# Distributed Learning of Neural Lyapunov Functions for Large-Scale Networked Dissipative Systems

Amit Jena, Tong Huang, S. Sivaranjani, Dileep Kalathil, and Le Xie

Abstract—This paper considers the problem of characterizing the stability region of a large-scale networked system comprised of dissipative nonlinear subsystems, in a distributed and computationally tractable way. One standard approach to estimate the stability region of a general nonlinear system is to first find a Lyapunov function for the system and characterize its region of attraction as the stability region. However, classical approaches, such as sum-of-squares methods and quadratic approximation, for finding a Lyapunov function either do not scale to large systems or give very conservative estimates for the stability region. In this context, we propose a new distributed learning based approach by exploiting the dissipativity structure of the subsystems. Our approach has two parts: the first part is a distributed approach to learn the storage functions (similar to the Lyapunov functions) for all the subsystems, and the second part is a distributed optimization approach to find the Lyapunov function for the networked system using the learned storage functions of the subsystems. We demonstrate the superior performance of our proposed approach through extensive case studies in microgrid networks.

Index Terms—Lyapunov function, stability assessment, distributed learning, dissipativity

### I. Introduction

Large-scale networked systems formed by interconnection of a several nonlinear subsystems are encountered in several practical applications such as infrastructure networks including power networks, transportation networks, and communications networks. For example, in power networks, a collection of renewable generators, storage, and loads in a small area can be modeled as a single microgrid with nonlinear dynamics, and a number of such microgrids form a network which balances the supply and demand of electric power in a larger geographical area. Analyzing the stability margins of the dynamics of such infrastructure systems is crucial in ensuring their safe and reliable operation. One standard approach to estimate the stability region of a general nonlinear system is to first find a Lyapunov function for the system and characterize its region of attraction as the stability region. The sum-of-squares approach is one popular method for finding a Lyapunov function for a dynamical system [1]–[5]. However, sum-of-squares approaches typically do not scale well to large systems, since a large number of semidefinite programs need to be solved for

Amit Jena, Dileep Kalathil and Le Xie are with the Department of Electrical and Computer Engineering at Texas A& M University, College Station, TX, USA. Email:{amit\_jena, dileep.kalathil, le.xie}@tamu.edu.

Tong Huang is with the MIT Laboratory of Information & Decision Systems, Cambridge, MA, USA. Email:tongh@mit.edu.

S. Sivaranjani is with the School of Industrial Engineering at Purdue University, West Lafayette, IN, USA. Email:sseetha@purdue.edu.

the sum-of-squares decomposition of polynomial systems even with a few states [1]. Another approach is to employ local linearizations and use quadratic approximations to find Lyapunov functions. However, this approach is typically conservative in the sense that stability can only be certified in a small vicinity of an equilibrium point of a nonlinear system, which may be insufficient to cover the normal range of operation in several practical applications such as microgrids in power systems [6].

Due to the large size and continually expanding scale of many real-world networked systems, it is especially important to develop distributed approaches to assess the stability of such systems. Analyzing the stability of a general networked system in a distributed manner from the stability assessment of its subsystems is a complex open problem. Such a distributed stability assessment, however, is possible for a class of networked systems where each subsystem satisfies passivity or dissipativity properties [7]–[10]. However, these results are typically confined to local stability assessments based on linearized system dynamics. Thus, how to find a Lyapunov function (and the corresponding stability region) for a large scale networked system in a distributed and computationally tractable way remains an important open problem.

In this paper, we consider the problem of finding a Lyapunov function and estimating its associated region of attraction in a distributed manner for a large scale networked system comprised of a number of subsystems. We assume that each subsystem is nonlinear and satisfies the dissipativity property [9]. Our approach has two stages: in the first stage, we develop a distributed learning approach to learn the storage functions (similar to the Lyapunov functions) for all the subsystems; in the second stage, we develop a distributed optimization approach to find the Lyapunov function for the networked system using the learned storage functions of the subsystems. Our main algorithm essentially is an iteration over these two parts until some convergence criterion is satisfied.

Specifically, in the first stage, we use a neural network approximation to learn a storage function for each subsystem such that the subsystem satisfies the dissipativity property. Our approach is inspired from the recent works which use the data-based function approximation capability of neural networks to *learn* a Lyapunov function for nonlinear systems [6], [11]–[13] in a centralized way. In addition to this empirical learning, we also use a satisfiability modulo theories (SMT) solver based falsifier that finds counterexamples to verify this local dissipativity property using the storage function; when no counterexample is found by the falsifier, the subsystem is provably dissipative [12]. In the second stage, we use the fact that the stability of the networked system can be guaranteed

by verifying some conditions on the dissipativity of the subsystems and the coupling between dissipative subsystems [9]. For efficient implementation, we formulate and solve this as a distributed optimization problem using an alternating direction method of multipliers (ADMM) procedure which iteratively updates the storage function for each subsystem until the network-level stability conditions are satisfied. This step also allows us to compute the Lyapunov function for the networked system and estimate its region of attraction. Additionally, we learn local controllers at the subsystem-level that enhance the stability region of the networked system.

We demonstrate the performance of our proposed distributed learning approach for stability assessment through extensive case studies in microgrid networks. Our choice of microgrid networks is motivated by their importance in real-world power systems and their natural structure as a network of smaller nonlinear subsystems [14]–[16]. In recent years, the stability assessment problem for a network of microgrids has also attracted a lot of attention from the power systems and controls community [17]–[19].

A preliminary version of this work was published as a conference paper [20]. In this work, we significantly expand and extend the results in [20]. In particular, we learn a local controller for each subsystem that improves the stability of the network system and makes it robust to bounded disturbances. We also consider a larger network based case study which clearly illustrates the superior performance our distributed learning approach both in terms of finding a larger stability region and in reduced training/computation time. In summary, the key contributions of this paper are as follows:

- We develop a distributed approach to learn the Lyapunov function and the stability region of a large scale networked system by exploiting the dissipativity property of its subsystems and leveraging recent advances in neural network based nonlinear function approximation/learning using data.
- We learn a local controller for each subsystem that further improves the stability of the entire networked system.

This paper is organized as follows. Section II gives the preliminaries and basic problem formulation, Section III describes our distributed learning based algorithm, Section IV presents the case studies, and Section V concludes with future research directions.

