Reinforcement Learning Based Distributed Control of Dissipative Networked Systems

Krishna Chaitanya Kosaraju, S. Sivaranjani, Wesley Suttle, Vijay Gupta, and Ji Liu

***Abstract*—We consider the problem of designing distributed controllers to stabilize a class of networked systems, where each subsystem is dissipative and designs a reinforcement learning based local controller to maximize an individual cumulative reward function. We develop an approach that enforces dissipativity conditions on these local controllers at each subsystem to guarantee stability of the entire networked system. The proposed approach is illustrated on a dc microgrid example, where the objective is to maintain voltage stability of the network using locally distributed controllers at each generation unit.**

***Index Terms*—Reinforcement learning, dissipativity theory, control barrier functions, distributed control.**

I. Introduction

DISTRIBUTED control of large-scale networked systems is a classical research topic, with practical applications in a variety of fields such as transportation, chemical reaction, and hydraulic networks, multibody mechanical systems, and microgrids [1]–[5]. The problem provides many challenges such as nonclassical information patterns, computational complexity due to the large state-space, scalability of control design methods, complex system dynamics that may be imperfectly known, and so on. Despite many important advances, the field continues to be a focus of intense research.

An interesting direction in recent times has been the utilization of reinforcement learning (RL) for distributed and multiagent control. RL is especially powerful for the control of systems where the dynamics and/or the environment are unknown [6]. In a typical RL-based design, the aim is to learn a controller that maximizes its cumulative reward while exploring the unknown environment. A wide variety of model-based and model-free algorithms is now available (see, e.g., [7] for a survey). While initially developed for single agent settings, the scope of RL-based techniques has also been expanded to multiagent networked systems (see [8]–[10] for surveys). Further, while the typical focus of RL-based techniques for controller design has been through simulations and demonstrations, a growing line of research now considers obtaining guarantees about concerns traditional to control theory, e.g., stability, safety, and robustness, through controllers obtained using RL [11], etc.

In this article, we consider the problem of guaranteeing stability when RL is used for distributed control of networked dynamical systems. Specifically, consider a large-scale system consisting of many subsystems that are coupled through their inputs and outputs, such as a network of microgrids. Each subsystem designs a local controller based on information about the subsystem state, inputs, and outputs. In particular, we assume that the controller is implemented using an RL algorithm since the dynamics of the subsystems may be unknown. Of note, however, different controllers may potentially use different RL algorithms. How do we design the controllers that guarantee that the entire system is still stable? There are at least two challenges here. First, we would like the control strategy to be distributed. While there exists wide literature on RL techniques for multiagent systems, distributed control strategies using RL that provide guarantees like stability, safety, and robustness [12] are still scant. Works that consider the problem of guaranteeing stability and robustness with RL controllers have largely been limited to contexts such as model-based RL and LQR designs for single-agent systems [13]–[16]. Second, most available literature on multiagent RL considers the case when all subsystems implement the same RL algorithm and further share information such as a global state or rewards with other subsystems. Development of RL-based controllers at the subsystems that ensure stability and robustness for the entire networked system, especially when different agents may not use the same RL algorithm, largely remains an open problem.

As a first step toward addressing this problem, we focus on a class of networked systems where each subsystem is dissipative [17] in open loop. Dissipativity is an input–output concept that can be used to guarantee a broad range of useful properties such as $\mathcal{L}_2$ stability, robustness with respect to disturbances, and stability under time-delays [18]–[20] and has been widely used in traditional control theory for distributed controller

Manuscript received May 11, 2021; revised August 22, 2021, August 22, 2021, and September 23, 2021; accepted October 4, 2021. Date of publication November 4, 2021; date of current version June 20, 2022. This work was supported in part by the US Army Research Laboratory (ARL) Cooperative Agreements under Grant W911NF-21-2-0098 and Grant W911NF1910483, in part by the National Science Foundation (NSF) under Grant CCF-1934904 and Grant ECCS-2020246, in part by the Defense Advanced Research Projects Agency (DARPA) under Grant FA8750-20-2-0502, and in part by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-21-1-0231. Recommended by Associate Editor L. Pavel. *(Corresponding author: Vijay Gupta.)*

Krishna Chaitanya Kosaraju and Vijay Gupta are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: Kosaraju,k.c.kosaraju@nd.edu; vgupta2@nd.edu).

S. Sivaranjani is with the Department of Electrical, and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: sivaranjani@tamu.edu).

Wesley Suttle and Ji Liu are with the Department of Applied Mathematics, and Statistics, and the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: wesley.suttle@stonybrook.edu; ji.liu@stonybrook.edu).

Digital Object Identifier 10.1109/TCNS.2021.3124896

synthesis [21]–[29]. In the context of RL, dissipativity has been used to enhance the convergence/performance of various learning schemes [30] and has been enforced as a system property for specific systems like Port-Hamiltonian systems [31], [32]. However, there has been limited literature on enforcing it using model-free RL techniques or on exploring its potential to permit distributed controller design that guarantees properties such as stability at the system level (a notable exception for L_2-stability of cascade interconnections of dynamically coupled linear systems is in [33]). The challenge in our formulation is that an RL controller aiming to optimize the local performance metric at a subsystem can easily disrupt the dissipativity of the subsystem with respect to the variables that it exchanges with the other subsystems.

In this article, we develop an RL-based distributed control design approach that exploits the dissipativity property of individual subsystems to guarantee stability of the entire networked system. Our proposed approach can be summarized as follows. We first use a control barrier function (CBF) to characterize the set of controllers that enforce a dissipativity condition at each subsystem (Propositions 2 and 3). We impose a minimal energy perturbation on the control input learned by the RL algorithm to project it to an input in this set (Theorem 3). Together, these results guarantee the stability of the entire networked system even when the subsystems utilize potentially heterogeneous RL algorithms to design their local controllers (Theorem 4).

Our approach of utilizing a CBF to impose the constraint that the controller designed for each subsystem using RL preserves the dissipativity of the subsystem in the closed loop parallels the use of CBFs to enforce safety in RL algorithms [11]. CBFs guarantee the existence of control inputs under which a superlevel set of a function (typically representing specifications like safety) is forward invariant under a given dynamics [34]–[36]. However, their use to impose input–output properties, such as dissipativity, is less studied. Here, we utilize CBFs to characterize the set of dissipativity ensuring controllers, and then learn a dissipativity ensuring controller for each subsystem from this set.

The main contribution of this work is a distributed approach to ensure stability of a networked system with dissipative subsystems when the individual subsystems utilize RL to design their own controllers. Beyond the specific stabilization problem that we focus on, integrating dissipativity (and other input–output) specifications into RL-based control is useful since it allows a wide landscape of tools from classical dissipativity theory to be integrated into RL-based control design. The proposed algorithm guarantees stability irrespective of the choice of the RL algorithm used at each subsystem. In particular, the results also hold for heterogeneous RL algorithms being used at each subsystem. We also note that as opposed to most existing literature on multiagent RL, the proposed approach requires only the output from neighboring subsystems to learn the control policy at each subsystem. In other words, to guarantee stability, no information about the states, rewards, or policies of other subsystems is required.

The rest of the article is organized as follows. In Section II, we present the model of the networked system, state the necessary assumptions, and provide the problem formulation. In Section III-A, we utilize CBFs to characterize the set of controllers that guarantees dissipativity of each subsystem. In Section III-B, we present an RL algorithm to compute a control input that preserves the dissipativity of each subsystem, and show that it stabilizes the networked system. In Section IV, we numerically illustrate our approach on a dc microgrid application. Finally, in Section V, we provide some directions for future work. Proofs of all the results in the article, and the definitions of dissipativity, are provided in the Appendix.

