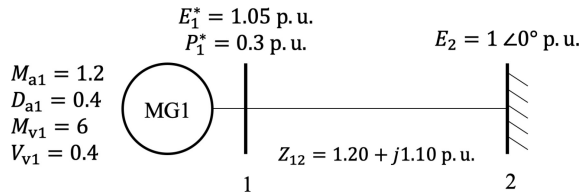


Fig. 5. A grid-connected microgrid [4].

#### IV. NUMERICAL EXPERIMENTS

This section tests and validates the proposed method in a grid-connected microgrid, a three-microgrid interconnection with mixed dynamics, and the IEEE 123-node feeder. All experiments in this section are conducted on a MacBook Pro (2.6 GHz Intel Core i5) with Python 3.7.7.

##### A. A Grid-Connected Microgrid

A grid-connected microgrid (MG) with angle-droop control is shown in Figure 5. The user-defined parameters required in Algorithm 3 are listed in Table I.

1) *Learned Lyapunov Function*: After 500 times of parameter updates, which takes 32.18 seconds, Algorithm 2 outputs a Lyapunov function. Figure 6 shows the Lyapunov function learned and its time derivative. As shown in Figure 6, the

TABLE I  
USER-DEFINED PARAMETERS

| Case Name                  | $p$    | $q$ | $n_{sr}$ | $n_i$   | $r$      | $\tau$ |
|----------------------------|--------|-----|----------|---------|----------|--------|
| A Grid-connected Microgrid | 6      | 500 | 100      | 5000    | 10       | 0.1    |
| Three Networked Microgrids | 8      | 500 | 100      | 5000    | 30       | 0.1    |
| IEEE 123-node Feeder       | 8      | 500 | 100      | 5000    | 10       | 0.5    |
| Case Name                  | $\eta$ | $u$ | $\alpha$ | $\beta$ | $\gamma$ | N/A    |
| A Grid-connected Microgrid | 0.01   | 1.5 | 1        | 5       | 0        | N/A    |
| Three Networked Microgrids | 0.02   | 0.4 | 3        | 1       | 3        | N/A    |
| IEEE 123-node Feeder       | 0.01   | 0.7 | 1        | 1       | 0        | N/A    |

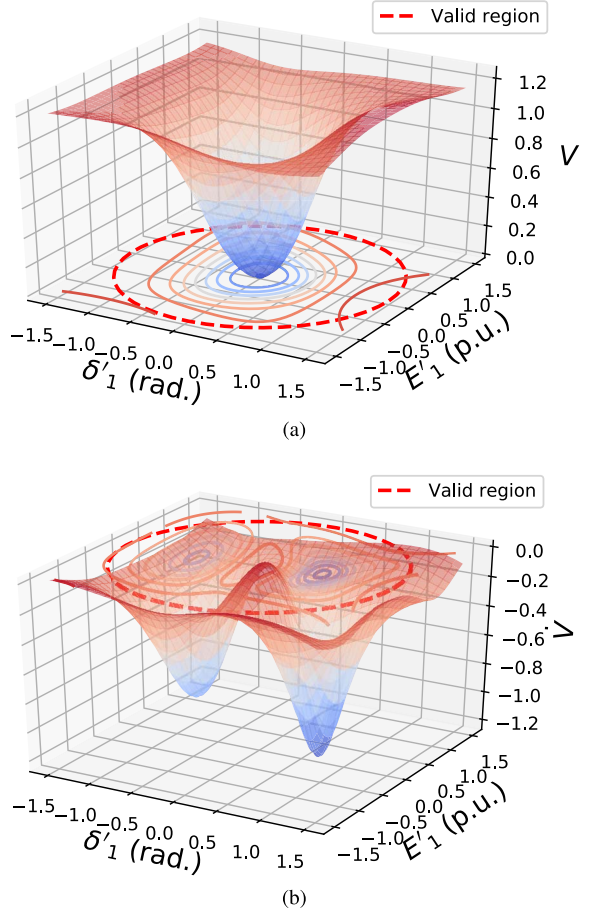


Fig. 6. (a) Lyapunov function and (b) its time derivative for a grid-tied MG.

function learned is positive definite in the valid region (VR)  $\mathcal{B}_{1.5}$  and its time derivative is negative definite in  $\mathcal{B}_{1.5}$ . This suggests that the function learned is a Lyapunov function.

2) *Estimated Stability Region*: Given the Lyapunov function learned with its VR  $\mathcal{B}_{1.5}$ , the stability region (SR) estimated by Algorithm 3 is  $\mathcal{S}_{1.01}$  which is defined by (25). In Figure 7, the red-solid circle is the boundary of  $\mathcal{S}_{1.01}$ , while the red-dash circle is the boundary of  $\mathcal{B}_{1.5}$ ; and the region enclosed by the red-solid circle is a SR. Besides, the SREst function suggests that  $d^*$  in (18) is 1.01 which is attained when  $\delta_1' = -0.82$  and  $E_1' = 1.26$ .

We proceed to check the correctness of the estimated SR  $\mathcal{S}_{1.01}$ . Since the test system only has two state variables, given the Lyapunov function learned, the largest SR can be found without solving optimization (18). For example, we can visualize a SR  $\mathcal{S}_d$  with a small  $d$ , say,  $d = 0.15$ . Figure 7-(b)

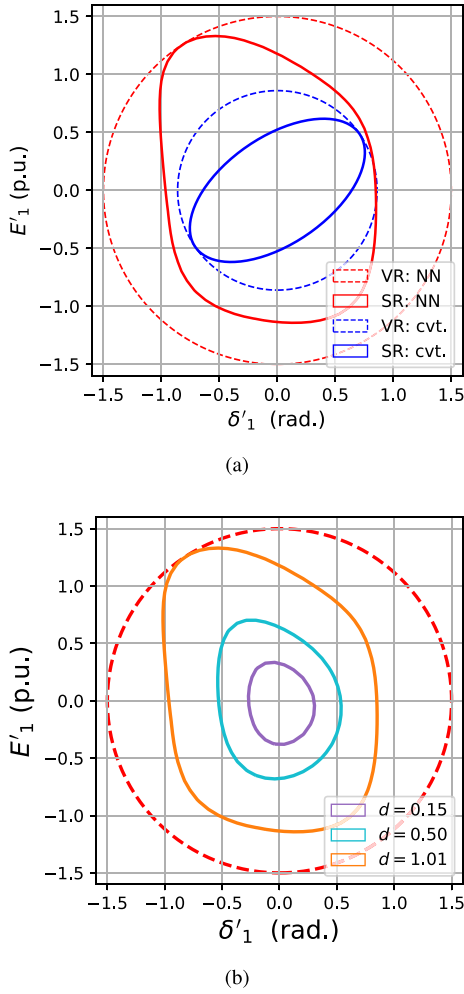


Fig. 7. (a) Comparison between the proposed (NN) and conventional (cvt.) methods: SR and VR (b) An alternative way to find  $\mathcal{S}_{d^*}$  by tuning  $d$ .