## II. PROBLEM FORMULATION

We consider a large scale nonlinear networked system formed by the interconnection of n nonlinear subsystems. The dynamics of the ith subsystem is given by

$$\dot{x}_i = f_i(x_i, u_i, v_i), \quad y_i = h_i(x_i, u_i, v_i),$$
 (1)

where  $x_i \in \mathbb{R}^{d_i^x}$ ,  $u_i \in \mathbb{R}^{d_i^u}$ ,  $v_i \in \mathbb{R}^{d_i^v}$ , and  $y_i \in \mathbb{R}^{d_i^y}$  are the state, primary input, secondary (local) input, and output of the subsystem i, respectively. The system dynamics and system output are specified by the continuously differentiable functions  $f_i$  and  $h_i$ . The state, primary input, secondary input and output of the networked system are then defined

as  $x = [x_1^\top, \dots, x_n^\top]^\top$ ,  $u = [u_1^\top, u_2^\top, \dots, u_n^\top]^\top$ ,  $v = [v_1^\top, v_2^\top, \dots, v_n^\top]^\top$ , and  $y = [y_1^\top, y_2^\top, \dots, y_n^\top]^\top$  respectively. The overall dynamics of the networked system is specified as

$$\dot{x} = f(x, u, v), \quad y = h(x, u, v), \tag{2}$$

where  $f = [f_1^\top, \dots, f_n^\top]^\top$ , and  $h = [h_1^\top, \dots, h_n^\top]^\top$ .

We assume that the subsystems are coupled through a primary and secondary input as

$$u = g^{\text{pri}}(y), \quad v_i = g_i^{\text{sec}}(y_i),$$
 (3)

where  $g^{\rm pri}$  and  $g_i^{\rm sec}$  specify the primary control law and secondary (local) control law, respectively. We adopt this two-level control structure for the following reason. Many large scale real-world networked systems, such as power systems, use a hierarchical control architecture with a dedicated secondary controller at each subsystem that maps its local output to a local control input [21], [22]. Such secondary controllers are known to critically improve the stability of the networked system using only local information [22]. Our separate modeling on the primary  $(u_i)$  and secondary  $(v_i)$  control inputs is motivated from such real-world systems. Later, we will also show experimental results which demonstrate the role of the secondary controller in improving the stability region of the networked system.

Without loss of generality, assume that the equilibrium point of (2)-(3) is given by  $(x^*, u^*, v^*, y^*) = o$ , where o is the origin. Also, denote  $(x_i^*, u_i^*, v_i^*, y_i^*)$  as  $o_i$ . For simplicity, we further restrict our consideration to a linear approximation of the coupling (3) between subsystems, given by

$$u = My, \quad v_i = K_i y_i, \tag{4}$$

where M and  $K_i$  are the Jacobian matrices of  $g^{\text{pri}}$  and  $g_i^{\text{sec}}$  evaluated at o and  $o_i$ , respectively. The overall dynamics of the networked system constituted by the interconnection of these n subsystems is now completely specified by (2) and (4) for all  $(x, u, v, y) \in B_o$ , where  $B_o$  is a neighborhood around the origin o.

We now define the *dissipativity* property for the subsystems.

**Definition 1** (Dissipativity [9]). The subsystem i described by (1) is said to be *dissipative* with respect to the *supply rate* function  $r_i(u_i, v_i, y_i)$  if there exists a continuously differentiable storage function  $\tilde{V}_i : \mathbb{R}^{d_i^x} \to \mathbb{R}$  such that  $\tilde{V}_i(o_i) = 0$ ,  $\tilde{V}_i(x_i) \geq 0$ , and

$$\dot{\tilde{V}}_i(x_i) = \nabla \tilde{V}_i(x_i)^{\top} f_i(x_i, u_i, v_i) \le r_i(u_i, v_i, y_i), \quad (5)$$

for all  $(x_i, u_i, v_i, y_i) \in B_{o_i}$ .

In this work, we assume that each subsystem satisfies the dissipativity property with respect to a quadratic supply rate function, as stated below.

**Assumption 1** (Dissipative subsystem). Every subsystem  $i \in \{1, 2, ..., n\}$ , described by (1), is *dissipative* with respect to the quadratic supply rate function given by

$$r_{i}(u_{i}, v_{i}, y_{i}) = \begin{bmatrix} u_{i} \\ y_{i} \\ v_{i} \\ y_{i} \end{bmatrix}^{\top} \begin{bmatrix} R_{i} & 0 \\ 0 & R_{i} \end{bmatrix} \begin{bmatrix} u_{i} \\ y_{i} \\ v_{i} \\ y_{i} \end{bmatrix}, \quad (6)$$

where 
$$R_i = \begin{bmatrix} R_i^{11} & R_i^{12} \\ R_i^{21} & R_i^{22} \end{bmatrix}$$
.

Next, we state the definitions of Lyapunov function and region of attraction (RoA). Note that we will use the terms RoA and stability region interchangeably in the following.

**Definition 2** (Lyapunov function and region of attraction [23]). For the system (2), a continuous differentiable scalar function  $V: \mathbb{R}^{d^x} \to \mathbb{R}, d^x = \sum_{i=1}^n d^{x_i}$ , is a *strict Lyapunov function* valid in a region  $\mathcal{V}_{\bar{c}} := \{x: \|x\|_2 \leq \bar{c}\}$  if the following conditions are satisfied:

(i) 
$$V(o) = 0$$
, (ii)  $V(x) > 0$  for all  $x \in \mathcal{V}_{\bar{c}} \setminus o$ , and (iii)  $\dot{V} = \nabla V(x)\dot{x} < 0$ , for all  $x \in \mathcal{V}_{\bar{c}} \setminus o$ .

Further, the region  $\mathcal{V}_{\bar{c}}$  is defined as the *region of attraction* (RoA) of the equilibrium point o, that is,  $x(0) \in \mathcal{V}_{\bar{c}} \implies \lim_{t \to \infty} x(t) = o$ .

The stability of the networked system described by (2) and (4) can be assessed by identifying a Lyapunov function satisfying Definition 2 and by characterizing the RoA. In general, sum-of-squares approaches [1]-[5] can be used to assess the stability of nonlinear systems. Alternatively, local linearizations of the dynamics may be used to compute a quadratic Lyapunov function [24]. However, these approaches often fail to find a meaningful Lyapunov function and RoA for large-scale, high dimensional, networked systems because of the following challenges. Firstly, sum-of-squares approaches do not scale well computationally and will quickly become intractable when the system dimension increases [1]-[5]. Secondly, sum-of-squares approaches and quadratic approximation based approaches are typically very conservative in their estimation of RoA even for small systems [6], [12]. This may lead to the design of conservative controllers and sub-optimal system operation.

In order to overcome these challenges, some recent works have exploited the data-based function approximation capability of neural network to *learn* a Lyapunov function for nonlinear systems [6], [11], [12], with impressive empirical performance in estimating the stability region for nontrivial nonlinear systems. However, the training time, number of samples needed, and computational complexity of these approaches also increase exponentially in the system dimension, which can be prohibitively high for a networked system constituted by a large number of smaller subsystems. In this work, we propose a distributed learning approach that exploits the dissipativity property of the subsystems to overcome these challenges and efficiently learn the Lyapunov function and stability region of the large-scale networked system.

## III. DISTRIBUTED LEARNING OF NEURAL LYAPUNOV FUNCTION FOR THE NETWORKED SYSTEM

In this section, we propose a two-stage distributed approach to learn a Lyapunov function for the networked system and characterize its RoA.