Notation: $\mathbb{R}^m$ denotes the space of m-dimensional real vectors, $\mathbb{R}$ denotes the space of real numbers, and $\mathbb{R}_+$ denotes the set of all positive real numbers. $\otimes$ denotes the Kronecker product. $z^\top$ denotes the transpose of a vector or a matrix z and $\|z\|_2$ (or simply $\|z\|$) denotes its 2-norm. For a symmetric matrix M and a vector z of compatible dimensions, $\|z\|_M^2$ is defined to be equal to $z^\top Mz$. Given square matrices M_1, M_2, ..., M_n, define the matrix $\text{diag}(M_i)$ as the block diagonal matrix whose main-diagonal blocks are matrices M_1, M_2, ..., M_n, and all OFF-diagonal blocks are zero matrices. For a symmetric matrix M, $\lambda_{\min}(M)$ denotes its smallest eigenvalue. I denotes the identity matrix with dimensions clear from the context. A directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a finite set of nodes (or vertices) $\mathcal{V}$ and a set of directed edges (or arcs) $\mathcal{E}$, together with a mapping from $\mathcal{E}$ to the set of pairs of $\mathcal{V}$. By convention, we disregard self-loops. Thus, to any arc $e \in \mathcal{E}$, there corresponds an ordered pair $(u, v) \in \mathcal{V} \times \mathcal{V}$, with $u \neq v$, representing the head vertex u and the tail vertex v. Given this, a shorthand notation is to simply say $(u, v) \in \mathcal{E}$. A graph is undirected if whenever $(u, v) \in \mathcal{E}$ then $(v, u) \in \mathcal{E}$. The in-neighbor set $\mathcal{N}_i$ of node i is the set of all vertices j such that $(j, i) \in \mathcal{E}$. Let $\mathcal{D} \subset \mathbb{R}^n$. A function $f : \mathcal{D} \to \mathbb{R}^n$ is Lipschitz if there exists a constant L satisfying $\|f(b) - f(a)\|_2 \leq L\|b - a\|_2$ for all a, $b \in \mathcal{D}$, and class C^1 if it is continuously differentiable. We denote a value obtained by sampling the probability distribution function $f_X(x)$ for a random variable X as $y \sim f_X(x)$. When the random variable is clear from the context, we denote the distribution function simply by $f(x)$.

II. Problem Formulation

We adapt the general framework described in [23] and is shown in Fig. 1.

Node dynamics: Consider a networked system described by a strongly connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $i \in V$ is a subsystem Σ_n^i, given by

$$\Sigma_{\mathrm{n}}^i : \begin{cases} x_{t+1}^i = f^i(x_t^i, u_t^i, \nu_t^i) \\ y_{u,t}^i = g^i(x_t^i, u_t^i) \\ y_{\nu,t}^i = h^i(x_t^i, \nu_t^i) \end{cases} \tag{1}$$

where at time t, $x_t^i \in \mathbb{R}^{n_i}$ denotes the state of the i-th subsystem, $u_t^i \in \mathbb{R}^{m_i}$ denotes the control input applied by the subsystem controller that needs to be designed, and $\nu_t^i \in \mathbb{R}^{p_i}$ is the input to the i-th subsystem that depends on the output of the other subsystems in the in-neighbor set of node i. The subsystem has two outputs: $y_{u,t}^i \in \mathbb{R}^{\overline{o}_i}$ which is the output that is used to design the control input u_t^i, and $y_{\nu,t}^i \in \mathbb{R}^{\hat{o}_i}$ which is the output that is

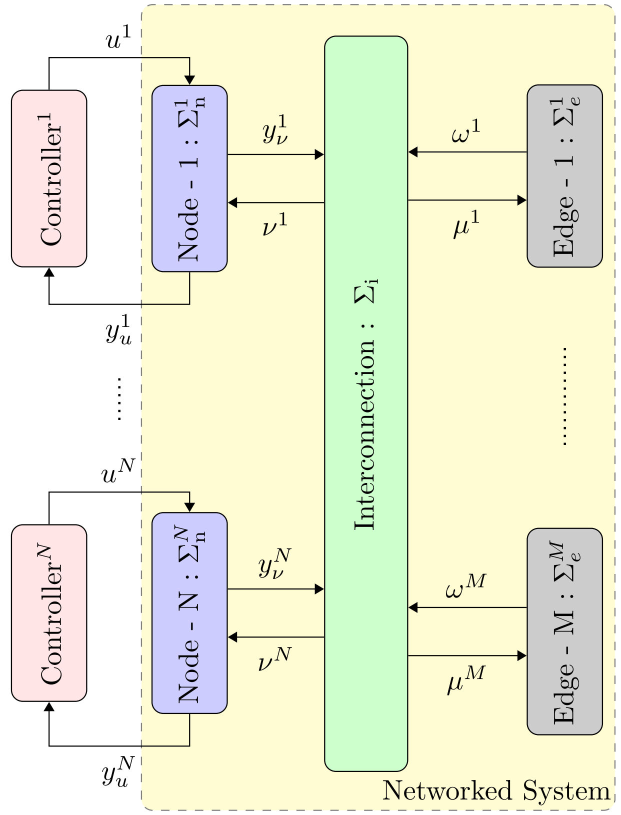


Fig. 1. Schematic of the system configuration.

used to compute the inputs ν_t^j for other subsystems j for whom i is an in-neighbor. We will define the exact relation between ν_t^i and $y_t^j, j \in \mathcal{N}_i$, later. Given that each subsystem corresponds to a unique node in the graph, we use the terms subsystem dynamics and node dynamics interchangeably. We assume that the state transition function f^i and the output functions g^i, h^i are of Class C^1. Without loss of generality, we assume that $(x^i = 0, u^i = 0, \nu^i = 0)$ is an equilibrium point of the subsystem Σ_{n}^i.

For future reference, define $x^\top \triangleq [x^{1\top}, \dots, x^{N\top}] \in \mathbb{R}^n$, $u^\top \triangleq [u^{1\top}, \dots, u^{N\top}] \in \mathbb{R}^m$, $y_u^\top \triangleq [y_u^{1\top}, \dots, y_u^{N\top}] \in \mathbb{R}^{\bar{o}}$, $y_\nu^\top \triangleq [y_\nu^{1\top}, \dots, y_\nu^{N\top}] \in \mathbb{R}^{\hat{o}}$, $y^i \triangleq [y_u^{i\top}, y_\nu^{i\top}] \in \mathbb{R}^o$, $y^\top \triangleq [y^{1\top}, \dots, y^{N\top}] \in \mathbb{R}^o$, and $\nu^\top \triangleq [\nu^{1\top}, \dots, \nu^{N\top}] \in \mathbb{R}^p$.

As stated earlier, definitions of dissipativity are provided in Appendix A for the sake of completeness. We make the following assumption throughout the article.