visualize  $\mathcal{S}_{0.15}$ . We keep increasing  $d$  gradually until the boundary of  $\mathcal{S}_d$  touches the boundary of  $\mathcal{B}_{1.5}$  for the first time. As can be observed in Figure 7-(b), when  $d = 1.01$ , the boundaries of  $\mathcal{S}_d$  and  $\mathcal{B}_{1.5}$  touch with each other at  $(-0.82, 1.26)$ . Therefore,  $\mathcal{S}_{1.01}$  is the largest SR that can be estimated based on the learned Lyapunov function. The SR obtained by such a procedure is consistent with the one estimated by function  $\text{SREst}$ .

3) *Comparison*: The proposed method is compared with a conventional method reported in [5]. Denote by  $\mathcal{S}'$  the SR estimated based on a quadratic Lyapunov function constructed in [5]. In Figure 7, the region enclosed by the blue-solid circle is  $\mathcal{S}'$ , while the blue-dash circle is the boundary of the VR of the quadratic Lyapunov function. It can be observed that  $\mathcal{S}_{1.01}$  is larger than  $\mathcal{S}'$ . This suggests that the propose method can provide a less conservative characterization of the SR than the conventional method.

Suppose that the grid-connected MG has an initial condition  $\mathbf{x}(0) = [-0.5, 1]^\top$ , due to a disturbance. Such an initial condition is inside  $\mathcal{S}_{1.01}$ , but outside  $\mathcal{S}'$ . Therefore,  $\mathcal{S}_{1.01}$  can conclude that the system trajectory tends to its equilibrium point, whereas  $\mathcal{S}'$  can conclude nothing about the system's asymptotic behavior under such a disturbance. The

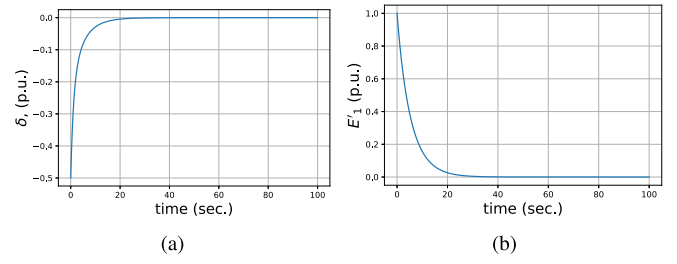


Fig. 8. Time-domain simulation for the grid-connected MG with initial conditions  $\delta'_1(0) = -0.5$  rad. and  $E'_1(0) = 1$  p.u.

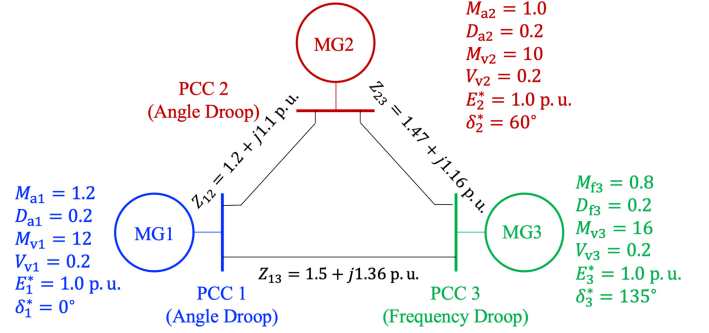


Fig. 9. Three Networked Microgrids with Mixed Dynamics.

time-domain simulation shown in Figure 8 confirms that all state variables tend to their pre-dispatched steady-state values.

### B. Three Networked Microgrids With Mixed Dynamics

Figure 9 shows a three-MG interconnection with mixed interface dynamics: the angle droop control is deployed in the PE interfaces of MGs 1 and 2, whereas the frequency droop control is deployed in the PE interfaces of MG 3. Since  $M_{vk} \gg M_{ak}$  for  $k = 1, 2$  and  $M_{v3} \gg M_{f3}$ , the time-scale separation is assumed [2]. We focus on the asymptotic behavior of phase angle and frequency. The user-defined parameters of Algorithm 3 are listed in Table I.

1) *Learned Lyapunov Function*: After taking 23737.53 seconds, Algorithm 3 outputs a Lyapunov function  $V_{\theta^*}$  valid in  $\mathcal{B}_{0.4}$ . Given  $\delta'_3 = 0$  and  $\omega'_3 = 0$ ,  $V_{\theta^*}$  and  $\dot{V}_{\theta^*}$  are visualized in Figure 10, where it is observed that  $V_{\theta^*} > 0$  and  $\dot{V}_{\theta^*} < 0$  in  $\mathcal{B}_{0.4}$ , suggesting  $V_{\theta^*}$  behaves like a Lyapunov function.

2) *Estimated Stability Region*: With the learned Lyapunov function, Algorithm 3 provides an estimated SR  $\mathcal{S}_{0.37}$ . Figure 11-(a) visualizes  $\mathcal{S}_{0.37}$  and  $\mathcal{B}_{0.4}$  in the  $\delta'_1$ - $\delta'_2$  space with  $\delta'_3 = 0.37$  and  $\omega'_3 = -0.14$ , where the red-solid circle is the boundary of  $\mathcal{S}_{0.37}$ , and the red-dash circle is the boundary of  $\mathcal{B}_{0.4}$ . The  $\text{SREst}$  function suggests that  $d^*$  in (18) is 0.37 which is attained when  $\mathbf{x}$  is  $[-0.07, 0.01, 0.37, -0.14]^\top$ . Figure 11-(a) shows that the boundary of  $\mathcal{S}_{0.37}$  touches the boundary of  $\mathcal{B}_{0.4}$  at point  $(-0.07, 0.01)$ .