A. Lyapunov Function for the Networked System from the Storage Functions of the Subsystems

Our key idea is to reduce the dimension of the problem by leveraging the fact that the Lyapunov function for the networked system can be constructed from the storage functions of the subsystems under the dissipativity assumption (Assumption 1) [9]. We formally state this result below.

**Proposition 1.** Consider the networked system specified by (2) and (4). Let Assumption 1 hold, and  $r_i$  as specified in (6) be the supply rate function and  $\tilde{V}_i$  be the corresponding storage function of subsystem i. If

$$\begin{bmatrix} M \\ I \\ K \\ I \end{bmatrix}^{\top} \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} M \\ I \\ K \\ I \end{bmatrix} \preceq 0, \tag{7}$$

where

$$R = \begin{bmatrix} R_1^{11} & & R_1^{12} & & \\ & \ddots & & & \ddots & \\ & & R_N^{11} & & & R_N^{12} \\ & & & R_1^{21} & & & R_1^{22} \\ & & \ddots & & & \ddots & \\ & & & R_N^{21} & & & R_N^{22} \end{bmatrix}$$

 $K = \text{diag}([K_1, \dots, K_n])$  is a block diagonal matrix, and I is an identity matrix of appropriate order, then,

(i) the networked system has a stable equilibrium at o, and (ii)  $V(x) = \sum\limits_{i=1}^n \tilde{V}_i(x_i)$  is a Lyapunov function for the networked system.

*Proof.* The proof follows a direct extension of the result given in [9, Chapter 2] as follows. Consider the networked system specified by (1)-(4). Let  $\tilde{V}_i$  be a valid storage function for subsystem i with respect to the supply rate function  $r_i$  such that it satisfies the dissipativity condition (5), for all  $i \in \{1, ..., n\}$ .

Now, considering the function  $V(x) = \sum_{i=1}^{n} \tilde{V}_{i}(x_{i})$ , we have

$$\dot{V}(x) = \sum_{i=1}^{N} \nabla \tilde{V}_{i} f_{i}(x_{i}, u_{i}, v_{i}) < \sum_{i=1}^{n} r_{i}(x_{i}, u_{i}, v_{i}), \quad (8)$$

where the inequality follows from the dissipativity condition (5). Assuming the specific form for  $r_i$  in (6), we have

$$\sum_{i=1}^{n} r_i(x_i, u_i, v_i) = \sum_{i=1}^{N} \begin{bmatrix} u_i \\ y_i \\ v_i \\ y_i \end{bmatrix}^{\top} \begin{bmatrix} R_i & 0 \\ 0 & R_i \end{bmatrix} \begin{bmatrix} u_i \\ y_i \\ v_i \\ y_i \end{bmatrix}$$

$$= y^{\top} \begin{bmatrix} M \\ I \\ K \\ I \end{bmatrix}^{\top} \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \begin{bmatrix} M \\ I \\ K \\ I \end{bmatrix} y, \qquad (9)$$

where the last equality follows from the definition of u and v with R as specified in the proposition statement.

Now, if (7) is satisfied, using (8)-(9), it immediately implies that  $\dot{V}(x) < 0$ . The positivity of V also directly follows from that of  $\tilde{V}_i$ s. Thus, V satisfies the condition to be a valid Lyapunov function for the networked system.

**Remark 1.** Proposition 1 will enable us to reduce the high dimensional problem of finding a Lyapunov function for the

networked system to n smaller problems of finding the storage functions of the subsystems. For example, assuming  $d_i^x = d$  for all i, instead of an nd dimensional problem of finding the Lyapunov function for the networked system, we only need to solve n different d dimensional problems of finding the storage functions of the subsystems. Since the computational complexity typically grows exponentially in the dimension of the system, this will reduce the exponential dependence on n to a linear dependence on n. This reduction in complexity is especially significant for large-scale systems formed by the interconnection of a large number of small subsystems.

# B. Learning Neural Storage Functions and Secondary Controllers for Subsystems

Neural networks can efficiently learn/approximate a broad class of nonlinear functions using data, leading to their success in many engineering applications. Recent works have exploited this property of neural networks to learn Lyapunov functions for nonlinear systems [6], [11], [12]. We follow a similar approach for learning the storage function of individual subsystems. In addition to learning the storage function, we will also learn the secondary controller  $K_i$ , in order to maximize the stability region of the networked system. We achieve these goals by designing a loss function for the neural network learning that penalizes any violation of the dissipativity condition given in (5), and finding the parameter which minimizes this loss.

For each susbsyetm i, we create the training datasets  $\mathcal{D}_i^x = \{x_i^1, \dots, x_i^N\}$  and  $\mathcal{D}_i^u = \{u_i^1, \dots, u_i^N\}$  by drawing N samples from their domains according to some distribution. We represent the storage function  $\tilde{V}_i$  as  $\tilde{V}_{\theta_i}$  where  $\theta_i$  is the parameter of the corresponding neural network. We then define the empirical loss function for subsystem i as

$$L(\theta_{i}, K_{i}) = \tilde{V}_{\theta_{i}}^{2}(0) + \frac{1}{N} \sum_{j=1}^{N} \left[ \max(-\tilde{V}_{\theta_{i}}(x_{i}^{j}), 0) + \max(0, \nabla \tilde{V}_{\theta_{i}}(x_{i}^{j})^{\top} f_{i}(x_{i}^{j}, u_{i}^{j}, K_{i}y_{i}^{j}) - r_{i}(u_{i}^{j}, K_{i}y_{i}^{j}, y_{i}^{j}) \right],$$
(10)

where the secondary control input  $v_i^j$  is specified as  $K_i y_i^j$ . We will train a neural network for each subsystem using the above loss function until convergence.

A neural network based learning algorithm can give only an *empirical guarantee* with respect to its loss function and training data. So, the parameters of the neural network obtained by minimizing the above empirical loss function may not *provably* yield a storage function that satisfies the condition for dissipativity (5). We overcome this issue as described below.

Satisfiability Modulo Theories (SMT) solvers are widely used to verify correctness of symbolic arithmetic expressions in a specified region  $\mathcal B$  in the space of real numbers [25]. In particular, given a symbolic expression (also known as logic formula)  $\Phi(x)$ , an SMT solver uses a falsification technique to identify *counterexamples*  $x \in \mathcal B$  where  $\Phi(x)$  is violated. When no such counterexamples exists,  $\Phi(x)$  is said to be provably valid in  $\mathcal B$ . Recent works have implemented SMT solvers to validate Lyapunov functions [6], [12]. We follow a similar

procedure to validate the storage functions learned by our neural network. Specifically, by using an SMT solver at every neural network training step, we validate  $\tilde{V}_{\theta_i}$  by checking the following first order logic formula over real numbers:

$$\Phi(x_i, u_i) = \left(\varepsilon_{u_i} \ge ||x_i||^2 \ge \varepsilon_{l_i}\right) \land \left(\varepsilon_{u_i} \ge ||u_i||^2 \ge \varepsilon_{l_i}\right) \land \left(\tilde{V}_{\theta_i}(x_i) \le 0 \lor \nabla V_{\theta_i}(x_i)^\top f_i(x_i, u_i, K_i y_i) - r_i(u_i, K_i y_i, y_i)\right) \ge 0\right),$$
(11)

where  $\epsilon_{l_i}$  is a small lower-bound to avoid arithmetic underflow, and  $\epsilon_{u_i}$  is an upper-bound that defines the verification region  $\mathcal{B}_i$ . We use a specific SMT solver called dReal [26] to verify the above logic formula.