Assumption 1 (Dissipative node dynamics): Each subsystem Σ_n^i with dynamics defined in (1) is dissipative, in the set $\mathcal{S}_n^i$, with respect to the supply-function

$$w_n^i(u^i, \nu^i, y_u^i, y_\nu^i) = \underbrace{u^{i\top} S_u^{i\top} y_u^i - \|u^i\|_{R_u^i}^2 - \|y_u^i\|_{Q_u^i}^2}_{\triangleq w_u^i(u^i, y_u^i)}$$
$$+ \underbrace{\nu^{i\top} S_\nu^{i\top} y_\nu^i - \|\nu^i\|_{R_\nu^i}^2 - \|y_\nu^i\|_{Q_\nu^i}^2}_{\triangleq w_\nu^i(\nu^i, y_\nu^i)}, \quad (2)$$

where S_u^i, $R_u^i = (R_u^i)^\top$, $Q_u^i = (Q_u^i)^\top$, S_ν^i, $R_\nu^i = (R_\nu^i)^\top$, and $Q_\nu^i = (Q_\nu^i)^\top$ are matrices of appropriate dimensions.

For future reference, define $S_u \triangleq \mathrm{diag}(S_u^i)$, $R_u \triangleq \mathrm{diag}(R_u^i)$, $Q_u \triangleq \mathrm{diag}(Q_u^i)$, $S_\nu \triangleq \mathrm{diag}(S_\nu^i)$, $R_\nu \triangleq \mathrm{diag}(R_\nu^i)$, and $Q_\nu \triangleq \mathrm{diag}(Q_\nu^i)$. Further, denote $\epsilon_\nu = \lambda_{\min}(Q_\nu)$, $\delta_\nu = \lambda_{\min}(R_\nu)$, $\epsilon_u = \lambda_{\min}(Q_u)$, $\delta_u = \lambda_{\min}(R_u)$.

Remark 1: Even though Assumption 1 states that the subsystem (1) is dissipative, it is an assumption in the "open loop." Note that the design of the controller that determines the input u^i has not yet been specified. The dissipativity property required for the system stability concerns the inputs μ^i and the outputs y_μ^i and this may easily be disrupted by the additional dynamics introduced through the design of the controller $u^i = \zeta^i(x^i)$. For a simple illustration of this fact, if Assumption 1 holds, then we have that the relation

$$\sum_{t=t_0}^{t-1} \sum_{i=1}^{N} \left(w_u^i(u_t^i, y_{u_t}^i) + w_\nu^i(\nu^i, y_\nu^i) \right) \geq 0 \quad (3)$$

holds for all $0 \leq t_0 \leq t$. Consider the subsystem (1) in closed-loop with a Lipschitz controller $u^i = \zeta^i(x^i) \in \mathbb{R}^{m_i}$. Then, we notice that

$$\sum_{t=t_0}^{t-1} \sum_{i=1}^{N} w_\nu^i(\nu_t^i, y_{\nu,t}^i) \geq -\sum_{t=t_0}^{t-1} \sum_{i=1}^{N} w_u^i(\zeta^i(x_t^i), y_{u,t}^i) \quad (4)$$

which implies that unless the controller has been designed to ensure that $w_u^i(\zeta^i(x^i), y_u^i) \leq 0$, dissipativity of the subsystem in the closed loop with the controller may not be preserved.

Edge dynamics: While the simplest form of coupling among the subsystems would be to equate the inputs ν_t^i for the subsystem i with the output y_t^j of subsystem j if $(j,i) \in \mathcal{E}$, inspired by [23], we consider a more general model that allows the edges in the graph $\mathcal{G}$ to be described a dynamic system as well. Specifically for edge $k \in \mathcal{E}$, the dynamics are given by

$$\Sigma_{\mathrm{e}}^k : \begin{cases} z_{t+1}^k = g^k(z_t^k, \mu_t^k) \\ \omega_t^k = j^k(z_t^k, \mu_t^k) \end{cases} \quad (5)$$

where $z_t^k \in \mathbb{R}^{q_i}$ denotes the edge subsystem state at time t, $\mu_t^k \in \mathbb{R}^{r_i}$ denotes the input at time t, and $\omega_t^k \in \mathbb{R}^{s_i}$ denotes the output at time t. We assume that the state transition function g^k and the output function j^k are of Class C^1. Once again, without loss of generality we assume that $(z^k = 0, \mu^k = 0)$ is an equilibrium point of the subsystem Σ_{e}^k. For future reference, define $z^\top \triangleq [z^{1\top}, \dots, z^{M\top}] \in \mathbb{R}^q$, $\omega^\top \triangleq [\omega^{1\top}, \dots, \omega^{M\top}] \in \mathbb{R}^s$, and $\mu^\top \triangleq [\mu^{1\top}, \dots, \mu^{M\top}] \in \mathbb{R}^r$, where M denotes the cardinality of the set $\mathcal{E}$.

Assumption 2 (Dissipative edge dynamics): Each subsystem Σ_e^k with its dynamics defined in (5) is dissipative in the set $\mathcal{S}_e^k$ with supply-function

$$w_e^k(\mu^k, \omega^k) = \mu^{k\top} S_e^{k\top} \omega^k - \|\mu^k\|_{R_e^k}^2 - \|\omega^k\|_{Q_e^k}^2 \quad (6)$$

where S_e^k, $R_e^k = (R_e^k)^\top$, $Q_e^k = (Q_e^k)^\top$ are matrices of appropriate dimensions.

For future reference, define $S_e \triangleq \mathrm{diag}(S_e^k)$, $R_e \triangleq \mathrm{diag}(R_e^k)$, and $Q_e \triangleq \mathrm{diag}(Q_e^k)$. Further, define $\epsilon_e = \lambda_{\min}(Q_e)$ and $\delta_e = \lambda_{\min}(R_e)$.

Interconnection among subsystems: The entire networked system is defined through the interconnection of the subsystems

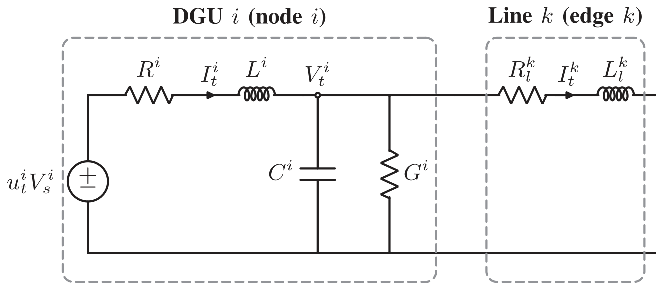


Fig. 2. Electrical scheme of DGU i and transmission line k as considered in Example 1.

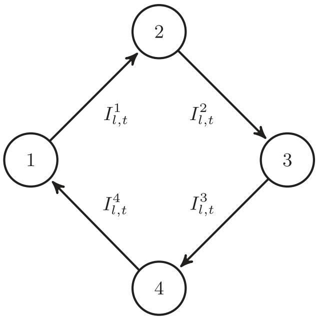


Fig. 3. Topology of network considered in Example 1.

defined by the nodes and edges by relating the inputs ν and outputs y_ν of the node subsystems with the inputs μ and outputs ω of the edge subsystems as specified below. Define $s^\top \triangleq [x^\top, z^\top]$ as the state variable of the overall network. Further, define

$$\begin{aligned} w_u(u, y_u) &\triangleq \left(u^\top S_u^\top y_u - \|u\|_{R_u}^2 - \|y_u\|_{Q_u}^2\right), \\ w_\nu(\nu, y_\nu) &\triangleq \left(\nu^\top S_\nu^\top y_\nu - \|\nu\|_{R_\nu}^2 - \|y_\nu\|_{Q_\nu}^2\right), \\ w_e(\mu, \omega) &\triangleq \left(\mu^\top S_e^\top \omega - \|\mu\|_{R_e}^2 - \|\omega\|_{Q_e}^2\right). \end{aligned} \tag{7}$$

Following [23], we model the interconnection among the subsystems through the equation

$$\Sigma_{\mathrm{i}} : \begin{bmatrix} \nu \\ \mu \end{bmatrix} = \begin{bmatrix} 0 & \mathcal{B} \\ -\mathcal{B}^\top & 0 \end{bmatrix} \begin{bmatrix} y_\nu \\ \omega \end{bmatrix} \tag{8}$$

for a suitably defined matrix $\mathcal{B}$. Further, we make the following assumption.