3) *Comparison*: Denote by  $\mathcal{S}''$  the SR estimated based on the Lyapunov function proposed in [5]. The blue-solid circle in Figure 11-(b) represents the boundary of  $\mathcal{S}''$  in the  $\delta'_1$ - $\delta'_2$  plane, given  $\delta'_3 = \omega'_3 = 0$ . Suppose that the pre-event condition  $\mathbf{x}(0)$  is  $[0.1, -0.1, 0, 0]^\top$ . Since  $\mathbf{x}(0)$  is inside  $\mathcal{S}_{0.37}$  but outside  $\mathcal{S}''$ , one can conclude that *all states tend to the equilibrium based*



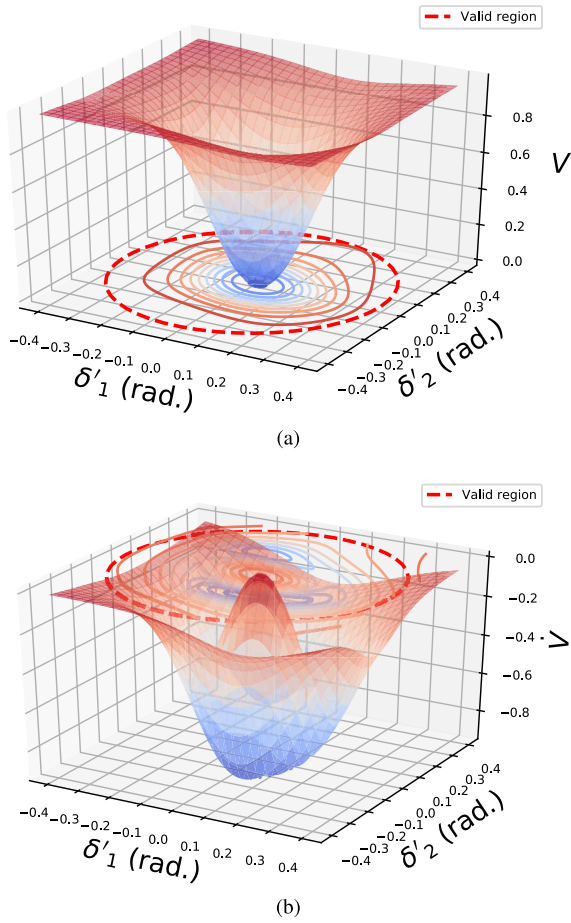


Fig. 10. (a) Lyapunov function and (b) time derivative for 3 networked MGs.

TABLE II  
DISTRIBUTION LINE PARAMETERS

| From-node # | To-node # | $R$ (p.u.) | $X$ (p.u.) |
|-------------|-----------|------------|------------|
| 18          | 135       | 1.2030     | 1.1034     |
| 13          | 152       | 1.0300     | 0.7400     |
| 151         | 300       | 1.4512     | 1.3083     |
| 54          | 94        | 1.5042     | 1.3554     |
| 97          | 197       | 1.4680     | 1.1550     |

on  $S_{0,37}$ , while the asymptotic behavior the system cannot be assessed by  $S''$  with  $\mathbf{x}(0)$ . The time-domain simulation confirms that all state variables indeed converge to their post-event steady-state values. In Figures 11-(a) and 11-(b), the reason why we observe different security (valid) regions estimated from the proposed method is that  $\delta_3'$  and  $\delta_4'$  are set to different values in these two cases.

### C. IEEE 123-Node Test Feeder

Figure 13 shows a 123-node distribution system [40] which is partitioned into 5 networked MGs [2]. We assume that each MG is managed by its MGCC and connects to the grid via a PE interface with angle droop control [2]. The impedances of the interconnection distribution lines are reported in Table II. The control parameters and pre-dispatched setpoints are listed in Table III. The user-defined parameters in Algorithm 3 are

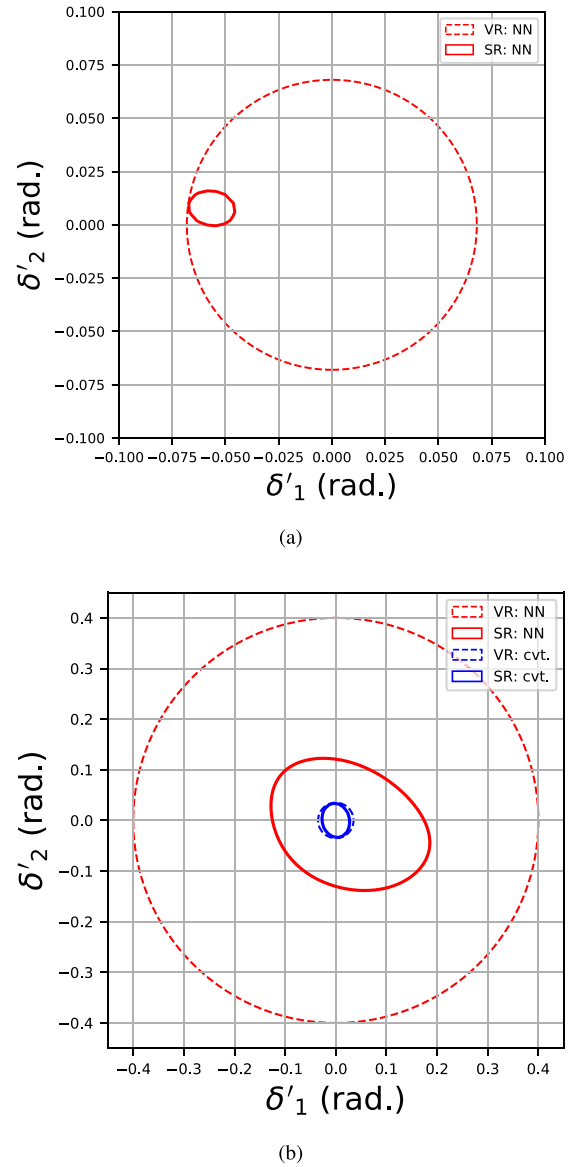


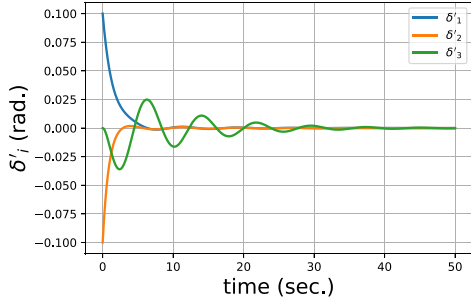
Fig. 11. (a) SR and VR around the touching point. (b) Comparison between the proposed (NN) and conventional (cvt.) methods.