Under the conditions as described in [12], [26], [27], an SMT solver always returns a counterexample if there exists any. We add these counterexamples to our training dataset  $\mathcal{D}_i$ , and retrain the neural network to improve the validity of  $\tilde{V}_{\theta_i}$  over  $\mathcal{B}_i$ . When no counterexample is returned, we complete the neural network training, and declare  $\tilde{V}_{\theta_i}$  as a valid storage function. Performing this SMT verification based neural network training for all subsystems, we get  $(\tilde{V}_{\theta_i}, K_i)$  which can provably certify the dissipativity of the subsystem i with respect to the supply rate function  $r_i$ , for all  $i \in \{1, \ldots, n\}$ .

## C. Distributed Learning of Neural Lyapunov Function for the Networked System

After learning the neural storage functions for all subsystems, we can form a candidate neural Lyapunov function Vfor the networked system as,  $V(x) = \sum_{i} \tilde{V}_{\theta_i}(x_i)$ . However, the fact that  $V_{\theta_i}$  is a valid storage function for subsystem i with respect to the supply rate function  $r_i$ , for all i, does not immediately imply that V is a valid Lyapunov function for the networked system. The linear matrix inequality (LMI) (7) involving M, K and  $R_i$ s (which represents  $r_i$ s) in Proposition 1 gives a sufficient condition for such a V to be valid Lyapunov function for the networked system. Therefore, if (7) is not satisfied, we change  $R_i$ s (equivalently  $r_i$ s) and relearn the storage functions for the subsystems with respect to these new supply rate functions. We repeat this procedure until convergence. We employ the alternating direction method of multipliers (ADMM) approach for the efficient implementation and guaranteed convergence of this proposed procedure [9], as described below.

To bring the condition (7) into the canonical optimization form, we will define the indicator function  $\mathbb{I}_{global}$  as

$$\mathbb{I}_{\text{global}}(R_1, \dots, R_n) := \begin{cases} 0 & \text{if (7) holds true} \\ \infty & \text{otherwise.} \end{cases}$$
 (12)

Similarly, to bring the dissipativity condition (5) of each subsystem into the optimization form, we define n indicator functions  $\mathbb{I}_{Local,i}$  as

$$\mathbb{I}_{\text{local},i}(R_i, \tilde{V}_{\theta_i}) := \begin{cases} 0 & \text{if (5) holds true} \\ \infty & \text{otherwise.} \end{cases}$$
 (13)

The problem of finding a Lyapunov function for the networked system can now be written as the following optimization problem:

$$\min_{\substack{(R_i, \tilde{V}_{\theta_i}, Z_i)_{i=1}^n \\ \text{s.t.}}} \quad \sum_{i=1}^n \mathbb{I}_{\text{local},i}\left(R_i, \tilde{V}_{\theta_i}\right) + \mathbb{I}_{\text{global}}\left(Z_1, \dots, Z_n\right)$$

We construct the optimization problem in above way because through the auxiliary variables  $Z_i$ s, this network-level optimization problem can be divided into n+1 smaller subproblems and can be solved in a distributed manner. The iterative steps of the algorithm are as follows:

$$R_i^{k+1} = \underset{R_i \text{ s.t. (5)}}{\arg\min} \|R_i - Z_i^k + S_i^k\|_F^2,$$
 (15a)

$$(Z_i^{k+1})_{i=1}^n = \underset{(Z_i)_{i=1}^n \text{ s.t. (5)}}{\arg\min} \sum_{i=1}^n \|R_i^{k+1} - Z_i + S_i^k\|_F^2, \quad (15b)_i^k$$

$$S_i^{k+1} = R_i^{k+1} - Z_i^{k+1} + S_i^k, \quad (15c)_i^k$$

$$S_i^{k+1} = R_i^{k+1} - Z_i^{k+1} + S_i^k, (15c)$$

where  $(S_i)_{i=1}^n$  are matrices to have consensus between  $(R_i)_{i=1}^n$ and  $(Z_i)_{i=1}^n$ , and  $||.||_F$  is the Frobenius norm.

The first step (15a) finds a new supply rate  $R_i^{k+1}$  close to the auxiliary variable  $Z_i^k$  given the consensus variable  $S_i^k$ . We then obtain the storage function  $\tilde{V}_{\theta_i}$  and secondary controller  $K_i$  with respect to  $R_i^{k+1}$  through the neural network training described in Section III-B. We emphasize that this step can be implemented in parallel for the subsystems. Since parallel neural network training can be performed at scale using the modern GPU architecture, this approach offer significant reduction in training time. The second step (15b) is to impose the Lyapunov condition for the networked system. It finds (auxiliary) supply rate functions  $Z_i^{k+1}$  close to  $R_i^{k+1}$  given the consensus variables  $S_i^k$  such that they satisfy the sufficient condition (7) to get a Lyapunov condition for the networked system. While this update step is centralized, we note that this is a simple convex optimization problem and does not involve any neural network training. Finally, step (15c) updates the consensus variables and the iteration continues until convergence.

We summarize the stages in the above subsections as a concise algorithm in Algorithm 1.

#### IV. CASE STUDIES

In this section, we demonstrate the performance of our algorithm by evaluating it on three test cases: (i) Three microgrids network with conventional angle droop control, (ii) Four microgrids network with quadratic voltage droop control, and (iii) IEEE 123-node test feeder network with conventional voltage and angle droop controls. We compare the performance of our algorithm with the following approaches for finding the Lyapunov functions.

• Quadratic control Lyapunov (QCL) function, which computes the Lyapunov function by linearizing the dynamics around the origin [28]. Additionally, a linear quadratic regulator (LQR) based secondary controller is designed for the improved stability of the system.