Assumption 3: Matrices S_ν and S_e in (7) satisfy

$$\mathcal{B}^\top S_\nu^\top - S_e \mathcal{B}^\top = 0. \tag{9}$$

An interpretation of (8) and Assumption 3 is that the edges of the system do not generate any energy. Although (8) appears intricate, most interconnected physical systems can be written in this form (see [23] for examples from various domains; an example of interconnected distributed generation units is discussed in detail below). Similarly, several relevant subclasses of dissipative systems including, but not limited to, $\mathcal{L}_2$ gain systems and passive systems satisfy Assumption 3; see [22] for other examples. For future reference, denote

$$\mathcal{B}_\delta(x) \triangleq \epsilon_e I + x\mathcal{B}^\top \mathcal{B} \tag{10}$$

$$\mathcal{B}_\epsilon(y) \triangleq yI + \delta_e \mathcal{B}\mathcal{B}^\top \tag{11}$$

where, as defined earlier, $\epsilon_e = \lambda_{\min}(Q_e)$ and $\delta_e = \lambda_{\min}(R_e)$.

Example 1: Consider the electrical schematic of a microgrid, containing four distributed generating units (DGUs) and interconnected through four transmission lines, as shown in Figs. 2 and 3. The DGUs correspond to the nodes and the transmission lines correspond to the edges of the graph describing this networked system. Let the DGUs and the transmission lines be numbered as shown in Fig. 3. Each DGU contains a dc–dc buck converter that is operating on a constant impedance load. The controller to be designed sets $u_t^i \in (0, 1)$ for the i-th DGU. Denote by I_t^k the current through the k-th transmission line at time t and by V_t^i the voltage across the i-th DGU at time t. Define the state of the subsystem at the i-th node (corresponding to the i-th DGU) by $x_t^i \triangleq [\, I_t^i \; V_t^i \,]^\top$. The dynamics of the DGU at node $i \in \mathcal{V} := \{1 \dots 4\}$, which forms the i-th subsystem, can be written as

$$\begin{aligned} I_{t+1}^i &= I_t^i - (T_s/L^i)(R^i I_t^i + V_t^i - u_t^i V_s) \\ V_{t+1}^i &= V_t^i + (T_s/C^i)(I_t^i - G^i V_t^i + \nu_t^i) \end{aligned} \tag{12}$$

where $T_s, L^i, C^i, R^i, G^i, V_s^i \in \mathbb{R}_{>0}$ are constants; $u_t^i \in (0,\ 1)$ is the local control input to be designed; and $\nu_t^i \in \mathbb{R}$ is the input to the i-th subsystem that depends on the output of the other subsystems in its in-neighbor set through the relations

$$\begin{bmatrix} \nu_t^1 \\ \nu_t^2 \\ \nu_t^3 \\ \nu_t^4 \end{bmatrix} = \begin{bmatrix} I_{l,t}^4 - I_{l,t}^1 \\ I_{l,t}^1 - I_{l,t}^2 \\ I_{l,t}^2 - I_{l,t}^3 \\ I_{l,t}^3 - I_{l,t}^4 \end{bmatrix} \tag{13}$$

where $I_{l,t}^k$ denotes the current through the edge k. We denote the outputs $y_{\nu,t}^i \triangleq V_t^i$.

The edges correspond to the transmission lines connected to each DGU. The dynamics of the transmission line at edge $k \in \mathcal{E} := \{1 \dots 4\}$ are given by

$$\begin{aligned} I_{l,t+1}^k &= I_{l,t}^k - (T_s/L_l^k)(R_l^k I_{l,t}^k + \mu_t^k) \\ \omega_t^k &= I_{l,t}^k \end{aligned} \tag{14}$$

where $L_l^k \; and \, R_l^k \in \mathbb{R}_{>0}$ are constants, $I_{l,t}^k \in \mathbb{R}$ denotes the state variable, and $\mu_t^k \in \mathbb{R}$ denotes the input from the nodes connected to the edge k defined as

$$\begin{bmatrix} \mu_t^1 \\ \mu_t^2 \\ \mu_t^3 \\ \mu_t^4 \end{bmatrix} = \begin{bmatrix} V_t^2 - V_t^1 \\ V_t^3 - V_t^2 \\ V_t^4 - V_t^3 \\ V_t^1 - V_t^4 \end{bmatrix}. \tag{15}$$

Define the incidence matrix $\mathcal{B} \in \mathbb{R}^{4\times 4}$ to model the network topology. Specifically, if the ends of each edge k are arbitrarily labeled with a $+$ and a $-$, then the entries of $\mathcal{B}$ are given by

$$\mathcal{B}_{ik} = \begin{cases} +1, & \text{if } i \text{ is the positive end of } k \\ -1, & \text{if } i \text{ is the negative end of } k \\ 0, & \text{otherwise.} \end{cases}$$

The interconnection between the nodes and edges can then be expressed as

$$\begin{bmatrix} \nu_t \\ \mu_t \end{bmatrix} = \begin{bmatrix} 0 & \mathcal{B} \\ -\mathcal{B}^\top & 0 \end{bmatrix} \begin{bmatrix} y_{\nu,t} \\ \omega_t \end{bmatrix} = \begin{bmatrix} \mathcal{B} I_{l,t} \\ \mathcal{B}^\top V_t \end{bmatrix}. \tag{16}$$

Controller design: We assume that each subsystem i wishes to design its controller to maximize the expected discounted cumulative reward

$$J^i = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t^i(x_t^i, u_t^i)\right] \tag{17}$$

where $\gamma \in (0, 1)$ is the discount factor, $r_t^i(x_t^i, u_t^i)$ is the per step reward function evaluated at time t, and the expectation is over any stochasticity that may arise due to the control policy itself. We assume that each agent utilizes an RL algorithm to design its controller. For a given control policy π^i, we define the value function V_π^i, and the state-action value function Q_π^i below

$$V_\pi^i(x^i) = \mathbb{E}_{\pi^i}\left[\sum_{t=0}^{\infty} \gamma^t r_t^i(x_t^i, u_t^i) \mid x_0^i = x^i\right], \tag{18}$$

$$Q_\pi^i(x^i, u^i) = \mathbb{E}_{\pi^i}\left[\sum_{t=0}^{\infty} \gamma^t r_t^i(x_t^i, u_t^i) \mid x_0^i = x^i, u_0^i = u^i\right], \tag{19}$$

$$A_\pi^i(x^i, u^i) = Q_\pi^i(x^i, u^i) - V_\pi^i(x^i). \tag{20}$$

Note that we do not assume that each subsystem utilizes the same RL algorithm. However, we assume that the RL algorithms converge.