TABLE III  
CONTROL PARAMETERS, PRE-EVENT MEASUREMENTS AND POST-EVENT SETPOINTS OF THE IEEE 123-NODE FEEDER

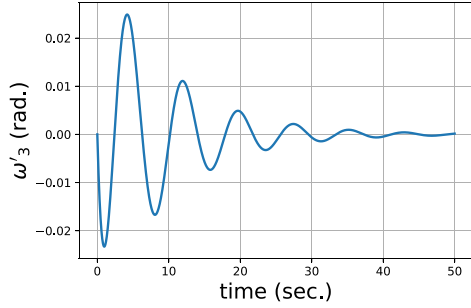
|                             | MG1 | MG2     | MG3    | MG4    | MG5    |
|-----------------------------|-----|---------|--------|--------|--------|
| $M_{ak}$                    | 1.2 | 1       | 0.8    | 1      | 1.2    |
| $D_{ak}$                    | 1.2 | 1.2     | 1.2    | 1.2    | 1.2    |
| $M_{vk}$                    | 12  | 10      | 16     | 10     | 12     |
| $D_{vk}$                    | 0.2 | 0.2     | 0.2    | 0.2    | 0.2    |
| Pre-event $\delta_k$ (rad.) | 0   | -0.8472 | 2.3062 | 0.5936 | 0.7732 |
| $\delta_k^*$ (rad.)         | 0   | -1.0472 | 2.3562 | 0.5236 | N/A    |
| $E_k^*$ (p.u.)              | 1.0 | 1.0     | 1.0    | 1.0    | N/A    |

reported in Table I. Note that the time-scale separation is assumed, as  $M_{vk} \gg M_{ak}$  for  $k = 1, 2, \dots, 5$  in Table III.

1) *Online Application of Estimated Stability Region:* Suppose that at time  $t = 0$ , MG 5 enters an islanded mode and the DSO would like to know if the remaining 4 networked MGs can be stabilized at a pre-dispatched operating point. During offline planning, Algorithm 3 computes a Lyapunov

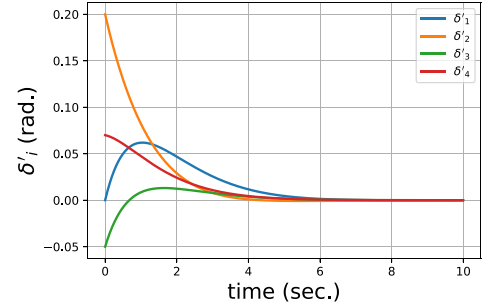


(a)

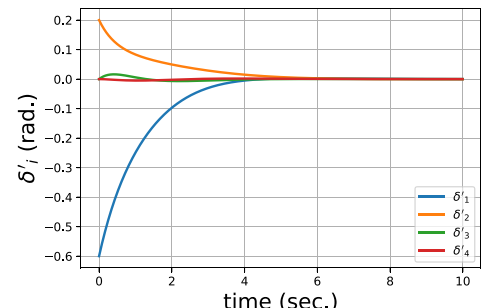


(b)

Fig. 12. Time-domain simulation of the 3 networked MGs with  $\mathbf{x}(0) = [0.1, -0.1, 0, 0]^T$ : (a) angle deviation and (b) frequency deviation.



(a)



(b)

Fig. 14. Time-domain simulation of interface variables in the 123-node feeder: (a) with MG 5 islanded; (b) with  $\mathbf{x}(0) = [-0.6, 0.2, 0, 0]^T$  rad.

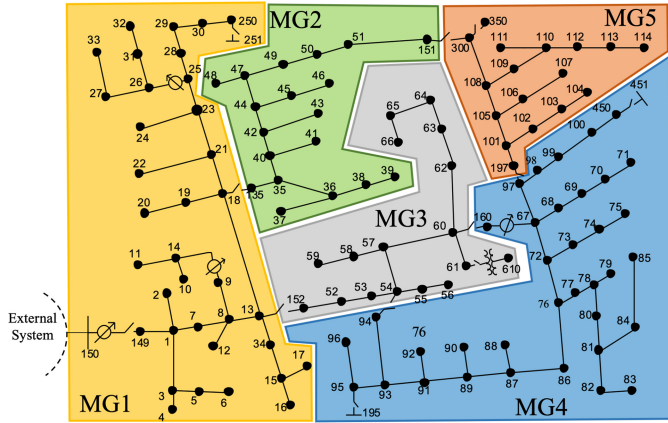
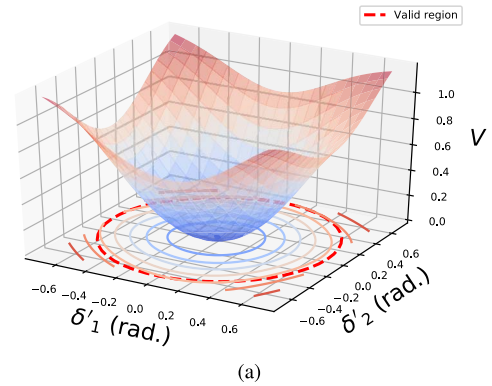


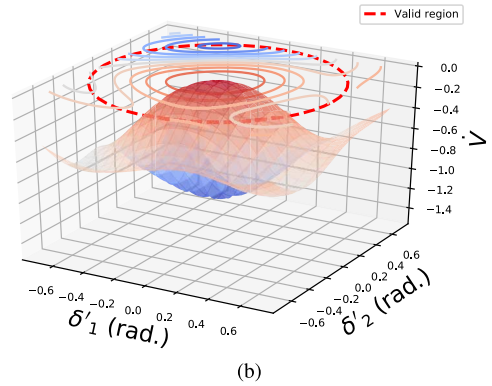
Fig. 13. IEEE 123-node Test Feeder [2].

function  $V_{\theta^*}$  and a stability region  $S_{0.69}$  for the contingency.  $S_{0.69}$  can be leveraged during real-time operation, in order to determine if the remaining MGs can tolerate the disturbance due to islanding of MG 5. The initial condition  $\mathbf{x}(0)$  can be obtained by collecting pre-event measurements at the MG interfaces. In this case study,  $V_{\theta^*}(\mathbf{x}(0)) = 0.12 < 0.69$ , suggesting that  $\mathbf{x}(0) \in S_{0.69}$ . Thus, without any simulation, the DSO can almost instantaneously conclude that all interface variables tend to their pre-dispatched values. Such a conclusion is confirmed by the time-domain simulation in Figure 14-(a).