Algorithm 1 Distributed Neural Lyapunov Function Learning Algorithm

- 1: **Input:** Initial supply rate matrices  $(R_i)_{i=1}^n$ , initial consensus metrics  $(S_i)_{i=1}^n$ , input-output coupling matrix M, tolerance level  $\eta$ .
- 2: repeat
- Learn neural storage functions  $(\tilde{V}_{\theta_i})_{i=1}^n$  and secondary controllers  $(K_i)_{i=1}^n$  according to the procedure described in Section III-B with respect to the current supply rate functions  $(R_i)_{i=1}^n$
- Update auxiliary variables  $(Z_i)_{i=1}^n$  according to (15b)
- Update consensus variables  $(S_i)_{i=1}^n$  according to (15c)
- Update supply rate functions  $(R_i)_{i=1}^n$  according to (15a)
- 7: **until**  $(\sum_{i} ||R_{i} Z_{i}|| > \eta)$
- 8: Output: Lyapunov function  $V = \sum_{i=1}^{n} \tilde{V}_{\theta_{i}}$  for the networked system
- Quadratic Lyapunov (QL) function, which finds the Lyapunov function in the same way as QCL but without a secondary controller.
- Centralized neural control Lyapunov (CNCL) function finds a Lyapunov function through neural network based centralized learning [12]. It also learns a secondary controller through neural network based learning.
- Centralized neural Lyapunov function (CNL) is the same as CNCL but without a secondary controller [6].

We evaluate two version of our algorithms:

- Distributed neural control (DNCL) Lyapunov function learns the the Lyapunov function as described in Algoirthm 1.
- Distributed neural Lyapunov (DNL) function is same as DNCL but without a secondary controller.

## A. Dynamics of the Network of Microgrids

Since our case studies focuses on the network of microgirds, we first give a brief description about their dynamics and the classical control schemes used in the literature.

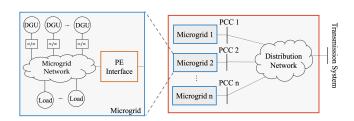


Fig. 1. A distribution system consisting of networked microgrids [6]. The internal structure of a single microgrid is shown in the left box.

Fig. 1 shows a network of microgrids where each microgrid comprises of several distributed generation units (DGUs), loads and power electronics (PE) interfaces [16]. We assume that the microgrids function in a grid-connected mode and they are networked with each other via their points of common coupling (PCCs) and distribution lines. The power injections to the PCCs are governed by the standard AC power flow equations as follows:

$$P_i = \sum_{k \neq i} E_i E_k Y_{ik} \cos(\delta_{ik} - \gamma_{ik}) + E_i^2 G_{ii}, \qquad (16a)$$

$$Q_i = \sum_{k \neq i} E_i E_k Y_{ik} \sin(\delta_{ik} - \gamma_{ik}) - E_i^2 B_{ii}, \tag{16b}$$

where  $\delta_{ik} = \delta_i - \delta_k$ ,  $E_i$  and  $\delta_i$  are voltage magnitude and phase angle of the *i*th microgrid;  $P_i$  and  $Q_i$  are active and reactive power injections at the *i*th PCC;  $G_{ii} + jB_{ii}$  is the *i*th diagonal element in the admittance matrix; and  $Y_{ik}$  and  $\gamma_{ik}$  are the magnitude and angle of the (i,k)th element in the admittance matrix respectively.

We consider two different control schemes that are widely used in the literature [29], [30].

1) Conventional droop control: In general, each microgrid employs a primary controller to stabilize the network and proportionally share the load among all the DGUs. This is achieved by setting up a control loop at a PE interface placed next to the microgrid. Droop controllers are the most extensively used control schemes, and they are based on active or reactive power decoupling [29]. The droop control based dynamics at *i*th microgrid is given by

$$J_{\delta_i} \Delta \dot{\delta_i} = -D_{\delta_i} \Delta \delta_i - \Delta P_i + K_i \Delta \delta_i, \tag{17a}$$

$$J_{E_i} \Delta \dot{E_i} = -D_{E_i} \Delta E_i - \Delta Q_i, \tag{17b}$$

where  $\Delta E_i = E_i - E_i^*$ ;  $\Delta \delta_i = \delta_i - \delta_i^*$ ;  $\Delta P_i = P_i - P_i^*$ ,  $\Delta Q_i = Q_i - Q_i^*$ ;  $E_i^*$ ,  $\delta_i^*$ ,  $P_i^*$ ,  $Q_i^*$  are the nominal set point values of the voltage magnitude, phase angle, active power and reactive power at the ith microgrid, respectively;  $J_{\delta_i}$  and  $J_{E_i}$  are tracking time constants;  $D_{\delta_i}$  and  $D_{E_i}$  are droop coefficients; and  $K_i$  is an output-feedback gain matrix. It is easy to verify that the above dynamics conform to the general form of dynamics given in (1)-(4) with  $x = [\Delta \delta_1, \Delta E_1, \ldots, \Delta \delta_n, \Delta E_n]^\top$ , h(x, u, v) = x,  $u = [\Delta P_1, \Delta Q_1, \ldots, \Delta P_n, \Delta Q_n]^\top$ ,  $v = [K_1 \Delta \delta_1, \ldots, K_n \Delta \delta_n]$ . The functions f can derived from (17a)-(17b) and g can derived from (16a)-(16b).

2) Quadratic droop control: Quadratic droop control [30] offers improvement both in theory and practice compared to the conventional droop control. The corresponding dynamics at *i*th microgrid is given by

$$J_{E_i} \Delta \dot{E_i} = -D_{E_i} \Delta E_i \left( \Delta E_i + E_i^* \right) - \Delta Q_i + K_i \Delta E_i, \tag{18}$$

such that (16b) holds true. Similar to the previous case, this dynamics also conform to general form given in (1)-(4), with  $x = [\Delta E_1, \dots, \Delta E_n]^{\top}$ , h(x, u) = x,  $u = [\Delta Q_1, \dots, \Delta Q_n]^{\top}$ ,  $v = [K_1 \Delta E_1, \dots, K_n \Delta E_n]$ . The functions f can derived from (17a)-(17b) and g can derived from (16a)-(16b).

In the remainder of this section, we consider specific test cases and validate the effectiveness of our algorithms for both conventional and quadratic droop control cases.

# B. Test Case I: Three microgrids network with conventional angle droop control

As standard in the power system literature, we assume that the dynamics of voltage magnitudes are predominantly slow

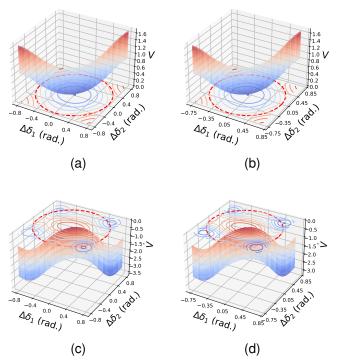


Fig. 2. Visualization of the distributed neural Lyapunov functions (V) and Lie derivatives  $(\dot{V})$  for the three microgrids system. (a) Projection of DNCL function on  $\Delta\delta_1$ - $\Delta\delta_2$  plane; (b) Projection of DNL function on  $\Delta\delta_1$ - $\Delta\delta_2$  plane; (c) Projection of Lie derivative of DNCL function on  $\Delta\delta_1$ - $\Delta\delta_2$  plane; (d) Projection of Lie derivative of DNL function on  $\Delta\delta_1$ - $\Delta\delta_2$  plane. The functions are positive and their Lie derivatives are negative, resulting in a valid local Lyapunov functions.

while the phase angles have fast dynamics, which is formalized by assuming that  $J_{E_i}\gg J_{\delta_i}$  in (17). This is referred to as *time scale separation* in power system literature [31]. With this assumption, we only focus on the behavior of the phase angle  $\Delta\delta_i$  and active power  $P_i$  for each subsystem  $i\in\{1,2,3\}$ . The distribution line parameters values are selected as given in [6], and the control parameters and set-point values are given in Table I. For neural network training, we use a two layer network, training sample size 2000, and learning rate 0.01. The parameters for SMT based verification are  $\epsilon_{l_i}=0.05$  and  $\epsilon_{u_i}=0.8$  for  $i\in\{1,2,3\}$ . The tolerance level for the algorithm is selected as  $\eta=10^{-6}$ . The algorithm converged with only three iterations of the outer loop.