Problem statement: Equations (1), (5), and (8) jointly define the networked system Σ under consideration, with state defined as $s_t^\top \triangleq [x_t^\top, z_t^\top]$. From Assumption 1, we know that the each subsystem i is dissipative with the supply-function $w_u^i(u^i, y_u^i) + w_\nu^i(\nu^i, y_\nu^i)$. However, since the subsystems use RL to design their local controllers, the closed loop subsystems may not remain dissipative (see Remark 1). Further, the control actions of all the subsystems may end up destabilizing the entire networked system. We are interested in the problem of how to design the RL algorithm at each subsystem to guarantee the stability of the networked system. Specifically, consider a networked system on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, described by (1), (5), and (8), and satisfying Assumptions 1, 2, and 3. Assume that the controller at each subsystem i is designed using an RL algorithm to maximize the discounted cumulative reward J^i in (17). How should the updates in the RL algorithms be done so that the control policies at convergence guarantee Lyapunov stability of the overall networked system?

III. Dissipativity Ensuring RL

In this section, we present the main results of the article through a new distributed RL algorithm that guarantees the stability of the entire networked system. The proposed approach is as follows.

1) *CBFs for dissipativity*: As stated in Remark 1, even though each subsystem i is dissipative with supply-function $w_u^i(u^i, y_u^i) + w_\nu^i(\nu^i, y_\nu^i)$, with the controller for the input u^i, the subsystem may no longer remain dissipative with the input-output pair $w_\nu^i(\nu^i, y_\nu^i)$. Our first step is to utilize CBFs to characterize the set of all controllers that ensure that the closed loop subsystem i is dissipative with respect to the input ν^i and output y^i (c.f. Fig. 1) with the supply-function

$$w_d^i(\nu^i, y^i) = \nu^{i\top} S_\nu^{i\top} y_\nu^i - \delta_d^i \|\nu^i\|_2^2 - \epsilon_d^i \|y^i\|_2^2 \tag{21}$$

where $\delta_d^i \in \mathbb{R}$ and $\epsilon_d^i \in \mathbb{R}$ are tuning parameters set by the designer.

2) *Projection-based RL algorithm for dissipativity*: In the second step, at each subsystem i, we consider the control input generated by an RL algorithm that seeks to maximize the discounted cumulative reward given by (17) and use a quadratic program (QP) to project this control input onto the set of control inputs that ensure that the closed loop subsystem remains dissipative with supply-function $w_d^i(\nu^i, y_\nu^i)$. Note that the RL algorithms used at different nodes can be different.

3) *Networked system stability*: We finally show that if each subsystem designs the controller to ensure that it is dissipative, the entire networked system is also stable.

We now develop these steps one by one. We will make the following assumption in the sequel.

Assumption 4: Denote $\delta_{\min} = \min(\delta_d^1, \ldots, \delta_d^N)$, and $\epsilon_{\min} = \min(\epsilon_d^1, \ldots, \epsilon_d^N)$. The conditions

$$\begin{aligned} \mathcal{B}_\delta(\delta_{\min}) &\geq 0, \\ \mathcal{B}_\epsilon(\epsilon_{\min}) &\geq 0, \end{aligned} \tag{22}$$

hold, where $\mathcal{B}_\delta$, and $\mathcal{B}_\epsilon$ have been defined in (10), and the inequality is in the positive semi-definite sense..

A. CBFs for Dissipativity

CBFs are now a popular tool for enforcing safety constraints in nonlinear control systems. The following definition follows the development in [37]–[39].

Definition 1 (Time-varying Zeroing CBFs): Consider a function $b : \mathbb{R}_+ \times \mathbb{R}^{n+q} \to \mathbb{R}$ that is continuously differentiable in both arguments. Define a closed set $\mathcal{C}$ as the super-level set of this function as follows:

$$\mathcal{C} \triangleq \left\{s \in \mathbb{R}^{n+q} \mid b(t, s) \geq 0\right\}. \tag{23}$$

The function $b(t, s_t)$ is a time-varying zeroing CBF, for the networked system Σ described by (1), (5), and (8) and with state s_t, if there exists an $\eta \in [0,\ 1]$ such that for all $s_t \in \mathcal{C}$, $t \in \mathbb{R}_+$

$$\sup_{u_t \in \mathbb{R}^m} \ [b(t+1, s_{t+1}) + (\eta - 1) b(t, s_t)] \geq 0. \tag{24}$$

CBFs can be used to derive sufficient conditions under which a super-level set of a function of the state of the networked system Σ is forward invariant. These conditions also characterize the set of control inputs achieving such forward invariance through the relation

$$\mathrm{B}(t,s_t) \triangleq \{u_t \in \mathbb{R}^m | b(t+1,s_{t+1}) + (\eta-1)b(t,s_t) \geq 0\}. \tag{25}$$

The following result, given for completeness for a discrete time setting such as ours, shows that the set $\mathcal{C}$ defined in (23) is forward invariant for every $u_t \in \mathrm{B}(t,s_t)$.

Proposition 1 (Discrete-time time-varying CBFs): Consider a time-varying zeroing CBF $b(t,s_t)$ and its super level set $\mathcal{C}$ defined in (23). Then any input $u_t \in \mathrm{B}(t,s_t)$, where $\mathrm{B}(t,s_t)$ is given in (25), will render the set $\mathcal{C}$ forward invariant.

Although dissipativity is a property defined by the input, and the output, we can utilize CBFs to characterize the set of controllers that ensures dissipativity in the closed loop of the subsystems, which in turn guarantee the stability of the overall networked system [40]. Following Proposition 1, we define a CBF for each subsystem i as follows. Denote

$$\tilde{w}^i(u^i,\nu^i,y^i_u,y^i_\nu) \triangleq w^i_n(u^i,\nu^i,y^i_u,y^i_\nu) - w^i_d(\nu^i,y^i). \tag{26}$$

Then, define the CBF

$$b^i(t,x^i_t) \triangleq -\sum_{\tau=t_0}^{t-1} \tilde{w}^i(u^i_\tau,\nu^i_\tau,y^i_{u,\tau},y^i_{\nu,\tau}), \tag{27}$$

whose super-level set is given by

$$\mathcal{C}^i = \left\{x^i_t \in \mathbb{R}^{n_i} \mid b^i(t,x^i_t) \geq 0\right\}. \tag{28}$$

To use the CBF $b^i(t,x^i_t)$ to enforce dissipativity of the closed loop subsystem, we proceed as follows. Denote

$$\begin{aligned}\mathrm{D}^i(x^i_t,\nu^i_t) \triangleq \{u^i \in \mathbb{R}^{m_i} | &-\tilde{w}^i(u^i_t,\nu^i_t,y^i_{u,t},y^i_{\nu,t}) \\ &+\eta^i b^i(t,x^i_t) \geq 0\},\end{aligned} \tag{29}$$

where $\eta^i \in [0,\ 1]$ is a designer specified parameter. We can then state the following result.

Proposition 2 (CBF for dissipativity): Consider the problem formulation in Section II. If $u^i \in \mathrm{D}^i(x^i,\nu^i)$ at all time steps, then the subsystem (1) is dissipative with respect to input ν^i and output y^i with supply-function $w^i_d(\nu^i,y^i_\nu)$.

From Proposition 2, if the set $\mathrm{D}^i(x^i_t,\nu^i_t)$ is non-empty, then any control input $u^i_t \in \mathrm{D}^i(x^i_t,\nu^i_t)$ renders (1) dissipative with respective to the supply-function $w^i_d(\nu^i_t,y^i_{\nu_t})$. We can choose a particular control input in this set from other considerations, such as minimizing the control cost. We can also use this set to ensure that the control input from an RL algorithm ensures that the subsystem is dissipative as shown next.