2) *Learned Lyapunov Function and Estimated Stability Region*: It takes 2901.69 seconds to learn the Lyapunov function  $V_{\theta^*}$ . Figure 15 visualizes  $V_{\theta^*}$  and  $\dot{V}_{\theta^*}$ . With  $V_{\theta^*}$ , SRESt computes a stability region which is visualized



(a)



(b)

Fig. 15. (a)  $V_{\theta^*}$  and (b)  $\dot{V}_{\theta^*}$  in the 123-node feeder.

in Figure 16 and it suggests that the solution to (18) is  $[-0.66, 0.03, 0.06, 0.22]^T$ . Figure 16-(a) visualizes the region  $S_{0.69}$  in the  $\delta'_1$ - $\delta'_2$  plane with  $\delta'_3 = 0.06$  and  $\delta'_4 = 0.22$ . It is

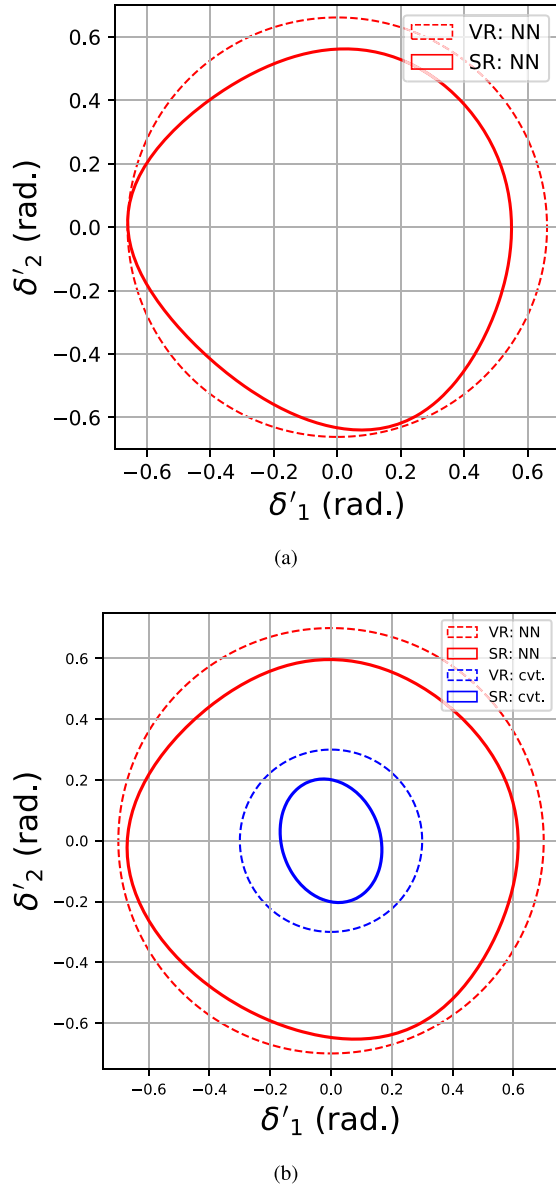


Fig. 16. (a) SR and VR around the touching point with  $\delta'_3 = 0.06$  and  $\omega'_3 = 0.22$ ; (b) comparison between the proposed (NN) and conventional (cvt.) methods with  $\delta'_3 = \omega'_4 = 0$ .

observed that the touching point of the boundaries of  $\mathcal{S}_{0.69}$  and  $\mathcal{B}_{0.7}$  is  $(-0.66, 0.03)$ .

3) *Comparison*: The comparison between the stability region estimated based on the proposed and conventional methods is shown in Figure 16-(b). Denoted by  $\mathcal{S}'''$  the SR estimated based on the quadratic Lyapunov function. Suppose that pre-event operating condition  $\mathbf{x}(0)$  is  $[-0.6, 0.2, 0, 0]^T$ . Such a condition is outside  $\mathcal{S}'''$  but inside  $\mathcal{S}_{0.69}$ . Therefore,  $\mathcal{S}_{0.69}$  can conclude that *the system trajectory will converge to the equilibrium whereas  $\mathcal{S}'''$  cannot*. The time-domain simulation shown in Figure 14-(b) confirms the convergence of the states given the pre-event condition.

Next, we compare the proposed approach with the SOS-based approach [21]. When being applied to the IEEE 123-node feeder, the procedure of Lyapunov function searching

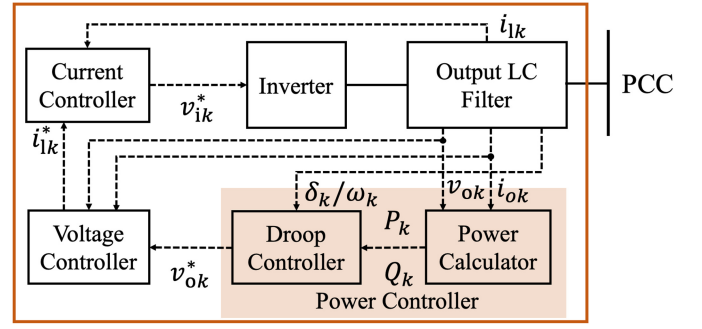


Fig. 17. Block diagram of the  $k$ -th PE interface.

in [21] stalls in a personal laptop, as it involves the operation of a  $43034 \times 43034$  (26.4 GB) array. However, a neural network-structured Lyapunov function can be obtained by the proposed approach in the same personal laptop, as shown in Section IV-C. Therefore, it can be observed that the proposed approach requires less computation power in addressing the 123-node feeder, compared with the SOS-based approach in [21].