Fig. 2(a) shows the DNCL function obtained using our algorithm, projected on the  $\Delta\delta_1$ - $\Delta\delta_2$  plane. Similarly, Fig. 2(b) shows DNL function obtained using our algorithm. Fig. 2(c) and Fig. 2(d) shows the Lie derivative  $(\dot{V})$  of the DNCL

TABLE I
CONTROL PARAMETERS AND REFERENCE SETPOINTS OF THE THREE
MICROGRIDS NETWORK [6]

	MG 1	MG 2	MG 3
$J_{\delta_i}$	1.2	1.0	0.8
$D_{\delta_i}$	1.2	1.2	1.2
$\delta_i^*$ (deg.)	0	55.67	-45.37
$E_i$ (p.u.)	1	1.05	0.95
$P_i^*$ (p.u.)	0.1706	1.4578	-0.0013

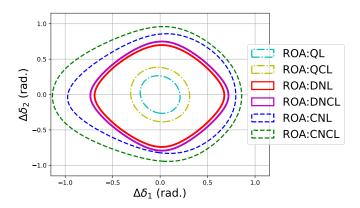


Fig. 3. Comparison of the estimated region of attraction (RoA) from different approaches for the three microgrid network. Projections on  $\Delta\delta_1$ - $\Delta\delta_2$  space (with  $\Delta\delta_3=0$ ) is shown here.

TABLE II
CONTROL PARAMETERS AND REFERENCE SETPOINTS OF THE FOUR
MICROGRIDS NETWORK [32]

	MG 1	MG 2	MG 3	MG 4
$J_{\delta_i}$	1	1	1	1
$D_{\delta_i}$	0.2	0.2	0.2	0.2
$\delta_i$ (deg.)	0	-6.30	-3.72	-10.027
$E_i^*$ (p.u.)	1	1	1	1
$Q_i^*$ (p.u.)	0.013	-0.013	0.017	-0.009

function and DNL function, respectively. We note that to compute Lie derivatives, we indeed considered the original nonlinear dynamics instead of the linear approximation for coupling. It is clear from these figures that the both DNCL and DNL functions are positive and their Lie derivatives are negative over the specified domain, and hence they satisfy the requirements to be a valid Lyapunov function as given in Definition 2. The plots of these Lyapunov functions and their Lie derivatives on rest of the projection planes follow the same pattern as in Fig. 2, and are omitted for brevity.

Fig. 3 shows the RoAs estimated using the Lyapunov functions obtained using our algorithm and other approaches mentioned in the beginning of this section. It is clear from the figue that, (i) the ROAs of DNCL and DNL are significantly larger than the one obtained through the classical approaches QCL and QL, and (ii) the RoA of DNCL is slightly larger than that of the DNL because the additional secondary controller in DNCL. We note that the effect of secondary controller on getting larger RoA will become more prominent as the system dimension increases, as we show in the next case studies. Thus, we show that our approach is less conservative in stability assessment as compared to traditional methods. Not surprisingly, the ROAs of DNCL and DNL are smaller than that of the centralized methods CNCL and CNL because of the performance limitations of distributed methods. However our distributed method offers significant reduction in the training time especially for large system, as shown in Table IV.

# C. Test Case II: Four microgrids network with quadratic voltage droop control

We next consider a line network of four microgrids with quadratic voltage droop control (18). All the set-point values and control parameters are shown in table II and the distribution line parameters are taken from [32]. For neural network training, we use a two layer network, training sample size 2000, and learning rate 0.01. The parameters for SMT based verification are  $\epsilon_{l_i}=0.06$  and  $\epsilon_{u_i}=0.7$ . The tolerance level for the algorithm is selected as  $\eta=10^{-6}$ . The algorithm converged with only four iterations of the outer loop.

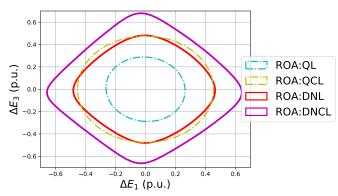


Fig. 4. Comparison of the estimated region of attraction (RoA) from different approaches for the four microgrids network. Projections on  $\Delta E_1$ - $\Delta E_3$  plane (with  $\Delta E_2 = \Delta E_4 = 0$ ) are shown.

Fig. 4 shows the RoAs estimated using the Lyapunov functions obtained using our algorithm and other approaches mentioned in the beginning of this section. We can make the following observations: (i) DNCL has larger ROA compared to QCL and QL, (ii) DNL's ROA is larger than QL but is comparable with QCL, and (iii) DNCL has larger ROA compared to DNL. ROAs of DNL and QCL are close to each other due to an LQR based secondary controller associated with QCL. Thus, our methods are better than the traditional methods for estimating the RoA. Because of the complexity in the dynamics due to quadratic droop control, the SMT verification associated with CNL and CNCL couldn't converge. Hence, we omit the RoAs of CNL and CNCL from Fig. 4. Finally, we conclude that for such systems that have complex non-linear nodal dynamics, DNL and DNCL are better suitable than centralized neural Lyapunov approaches.