B. Dissipativity Ensuring RL Policies

We now consider the case when an RL algorithm is used for designing the control inputs u^i and show how the input can be chosen to one that preserves the dissipativity of the closed-loop subsystem Σ^i_n with respective to the supply-function $w^i_d(\nu^i,y^i_\nu)$. The key idea is similar to shielded RL techniques [11], [41], [42] and uses the CBF-based characterization of the set of dissipativity ensuring controllers obtained above to both project the control policy and to guide the future exploration of the RL algorithm.

We assume that the RL algorithm proceeds in an episodic fashion. Let $\pi^{\mathrm{RL}_i}_k$ denote the policy at the k-th policy iteration of the RL algorithm. This policy will in general be stochastic and may be parameterized by some parameters θ^i_k that may correspond to, e.g., the neural network being used to learn the policy. The paramterization is not relevant to our arguments and to minimize notational complexity, we suppress it in the sequel. Let $u^{\mathrm{RL_i}}_k(x^i_t) \sim \pi^{\mathrm{RL}_i}_k(\cdot|x^i_t)$. Our algorithm proceeds by projecting this input on the set of dissipativity ensuring controllers. Specifically, we propose that the overall dissipativity ensuring control input (denoted by u^{DEC}) in the k-th episode takes the following structure:

$$u^{\mathrm{DEC}_i}_k(x^i_t) = u^{\mathrm{FF}_i}_k(x^i_t) + u^{\mathrm{CBF}_i}_k(x^i_t,u^{\mathrm{FF}_i}_k) \tag{30}$$

where $u^{\mathrm{FF}_i}_k(x^i)$ represents the feedforward compensation, given by

$$u^{\mathrm{FF}_i}_k(x^i_t) = u^{\mathrm{RL_i}}_k(x^i_t) + \sum_{j=0}^{k-1} u^{\mathrm{CBF}_i}_j(x^i_t,u^{\mathrm{FF}_i}_j(x^i_t)) \tag{31}$$

and $u^{\mathrm{CBF}_i}_k$ is computed using the optimization problem:

$$\begin{aligned}u^{\mathrm{CBF}_i}_k(x^i_t,u^{\mathrm{FF}_i}_k) = \arg&\min_{a^i_t \in \mathbb{R}^{m_i}} \|a^i_t\| \\ &\text{s.t.} -\tilde{w}(u^i_t,\nu^i_t,y^i_{u_t},y^i_\nu) + \eta^i b^i(t,x^i_t) \geq 0, \\ &a^i_t + u^{\mathrm{FF}_i}_k(x^i_t) = u^i_t.\end{aligned} \tag{32}$$

As in the usual CBF-based works, the formulation in the relation (30) seeks to minimize the energy of the perturbation needed to project the control input in the set of dissipativity ensuring controllers [11], [37]. The feedforward compensation in (31) is split into two parts: $u^{\mathrm{RL}_i}_k(x^i)$ represents the control input obtained from the RL policy. However, this might not ensure dissipativity of the closed loop subsystem. The second term in (31) represents our best guess to rectify the input to ensure dissipativity. Furthermore, the term u^{CBF_i} in (30) may be interpreted as the feedback part of the controller. The complete algorithm description is given in Algorithm III-B.

We assume that the parameter $Max\_Episodes$ has been chosen to be large enough that the algorithm converges. Upon convergence, denote $u^{\mathrm{DEC}_i}(x^i_t)$ to be the final deployed controller $u^i_k(x^i_t)$ for $k = Max\_Episodes$. The following result shows that Algorithm III-B renders the closed loop subsystem dissipative. For brevity, we skip the proof as it is a direct consequence of Proposition 2 and Definition 2.

Proposition 3: Consider the problem formulation in Section II. Let the controller $u^{\mathrm{DEC}_i}(x^i_t)$ designed with Algorithm III-B be used as the input u^i_t for subsystem (1). If there exists a solution to the optimization problem (32) for all (x^i,ν^i), then the closed-loop subsystem (30) is dissipative with supply-function $w^i_d(\nu^i,y^i_\nu)$.

Remark 2: Computing $u^{\mathrm{FF}_i}_k(x)$ requires the solution of k optimization problem. Specifically, from the second term on the right side of (31), to compute $u^{\mathrm{FF}_i}_k(x^i_t)$, we need to solve $k-1$ optimization problems of the form defined in (32). Further, to compute $u^{\mathrm{FF}_i}_k(x^i_t)$ using (31), we need to solve it one more time. Thus, in total, we need to solve k optimization problems. Further,

Algorithm 1: RL-DEC Algorithm.

```
for $i = 1, \dots, N$ do
    Initialize RL input $\pi_0^{\mathrm{RL}_i}$, and arrays $\hat{D}^i$ and $\hat{A}^i$.
end
for $t = 0, \dots, T$ do
    for $i = 1, \dots, N$ do
        Sample $u_0^{\mathrm{RL}_i}(x_t^i) \sim \pi_0^{\mathrm{RL}_i}$ and compute
          $u_0^{\mathrm{CBF}_i}(x_t^i, u_0^{FF_i})$ using (32).
        Deploy $u_0^i(x_t^i) = u_0^{\mathrm{RL}_i}(x_t^i) + u_0^{\mathrm{CBF}_i}(x_t^i, u_0^{FF_i})$
        Store state-action pairs $(x_t^i, u_0^{\mathrm{CBF}_i}(x_t^i, u_0^{FF_i}))$
          in $\hat{A}^i$
    end
    for $i = 1, \dots, N$ do
        Observe $x_t^i, u_0^i(x_t^i), x_{t+1}^i, r_t^i$ and store in $\hat{D}^i$ for
          use in the RL algorithm
    end
end
for $i = 1, \dots, N$ do
    Collect Episode Reward $\sum_{t=1}^{T} r_t^i$
end
Set $k = 1$ (representing the $k$-th episode or input
  iteration step)
while $k < Max\_Episodes$ do
    for $i = 1, \dots, N$ do
        Do input iteration using RL algorithm based on
          previously observed episode to obtain $\pi_k^{\mathrm{RL}_i}$
    end
    Initialize state $s_0$ from an initial state distribution
    for $t = 0, \dots, T$ do
        for $i = 1, \dots, N$ do
            Compute the feed-forward term $u_k^{\mathrm{FF}_i}(x_t^i) =$
              $u_k^{\mathrm{RL_i}}(x_t^i) + \sum_{j=0}^{k-1} u_j^{\mathrm{CBF}_i}(x_t^i, u_j^{FF_i}(x_t^i))$
            Use (32) solve for $u_k^{\mathrm{CBF}_i}(x_t^i, u_k^{FF_i})$
            Deploy controller
              $u_k^i(x_t^i) = u_k^{\mathrm{FF}_i}(x_t^i) + u_k^{\mathrm{CBF}_i}(x_t^i, u_k^{\mathrm{FF}_i}(x_t^i))$
            Store state-action pairs $(x_t^i, u_k^i(x_t^i))$
        end
        for $i = 1, \dots, N$ do
            Observe $x_t^i, u_k^i(x_t^i), x_{t+1}^i, r_t^i$ and store in
              $\hat{D}^i$ for use in the RL algorithm
        end
    end
    $k = k + 1$
end
```

the knowledge of all $u_0^{\mathrm{RL_i}}, \dots, u_{k-1}^{\mathrm{RL}_i}$ is required. Consequently, for large k, the proposed algorithm can become memory intensive and computationally expensive. However, we need not compute $u_k^{\mathrm{FF}_i}(x)$ very accurately because of the presence of the feedback term $u_k^{CBF_i}$. This raises the possibility of approximating $u_k^{\mathrm{FF}_i}(x)$ by using a feed-forward neural network $u_{\phi_k}^{\mathrm{bar}}$ to learn the term $\sum_{j=0}^{k-1} u_j^{\mathrm{CBF}_i}$. In this case, (31) should be replaced by

$$u_k^{\mathrm{FF}_i}(x) = u_k^{\mathrm{RL_i}}(x) + u_{\phi_k^i}^{\mathrm{bar}}(x) \tag{33}$$