## V. CONCLUDING REMARKS

In this paper, we propose a TSA tool for networked microgrids based on tailor-designed Neural Lyapunov methods. Assessing transient stability is formulated as a problem of estimating the stability region of networked microgrids. We use neural networks to learn a Lyapunov function in the state space. The optimal stability region is estimated based on the function learned, and it can be used for both offline design and online operation. The effectiveness of the proposed TSA tool is tested and validated in 3 scenarios: 1) a grid-connected microgrid, 2) a three networked microgrids with heterogeneous dynamics, and 3) the IEEE 123-node test feeder. Future work will investigate algorithms to speed up the computation for learning a Lyapunov function in larger networked microgrids.

## APPENDIX A

### DETAILED MODEL OF PE INTERFACES

The typical components of a PE interface are summarized in Figure 17. Next, we describe the dynamics of the PE interface by presenting the dynamics of each component in Figure 17.

#### A. Power Controller

As shown in Figure 17, the power controller includes two functional blocks: a power calculator, and a droop controller. The power calculator takes as inputs the instantaneous voltage  $v_{ok}$  and current  $i_{ok}$  from the LC filter, and it aims to compute the fundamental components of real and reactive power  $P_k$  and  $Q_k$  of the  $k$ -th interface. The dynamics of the power calculator can be described by [41]

$$\dot{P}_k = -\omega_c P_k + \omega_c (v_{odk} i_{odk} + v_{oqk} i_{oqk}) \quad (26a)$$

$$\dot{Q}_k = -\omega_c Q_k + \omega_c (v_{oqk} i_{odk} - v_{odk} i_{oqk}) \quad (26b)$$

where  $\omega_c$  denotes the cut-off frequency of the low-pass filters in the power calculator [41]; and  $v_{odk}$  ( $i_{odk}$ ) and  $v_{oqk}$  ( $i_{oqk}$ ) are

the direct and quadrature components of  $v_{ok}$  ( $i_{ok}$ ). The droop controller leverages some local signals as the power balance indicators [2], and it tunes the interface response according to the measurements of these signals. Common selections of these local signals include frequency, voltage magnitude and voltage angle that are measured at the microgrid PCC. Specifically, the *frequency droop control* takes the terminal frequency  $\omega_k$  as the balance indicator for real power [2]. Such a control strategy requires no communication between PE interfaces and it introduces the following dynamics:

$$\dot{\delta}_k = \omega_k - \omega_n, \quad M_{fk}\dot{\omega}_k = -D_{fk}(\omega_k - \omega_n) + P_k^* - P_k \quad (27)$$

where  $\omega_n$  denotes the nominal frequency;  $M_{fk}$  and  $D_{fk}$  are control parameters for the frequency droop controller; and  $P_k^*$  and  $Q_k^*$  are dispatched by the DSO. The *angle droop control* considers the voltage phase angle  $\delta_k$  as the balance indicator [2], [42], [43]. Though the angle droop control requires communication, it provides better frequency regulation [42], compared with the frequency droop control. The angle droop control will introduce the following dynamics:

$$M_{ak}\dot{\delta}_k + (\delta_k - \delta_k^*) = D_{ak}(P_k^* - P_k), \quad \omega_k = \dot{\delta}_k + \omega_n \quad (28)$$

where  $M_{ak}$  and  $D_{ak}$  denote control parameters of the angle droop controllers; and  $\delta_k^*$  and  $P_k^*$  are dispatched by the DSO. Besides, the droop controller tunes the setpoint  $v_{ok}^*$  of the voltage controller in Figure 17 according to

$$M_{vk}\dot{v}_{ok}^* = D_{vk}(Q_k^* - Q_k) - (v_{odk}^* - E_k^*), \quad v_{oqk}^* = 0 \quad (29)$$

where  $v_{odk}^*$  and  $v_{oqk}^*$  are the direct and quadrature components of  $v_{ok}^*$ , respectively; and  $E_k^*$ ,  $Q_k^*$  are dispatched by the DSO.

### B. Voltage and Current Controllers

The dynamics of the voltage and current controllers in Figure 17 can be described by the following differential and algebraic equations [41]:

$$\dot{\xi}_{dk} = v_{odk}^* - v_{odk}, \quad \dot{\xi}_{qk} = v_{oqk}^* - v_{oqk}, \quad (30a)$$

$$\dot{\psi}_{dk} = i_{ldk}^* - i_{ldk}, \quad \dot{\psi}_{qk} = i_{lqk}^* - i_{lqk}, \quad (30b)$$

$$i_{ldk}^* = K_{ivk}\xi_{dk} + F_k i_{odk} + K_{pvk}(v_{odk}^* - v_{odk}) - \omega_n C_{fk} v_{oqk} \quad (30c)$$

$$i_{lqk}^* = K_{ivk}\xi_{qk} + F_k i_{oqk} + K_{pvk}(v_{oqk}^* - v_{oqk}) + \omega_n C_{fk} v_{odk} \quad (30d)$$

$$v_{idk}^* = K_{ick}\psi_{dk} + K_{pck}(i_{ldk}^* - i_{ldk}) - \omega_n L_{fk} i_{lqk} \quad (30e)$$

$$v_{iqk}^* = K_{ick}\psi_{qk} + K_{pck}(i_{lqk}^* - i_{lqk}) + \omega_n L_{fk} i_{ldk} \quad (30f)$$

where  $\xi_{dk}$  and  $\xi_{qk}$  are state variables of the voltage controller;  $\psi_{dk}$  and  $\psi_{qk}$  are state variables of the current controller;  $K_{ivk}$ ,  $F_k$ , and  $K_{pvk}$  are control parameters of the voltage controller;  $C_{fk}$  and  $L_{fk}$  are capacitance and inductance of the output LC filter;  $i_{ldk}^*$ ,  $i_{lqk}^*$  are the setpoints of the current controller; and  $v_{idk}^*$ ,  $v_{iqk}^*$  are the setpoints of the inverter in Figure 17.