# D. Test Case III: IEEE 123-node test feeder network with conventional voltage and angle droop control

Fig. 5 shows the IEEE 123-node test feeder system partitioned into five microgrids. We do not assume any time scale

TABLE III

CONTROL PARAMETERS AND REFERENCE SET-POINTS OF IEEE 123-NODE
TEST FEEDER NETWORK [21]

	MG 1	MG 2	MG 3	MG 4	MG 5
$J_{\delta_i}$	8	12	12	9	10
$J_{E_i}$	12.9	10.2	11.56	10.83	11.73
$D_{\delta_i}$	2.356	2.2	2.356	2.356	2.08
$D_{E_i}$	2.50	2.0	2.222	2.083	2.272
$\delta_i^*$ (deg.)	0	0.233	0.110	0.158	0.052
$E_i^*$ (p.u.)	1	1.003	1	1.003	0.999
$P_i^*$ (p.u.)	-0.13	0.57	-0.25	0.53	-0.72
$Q_i^*$ (p.u.)	0.88	-0.01	-0.1	0.08	-0.85

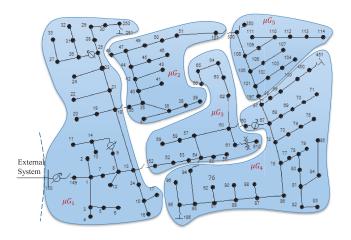


Fig. 5. IEEE 123-node test feeder converted into a five-microgrid system [21]

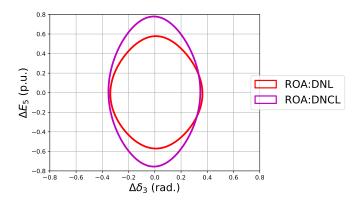


Fig. 6. Comparison of the estimated region of attraction (RoA) from DNCL and DNL for the IEEE 123-node test feeder network. Projections on  $\Delta\delta_3$ - $\Delta E_5$  plane (with  $\Delta\delta_1=\Delta\delta_2=\Delta\delta_4=\Delta\delta_5=\Delta E_1=\Delta E_2=\Delta E_3=\Delta E_4=0$ ) are shown.

separation for this case, and consider conventional voltage and angle droop controls as primary control schemes. The set-point values and control parameters are given in Table III, and the distribution line parameters are selected as given in [21]. The system dynamics is as specified in (17). We linearize the input-output coupling relationship around origin o and convert the dynamics to the standard form as given in (1)-(4). For neural network training, we use a two layer network, training sample size 4000, and learning rate 0.01. The parameters for SMT based verification are  $\epsilon_{l_i} = 0.06$  and  $\epsilon_{u_i} = 0.6$ . The tolerance level for the algorithm is selected as  $\eta = 10^{-6}$ . The algorithm converged with only four iterations of the outer loop.

Fig. 6 shows the RoAs estimated using our DNCL and DNL functions. Similar to the previous case studies, the RoA for DNCL is larger than that of DNL due to the secondary controller. We note that due to the large scale nature of the system, the RoAs estimated using the classical QCL and QL approaches are significantly smaller (and barely visible) compared to that of DNCL and DNL, and hence we have omitted them from the figure. Moreover, again due to the large scale nature of the system, the SMT-based verification for the CNCL and CNL failed to converge even after a

TABLE IV

COMPARISON OF THE TRAINING TIME FOR CENTRALIZED (CNCL, CNL) AND DECENTRALIZED (DNCL, DNL) LEARNING APPROACHES.

	DNL	DNCL	CNL	CNCL
	(sec.)	(sec.)	(sec.)	(sec.)
IEEE 123-node network	1679	1383.42		

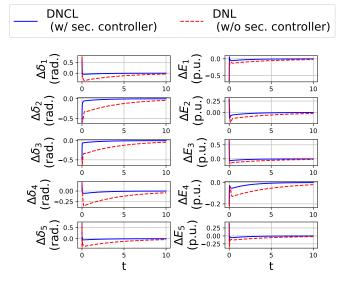


Fig. 7. Time domain simulation comparison for the 123-node feeder with initial state  $x(0) = [0.65, -0.65, -0.6, 0.3, -0.5, 0.65, 0.2, -0.2, 0.7, 0.4]^{\top}$ . The secondary controller (DNCL) stabilizes the system faster than DNL.

long time. Since CNCL and CNL functions learning did not converge, we were also not able to estimate any RoAs based on these functions, and they are omitted from the figure. We note that this further emphasizes the computational benefit of our *distributed* approach, which converges fast even with SMT-based verification. Table IV shows the training time comparsion.

## E. Advantages Due to the Secondary Controller

We observe that the learned secondary controller in the DNCL provides faster stabilization and robustness to perturbation, in addition to the larger RoA estimate, when compared to DNL where there is no secondary controller. The larger RoA of DNCL is clear from Fig. 3, 4, and 6. The faster convergence is demonstrated through a time domain simulation in Fig. 7. The increased robustness due to the secondary controller is demonstrated in Fig. 8. We add a small perturbation noise to the system state in the time domain simulation. From the figure, it is clear that the secondary controller is able to quickly suppress the disturbance.

#### V. CONCLUSION

This paper proposes a novel distributed learning based framework for assessing Lyapunov stability of a class of networked nonlinear systems, where each subsystem is dissipative. The objective of the proposed framework is to construct a Lyapunov function in a distributed manner and to

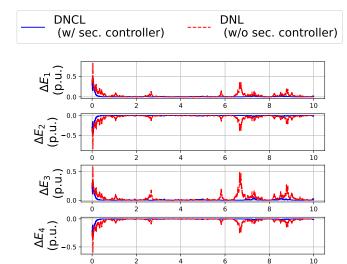


Fig. 8. Time domain simulation comparison for four microgrid network with initial state  $x(0) = [0.35, -0.35, 0.25, -0.25]^{\mathsf{T}}$ . The secondary controller (DNCL) is able to suppress the disturbance resulting in a smoother trajectory.

estimate the associated region of attraction for the networked system. We begin by leveraging a neural network function approximation to learn a storage function for each subsystem such that a local dissipativity property is satisfied by each subsystem. We next use a satisfiability modulo theories (SMT) solver based falsifier that verifies the local dissipativity of each subsystem by determining an absence of counterexamples that violate the local dissipativity property, as established by the neural network approximation. Finally, we verify network level stability by using an alternating direction method of multipliers (ADMM) approach to update the storage function of each subsystem in a distributed manner until a global stability condition for the network of dissipative subsystems is satisfied. We demonstrate the performance of the proposed algorithm and its advantages on three different case studies in microgrid networks. We note that this paper focuses on the synthesis of linear secondary controllers. Future work will focus on learning reinforcement learning based non-linear controllers that would further expand the RoA of the system and enhance its stability margins.

### REFERENCES

- P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.
- [2] D. Henrion and A. Garulli, Positive polynomials in control. Springer Science & Business Media, 2005, vol. 312.
- [3] Z. Jarvis-Wloszek, R. Feeley, W. Tan, K. Sun, and A. Packard, "Some controls applications of sum of squares programming," in *IEEE Confer*ence on Decision and Control (CDC), 2003, pp. 4676–4681.
- [4] U. Topcu, A. K. Packard, P. Seiler, and G. J. Balas, "Robust region-of-attraction estimation," *IEEE Transactions on Automatic Control*, vol. 55, no. 1, pp. 137–142, 2009.
- [5] U. Topcu, A. Packard, and P. Seiler, "Local stability analysis using simulations and sum-of-squares programming," *Automatica*, vol. 44, no. 10, pp. 2669–2675, 2008.
- [6] T. Huang, S. Gao, and L. Xie, "A neural lyapunov approach to transient stability assessment of power electronics-interfaced networked microgrids," *IEEE Transactions on Smart Grid*, vol. 13, no. 1, pp. 106–118, 2021.