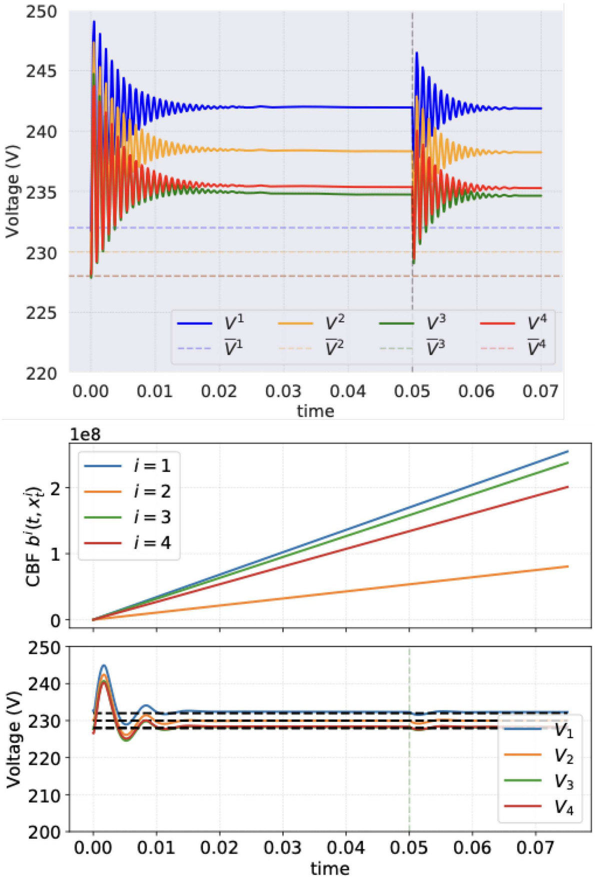


Fig. 4. (Top) Time evolution of voltage with trained RL controller without proposed CBF component. (Middle) If the CBF component is included, time evolution of CBF, (Bottom) and voltage across the load of each DGU, considering a load variation of 5% at time $t = 0.05$ s.

where ϕ_k parameterizes the neural network, which is updated using the data from previously collected samples.

The following is the main result of the article, which shows that the controller calculated using Algorithm III-B stabilizes the networked system.

Theorem 4 (Stability of networked system in closed-loop): Consider the problem formulation in Section II with Assumption 4. If u_t^i is chosen to be equal to $u^{\mathrm{DEC}_i}(x_t^i)$ at all time steps and for all subsystems i, then the networked system defined by (1), (5), and (8) is Lyapunov stable with respect to the origin. Further, suppose that $\mathcal{B}_\delta(\delta_{\min}) > 0$, $\mathcal{B}_\epsilon(\epsilon_{\min}) > 0$, and $R_u \triangleq \mathrm{diag}(R_u^i) > 0$. If the systems (1) and (5) are zero state detectable, then the networked system defined by (1), (5), and (8) is also asymptotically stable with respect to the origin.

The definition of *zero-state detectability* is provided in Definition 3 of Appendix A.

Remark 3 (Decentralized and Distributed): In (32), each agent needs to evaluate $\tilde{w}$ which requires the information of ν_t. From (8), computing ν_t requires information from its neighbors. Then, the proposed RL algorithm is distributed. However, in the event when the desired supply-function w_d is equal to w_ν, then $\tilde{w} = w_u$. Consequently, the RL algorithm takes a decentralized form.

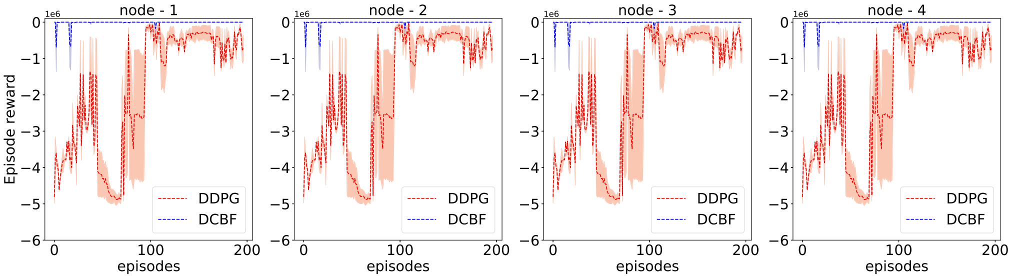


Fig. 5. Comparison of accumulated rewards from nodes of dc microgrid for each episode during training using DDPG and the proposed dissipative CBF approach.

IV. Case Study: DC Microgrid

We now evaluate the proposed CBF-based RL Algorithm III-B in simulation. We consider the dc microgrid in Example 1 with 4 DGUs, interconnected through resistive and inductive lines as shown in Fig. 3. The control objective is to regulate the voltage V^i across the load of each DGU to its desired value $\overline{V}^i \in \mathbb{R}$. Thus, we define the set of all feasible forced equilibria of the node subsystems (12) and the edge subsystems (14) as

$$\mathcal{C}_i^n = \Big\{(\overline{I}^i, \overline{V}^i, \overline{u}^i, \overline{\nu}^i) \in \mathbb{R}^4 |\ R^i\overline{I}^i + \overline{V}^i - \overline{u}^i V_s^i = 0,\ \overline{I}^i - G\overline{V}^i + \overline{\nu}^i = 0\Big\} \tag{34}$$

and

$$\mathcal{C}_k^e = \Big\{(\overline{I}_l^i, \overline{\mu}^i) \in \mathbb{R}^2 | R_l^i \overline{I}_l^i + \overline{\mu}^i = 0\Big\} \tag{35}$$

respectively. In the development above, we have assumed that $(s = 0)$ is the desired equilibrium. However, the results are agnostic to the choice of the equilibrium. Since the objective in this case study is to stabilize the system at a nontrivial operating point $(\overline{I}^i, \overline{V}^i, \overline{u}^i, \overline{\nu}^i, \overline{I}_l^i, \overline{\mu}^i) \in \mathcal{C}_i^n \times \mathcal{C}_k^e$, we shift the equilibrium of the networked system to the trivial equilibrium via a simple change of variables. In what follows, for a given variable ν, denote the error between $\tilde{\nu} = \nu - \overline{\nu}$.