### C. Output Filter

With the switching dynamics ignored in the inverter, we have  $v_{idk}^* = v_{idk}$  and  $v_{iqk}^* = v_{iqk}$ , where  $v_{idk}$  and  $v_{iqk}$  are

TABLE IV  
PE INTERFACE PARAMETERS OF FIGURE 2 OF THE MANUSCRIPT

|           |               |            |           |            |              |
|-----------|---------------|------------|-----------|------------|--------------|
| Parameter | $r_{line}$    | $L_{line}$ | $L_{f1}$  | $C_{f1}$   | $r_{f1}$     |
| Value     | 0.38 $\Omega$ | 2.2 mH     | 1.35 mH   | 50 $\mu$ F | 0.1 $\Omega$ |
| Parameter | $\omega_{c1}$ | $K_{pv1}$  | $K_{iv1}$ | $K_{pc1}$  | $K_{ic1}$    |
| Value     | 31.41 rad/s   | 0.05       | 390       | 10.5       | 1.6e3        |
| Parameter | $F_1$         | $T_{a1}$   | $T_{v1}$  | $D_{a1}$   | $D_{v1}$     |
| Value     | 0.75          | 1 p.u.     | 10 p.u.   | 0.2 p.u.   | 0.03 p.u.    |

two state variables of the output filter associated with the  $k$ -th interface. The following differential equations describe the dynamics of the output filter associated with the  $k$ -th PE interface [41]:

$$L_{fk}\dot{i}_{dk} = -r_{fk}i_{dk} + \omega_k L_{fk}i_{qk} + v_{idk} - v_{odk} \quad (31a)$$

$$L_{fk}\dot{i}_{qk} = -r_{fk}i_{qk} + \omega_k L_{fk}i_{dk} + v_{iqk} - v_{oqk} \quad (31b)$$

$$C_{fk}\dot{v}_{odk} = \omega_k C_{fk}v_{oqk} + i_{dk} - i_{odk} \quad (31c)$$

$$C_{fk}\dot{v}_{oqk} = \omega_k C_{fk}v_{odk} + i_{qk} - i_{oqk} \quad (31d)$$

where  $r_{fk}$  is the resistance of the output filter; and  $v_{odk}$ ,  $v_{oqk}$ ,  $i_{odk}$  and  $i_{oqk}$  are the variables interfacing with the distribution system network. Equations (26)-(31) define detailed dynamics of a PE interface.

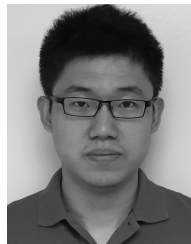
## APPENDIX B SIMULATION PARAMETERS OF FIGURE 2

This section aims to provide parameters for the simulation shown in Figure 2. The topology of the system is shown in Figure 5 (the parameters marked in Figure 5 are *not* for this test case). The voltage base  $V_b = 381.05$  V and the power base  $S_b = 10$  kVA. The nominal frequency  $\omega_n$  is 50 Hz. The detailed model of the PE interface includes (26), (28)-(31) with parameters [41] listed in Table IV. The resistance  $r_{line}$  and inductance  $L_{line}$  of the tie line are shown in Table IV. At Bus 2, a three-phase-to-ground fault occurs at the 1-st second and it is cleared 3 cycles later. Figure 2 shows the system response to such an event.

## REFERENCES

- [1] T. Huang, "Physical and cyber anomaly management in massively digitized power systems," Ph.D. dissertation, Dept. Electr. Comput. Eng., Texas A&M Univ., College Station, TX, USA, 2021.
- [2] Y. Zhang and L. Xie, "A transient stability assessment framework in power electronic-interfaced distribution systems," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 5106–5114, Nov. 2016.
- [3] 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.
- [4] T. Huang, S. Gao, X. Long, and L. Xie, "A neural Lyapunov approach to transient stability assessment in interconnected microgrids," in *Proc. 54th Hawaii Int. Conf. Syst. Sci.*, 2021, pp. 1–10.
- [5] H.-D. Chiang, "Study of the existence of energy functions for power systems with losses," *IEEE Trans. Circuits Syst.*, vol. 36, no. 11, pp. 1423–1429, Nov. 1989.
- [6] H.-D. Chiang, C.-C. Chu, and G. Cauley, "Direct stability analysis of electric power systems using energy functions: Theory, applications, and perspective," *Proc. IEEE*, vol. 83, no. 11, pp. 1497–1529, Nov. 1995.
- [7] A.-A. Fouad and V. Vittal, "The transient energy function method," *Int. J. Electr. Power Energy Syst.*, vol. 10, no. 4, pp. 233–246, 1988.
- [8] A.-A. Fouad and V. Vittal, *Power System Transient Stability Analysis Using the Transient Energy Function Method*. New York, NY, USA: Pearson Educ., 1991.