- [7] M. Vidyasagar, "New passivity-type criteria for large-scale interconnected systems," *IEEE Transactions on Automatic Control*, vol. 24, no. 4, pp. 575–579, 1979.
- [8] P. Moylan and D. Hill, "Tests for stability and instability of interconnected systems," *IEEE Transactions on Automatic Control*, vol. 24, no. 4, pp. 574–575, 1979.
- [9] M. Arcak, C. Meissen, and A. Packard, Networks of dissipative systems: compositional certification of stability, performance, and safety. Springer, 2016.
- [10] E. Agarwal, S. Sivaranjani, V. Gupta, and P. J. Antsaklis, "Distributed synthesis of local controllers for networked systems with arbitrary interconnection topologies," *IEEE Transactions on Automatic Control*, vol. 66, no. 2, pp. 683–698, 2020.
- [11] G. Manek and J. Z. Kolter, "Learning stable deep dynamics models," Advances in neural Information Processing Systems, 2019.
- [12] Y.-C. Chang, N. Roohi, and S. Gao, "Neural lyapunov control," Advances in neural information processing systems, 2019.
- [13] T. Huang, S. Gao, X. Long, and L. Xie, "A neural lyapunov approach to transient stability assessment in interconnected microgrids." in *HICSS*, 2021, pp. 1–10.
- [14] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke et al., "Trends in microgrid control," *IEEE Transactions on smart grid*, vol. 5, no. 4, pp. 1905–1919, 2014.
- [15] F. Katiraei, R. Iravani, N. Hatziargyriou, and A. Dimeas, "Microgrids management," *IEEE power and energy magazine*, vol. 6, no. 3, pp. 54– 65, 2008.
- [16] R. Zamora and A. K. Srivastava, "Multi-layer architecture for voltage and frequency control in networked microgrids," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 2076–2085, 2016.
- [17] P. Shamsi and B. Fahimi, "Stability assessment of a dc distribution network in a hybrid micro-grid application," *IEEE Transactions on Smart Grid*, vol. 5, no. 5, pp. 2527–2534, 2014.
- [18] J. He, X. Wu, X. Wu, Y. Xu, and J. M. Guerrero, "Small-signal stability analysis and optimal parameters design of microgrid clusters," *IEEE Access*, vol. 7, pp. 36896–36909, 2019.
- [19] Y. Song, D. J. Hill, T. Liu, and Y. Zheng, "A distributed framework for stability evaluation and enhancement of inverter-based microgrids," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 3020–3034, 2017.
- [20] A. Jena, T. Huang, S. Sivaranjani, D. Kalathil, and L. Xie, "Distributed learning-based stability assessment for large scale networks of dissipative systems," in 2021 60th IEEE Conference on Decision and Control (CDC). IEEE, 2021, pp. 1509–1514.
- [21] S. Sivaranjani, E. Agarwal, V. Gupta, P. Antsaklis, and L. Xie, "Distributed mixed voltage angle and frequency droop control of microgrid interconnections with loss of distribution-pmu measurements," *IEEE Open Access Journal of Power and Energy*, vol. 8, pp. 45–56, 2020.
- [22] F. Okou, L.-A. Dessaint, and O. Akhrif, "Power systems stability enhancement using a wide-area signals based hierarchical controller," *IEEE Transactions on Power Systems*, vol. 20, no. 3, pp. 1465–1477, 2005
- [23] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice hall Upper Saddle River, NJ, 2002, vol. 3.
- [24] H.-D. Chiang, "Study of the existence of energy functions for power systems with losses," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 11, pp. 1423–1429, 1989.
- [25] C. Barrett and C. Tinelli, "Satisfiability modulo theories," in *Handbook of model checking*. Springer, 2018, pp. 305–343.
- [26] S. Gao, S. Kong, and E. M. Clarke, "dreal: An smt solver for nonlinear theories over the reals," in *International conference on automated* deduction. Springer, 2013, pp. 208–214.
- [27] S. Gao, J. Avigad, and E. M. Clarke, "δ-complete decision procedures for satisfiability over the reals," in *International Joint Conference on Automated Reasoning*. Springer, 2012, pp. 286–300.
- [28] H.-D. Chang, C.-C. Chu, and G. Cauley, "Direct stability analysis of electric power systems using energy functions: theory, applications, and perspective," *Proceedings of the IEEE*, vol. 83, no. 11, pp. 1497–1529, 1995.
- [29] J. M. Guerrero, J. C. Vasquez, J. Matas, L. G. De Vicuña, and M. Castilla, "Hierarchical control of droop-controlled ac and dc microgrids—a general approach toward standardization," *IEEE Transactions* on industrial electronics, vol. 58, no. 1, pp. 158–172, 2010.
- [30] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Voltage stabilization in microgrids via quadratic droop control," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1239–1253, 2016.

- [31] J. R. Winkelman, J. H. Chow, J. J. Allemong, and P. V. Kokotovic, "Multi-time-scale analysis of a power system," *Automatica*, vol. 16, no. 1, pp. 35–43, 1980.
- [32] A. Teixeira, K. Paridari, H. Sandberg, and K. H. Johansson, "Voltage control for interconnected microgrids under adversarial actions," in 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA). IEEE, 2015, pp. 1–8.

PLACE PHOTO HERE Amit Jena received the Integrated M.S. degree in mathematics from National Institute of Technology, Rourkela, India, M.S. degree in electrical engineering from Iowa State University, IA, USA. He is currently a Ph.D. student in electrical engineering at Texas A& M University, college station, TX, USA. His research interests includes reinforcement learning, meta learning, with application in power systems.

PLACE PHOTO HERE 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.

PLACE PHOTO HERE S. Sivaranjani received the B.E. degree from PES Institute of Technology, Bengaluru, India, M.S. degree from the Indian Institute of Science, Bengaluru, India, and Ph.D. degree from the University of Notre Dame, IN, USA, respectively, all in electrical engineering. She is currently an Assistant Professor in the School of Industrial Engineering at Purdue University, West Lafayette, IN. Previously, she was Postdoctoral Researcher with the Department of Electrical and Computer Engineering, Texas A& M University, College Station, TX, USA. Her research

interests include data-driven and distributed control for large-scale networked systems, with applications to energy systems and transportation networks.

PLACE PHOTO HERE Dileep Kalathil (Senior Member, IEEE) received his Ph.D. degree from the University of Southern California (USC) in 2014. From 2014 to 2017, he was a Postdoctoral Researcher with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Texas A&M University. His main research focus is on reinforcement learning theory and algorithms, with applications in energy systems, communication networks and

mobile robotics. He was a recipient of the Best Academic Performance from the EE Department, IIT Madras and the Best Ph.D. Dissertation Prize in the USC Department of Electrical Engineering, NSF CRII Award in 2019 and NSF CAREER award in 2021.

PLACE PHOTO HERE Le Xie (Fellow, 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.