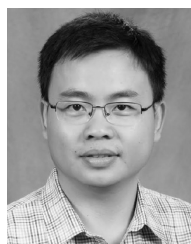
- [9] Y.-H. Moon, E.-H. Lee, and T.-H. Roh, "Development of an energy function reflecting the transfer conductances for direct stability analysis in power systems," *IEEE Proc. Gener. Transm. Distrib.*, vol. 144, no. 5, pp. 503–509, 1997.
- [10] Y.-H. Moon, B.-H. Cho, T.-H. Rho, and B.-K. Choi, "The development of equivalent system technique for deriving an energy function reflecting transfer conductances," *IEEE Trans. Power Syst.*, vol. 14, no. 4, pp. 1335–1341, Nov. 1999.
- [11] L. Wehenkel, T. Van Cutsem, and M. Ribbens-Pavella, "An artificial intelligence framework for online transient stability assessment of power systems," *IEEE Trans. Power Syst.*, vol. 4, no. 2, pp. 789–800, May 1989.
- [12] K. Sun, S. Likhate, V. Vittal, V. S. Kolluri, and S. Mandal, "An online dynamic security assessment scheme using phasor measurements and decision trees," *IEEE Trans. Power Syst.*, vol. 22, no. 4, pp. 1935–1943, Nov. 2007.
- [13] M. He, J. Zhang, and V. Vittal, "Robust online dynamic security assessment using adaptive ensemble decision-tree learning," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4089–4098, Nov. 2013.
- [14] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, vol. 199. Englewood Cliffs, NJ, USA: Prentice Hall, 1991.
- [15] D. Issicaba, J. A. P. Lopes, and M. A. da Rosa, "Adequacy and security evaluation of distribution systems with distributed generation," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1681–1689, Aug. 2012.
- [16] J. Alipoor, Y. Miura, and T. Ise, "Stability assessment and optimization methods for microgrid with multiple VSG units," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 1462–1471, Mar. 2018.
- [17] Y. Zhang, L. Xie, and Q. Ding, "Interactive control of coupled microgrids for guaranteed system-wide small signal stability," *IEEE Trans. Smart Grid*, vol. 7, no. 2, pp. 1088–1096, Mar. 2016.
- [18] P. Shamsi and B. Fahimi, "Stability assessment of a DC distribution network in a hybrid micro-grid application," *IEEE Trans. Smart Grid*, vol. 5, no. 5, pp. 2527–2534, Sep. 2014.
- [19] M. Kabalan, P. Singh, and D. Niebur, "Large signal Lyapunov-based stability studies in microgrids: A review," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2287–2295, Sep. 2017.
- [20] M. Kabalan, P. Singh, and D. Niebur, "Nonlinear Lyapunov stability analysis of seven models of a DC/AC droop controlled inverter connected to an infinite bus," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 772–781, Jan. 2019.
- [21] T. Huang, H. Sun, K. J. Kim, D. Nikovski, and L. Xie, "A holistic framework for parameter coordination of interconnected microgrids against disasters," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, 2020, pp. 1–5.
- [22] L. Aolaritei, S. Bolognani, and F. Dörfler, "Hierarchical and distributed monitoring of voltage stability in distribution networks," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6705–6714, Nov. 2018.
- [23] R. El Helou, D. Kalathil, and L. Xie, "Communication-free voltage regulation in distribution networks with deep PV penetration," in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2020, p. 10.
- [24] H. Zhu and H. J. Liu, "Fast local voltage control under limited reactive power: Optimality and stability analysis," *IEEE Trans. Power Syst.*, vol. 31, no. 5, pp. 3794–3803, Sep. 2016.
- [25] P. W. Sauer and M. A. Pai, *Power System Dynamics and Stability*, vol. 101. Upper Saddle River, NJ, USA: Prentice Hall, 1998.
- [26] T. Huang, M. Wu, and L. Xie, "Prioritization of PMU location and signal selection for monitoring critical power system oscillations," *IEEE Trans. Power Syst.*, vol. 33, no. 4, pp. 3919–3929, Jul. 2018.
- [27] T. Huang, B. Satchidanandan, P. R. Kumar, and L. Xie, "An online detection framework for cyber attacks on automatic generation control," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6816–6827, Nov. 2018.
- [28] T. Huang, N. M. Freris, P. R. Kumar, and L. Xie, "Localization of forced oscillations in the power grid under resonance conditions," in *Proc. 52nd Annu. Conf. Inf. Sci. Syst. (CISS)*, 2018, pp. 1–5.
- [29] T. Huang, N. M. Freris, P. R. Kumar, and L. Xie, "A synchrophasor data-driven method for forced oscillation localization under resonance conditions," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 3927–3939, Sep. 2020.
- [30] T. Huang, B. Wang, J. Ramos-Ruiz, P. Enjeti, P. R. Kumar, and L. Xie, "Detection of cyber attacks in renewable-rich microgrids using dynamic watermarking," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, 2020, pp. 1–5.
- [31] A. Ng, "CS229 lecture notes," Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, 2000.
- [32] M. S. Modarresi, T. Huang, H. Ming, and L. Xie, "Robust phase detection in distribution systems," in *Proc. IEEE Texas Power Energy Conf. (TPEC)*, 2017, pp. 1–5.
- [33] S. Gao, J. Avigad, and E. M. Clarke, " $\delta$ -complete decision procedures for satisfiability over the reals," in *Proc. Int. Joint Conf. Autom. Reason.*, 2012, pp. 286–300.
- [34] Y.-C. Chang, N. Roohi, and S. Gao, "Neural Lyapunov control," 2020. [Online]. Available: arXiv:2005.00611.
- [35] T. L. Vu and K. Turitsyn, "Lyapunov functions family approach to transient stability assessment," *IEEE Trans. Power Syst.*, vol. 31, no. 2, pp. 1269–1277, Mar. 2016.
- [36] D. A. Knoll and D. E. Keyes, "Jacobian-free Newton–Krylov methods: A survey of approaches and applications," *J. Comput. Phys.*, vol. 193, no. 2, pp. 357–397, 2004.
- [37] P. Kundur, "Power system stability," in *Power System Stability and Control*. Boca Raton, FL, USA: CRC Press, 2007.
- [38] A. Mešanović *et al.*, "Structured controller parameter tuning for power systems," *Control Eng. Pract.*, vol. 101, Aug. 2020, Art. no. 104490.
- [39] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.
- [40] K. P. Schneider *et al.*, "Analytic considerations and design basis for the IEEE distribution test feeders," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3181–3188, May 2018.
- [41] 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.
- [42] R. R. Kolluri, I. Mareels, T. Alpcan, M. Brazil, J. de Hoog, and D. A. Thomas, "Power sharing in angle droop controlled microgrids," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4743–4751, Nov. 2017.
- [43] Y. Pan, L. Chen, X. Lu, J. Wang, F. Liu, and S. Mei, "Stability region of droop-controlled distributed generation in autonomous microgrids," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2288–2300, Mar. 2019.



**Tong Huang** (Member, IEEE) received the Ph.D. degree in electrical engineering from Texas A&M University, College Station, TX, USA, in 2021, where he is currently a Postdoctoral Researcher. From September 2018 to December 2018, he was a visiting Ph.D. student with the Laboratory for Information and Decision Systems, MIT. His industry experience includes an internship with ISO-New England in 2018 and with Mitsubishi Electric Research Laboratories in 2019. He received the Best Paper Award at the 2020 IEEE Power and Energy Society General Meeting, the Best Paper Award at the 54-th Hawaii International Conference on System Sciences, the Thomas W. Powell'62 and Powell Industries Inc., Fellowship, and the Texas A&M Graduate Teaching Fellowship.



**Sicun Gao** received the Ph.D. degree from Carnegie Mellon University and was a Postdoctoral Researcher with MIT. He is an Assistant Professor of Computer Science and Engineering with the University of California at San Diego. He works on search and optimization algorithms for improving the reliability of automation and autonomous systems. He is a recipient of the Air Force Young Investigator Award, the NSF Career Award, and the Silver Medal for the Kurt Godel Research Prize.



**Le Xie** (Senior Member, IEEE) received the B.E. degree in electrical engineering from Tsinghua University, Beijing, China, in 2004, the M.S. degree in engineering sciences from Harvard University, Cambridge, MA, USA, in 2005, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, in 2009. He is currently a Professor with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA. His research interests include modeling and control of large-scale complex systems, smart grids application with renewable energy resources, and electricity markets.