In [43], the authors show that the subsystems at the node (12) and the edge (14) are dissipative with the supply-functions

$$w_n^i(\tilde{u}^i, \tilde{\nu}^i, \tilde{y}_u^i, \tilde{y}_\nu^i) = \underbrace{\tilde{u}^{i\top}\tilde{y}_u^i - R^i\|\tilde{y}_u^i\|_2^2}_{w_u^i(\tilde{u}^i, \tilde{y}_u^i)} + \underbrace{\tilde{\nu}^{i\top}\tilde{y}_\nu^i - G^i\|\tilde{y}_\nu^i\|_2^2}_{w_\nu^i(\tilde{\nu}^i, \tilde{y}_\nu^i)} \tag{36}$$

and

$$w_e^k(\tilde{\mu}^k, \tilde{\omega}^k) = \tilde{\mu}^{k\top}\tilde{\omega}^k - R_l^k\|\tilde{\omega}^k\|_2^2 \tag{37}$$

respectively. As a next step, we define the desired supply-function corresponding to (21) as

$$w_d^i(\tilde{\nu}^i, \tilde{y}^i) = w_\nu^i(\tilde{\nu}^i, \tilde{y}_\nu^i) - R^i\|\tilde{y}_u^i\|_2^2$$

where we chose $\delta_d^i = 0$, $\epsilon_d^i = R^i$, which satisfies (22) in Assumption 4. Consequently, using (26) we compute the resulting CBF as

$$b^i(t, x_t^i) = -\sum_{\tau=t_0}^{t-1}\left(\tilde{u}^{i\top}\tilde{y}_u^i - G^i\|\tilde{y}_\nu^i\|_2^2\right),\ t \geq t_0 \geq 0 \tag{38}$$

and its super-level is defined as in (28). Finally, we define the instantaneous reward function at each node as

$$r^i(V^i) := -k^i\left(\tilde{V^i}\right)^2 \tag{39}$$

where $k^i \in \mathbb{R}_{>0}$. For numerical simulation, the parameters of the microgrids are taken from [43, Tables 3 and 4].

Though the general framework described in the preceding can be used with almost any RL algorithm, we chose to use deep deterministic policy gradient (DDPG) [44] to showcase the performance of Algorithm III-B. Fig. 5 compares the accumulated rewards of vanilla DDPG and the proposed dissipativity-ensuring Algorithm III-B using DDPG during training. As the plot shows, Algorithm III-B coupled with DDPG converges faster that the vanilla DDPG algorithm; however, this may not be a general observation.

Next, we validate the performance of the controllers designed using the proposed approach. The voltage across the load and the value of the CBF at each node are plotted in Fig. 4. At $t = 0$ s, we start by initializing the microgrid near the desired operating point. We observe that the voltage signals stabilize to their desired values. However, in the dc microgrid, the value of load G^i is unknown and subject to change over time. To verify the robustness of the controller with respect to this uncertainty, the load at each DGU was increased by 5% of its original value at $t = 0.05$ s. In Fig. 4 we see that, after a minor perturbation, the voltage signals again stabilized to their desired values. Furthermore, the CBF is positive, thus validating the dissipativity-ensuring nature of the proposed approach.

V. Conclusion

We considered the problem of designing distributed controllers to stabilize a class of networked systems, where each subsystem is dissipative. We assumed that each subsystem designs a local controller using reinforcement learning to optimize its own reward function. We develop an approach that enforces dissipativity conditions on the local controller design to guarantee stability of the entire networked system. The proposed approach was illustrated on a microgrid example.

Appendix A
Dissipativity

Consider the following discrete time nonlinear system with state $x \in \mathbb{R}^n$ and inputs $a \in \mathbb{R}^m$

$$\begin{cases} x_{t+1} &= f(x_t, a_t), \\ y_t &= h(x_t, a_t) \end{cases} \tag{40}$$

where the functions f, h as assumed to be sufficiently smooth. Consider the mapping $w : \mathbb{R}^m, \mathbb{R}^m \to \mathbb{R}$. Then, dissipativity of system Σ with $w(a_t, y_t)$ as supply-function is defined as follows:

Definition 2 (Dissipativity [45]): System (40) is said to be dissipative with respect to the supply-function $w(a_t, y_t)$, if there exists a nonnegative function $\mathcal{V} : \mathbb{R}^n \to \mathbb{R}_+$, called as storage function, satisfying $\mathcal{V}(0) = 0$ such that for all $s_{t_0} \in X$, all $t > t_0 \geq 0$ and all $a_t \in A$

$$\mathcal{V}(x_t) - \mathcal{V}(s_{t_0}) \leq -\sum_{i=t_0}^{t-1} \mathcal{D}(x_t) + \sum_{i=t_0}^{t-1} w(a_t, y_t) \tag{41}$$

where $\mathcal{D}(x_t) \in \mathbb{R}_+$ is a nonnegative function, and s_t is the state at time t, resulting from state s_{t-1} with input u_{t-1}. Furthermore, we call the system QSR *dissipative* if the inequality (41) holds with

$$w(a_t, y_t) = -\|y_t\|_Q^2 + a_t^\top S y_t - \|a_t\|_R^2 \tag{42}$$

where $Q = Q^\top$, S, and $R = R^\top$ are matrices of appropriate dimensions.

Definition 3 (Zero-state detectability): Consider (40) with $f(0,0) = 0$, and $h(0,0) = 0$. Then system (40) is called zero-state detectable if $a_t = 0$ *and* $y_t = 0 \Rightarrow \lim_{t\to\infty} x_t \to 0$.

Appendix B
Proofs

Proof of Proposition 1: Without loss of generality, we assume the initial state as $s_0 \in \rho_0$ at time $t = 0$ and $b(0, s_0) \geq 0$. It suffices to show that $b(t, s_t) \geq 0$, for all $a_t \in \mathrm{B}(t, s_t)$. From (24) and (25), for all $a_t \in \mathrm{B}(t, s_t)$, we have

$$b(t+1, s_{t+1}) \geq (1-\eta) b(t, s_t). \tag{43}$$

Now, consider the following boundary value problem:

$$X_{t+1} = (1-\eta) X_t \tag{44}$$

with initial condition $X_0 = b(0, s_0) \geq 0$. Then, the solution to (44) is $X_t = (1-\eta)^t X_0 \geq 0, \forall k \in \mathbb{Z}^+, 0 < \eta \leq 1$. From (43) and (44)

$$b(t, s_t) \geq X_t. \tag{45}$$

Thus, $\mathcal{C}$ is forward invariant.

Proof of Proposition 2: Consider the barrier function $b^i(t, x_t^i)$ defined in (28). From Proposition 1, for all $u_t \in \mathrm{D}^i(x_t^i, \nu_t)$, it implies that $\mathcal{C}^i$ is forward invariant. Consequently, we have $b^i(t, x_t^i) = -\sum_{\tau=t_0}^{t-1} \tilde{w} \geq 0$

$$\Rightarrow \sum_{\tau=t_0}^{t-1} \tilde{w} \leq 0 \tag{46}$$

$$\Rightarrow \sum_{\tau=t_0}^{t-1} (w_n - w_d) \leq 0 \Rightarrow \sum_{\tau=t_0}^{t-1} w_d \geq \sum_{\tau=t_0}^{t-1} w_n. \tag{47}$$

From Assumption 1 the subsystem (1) is dissipative, which further implies $\sum_{\tau=t_0}^{t-1} w_d \geq \sum_{\tau=t_0}^{t-1} w_n \geq 0$. From Definition 2, we conclude the proof.

Proof of Theorem 4: As a consequence of Assumption 1, Proposition 3 implies that node dynamics in closed-loop with control input (30) are dissipative with supply-function (21) $w_d^i(\nu^i, y^i)$. Consequently, for all $i \in \mathcal{V}$ there exists a storage function $\mathcal{V}_d^i : \mathbb{R}^n \to \mathbb{R}_+$, satisfying

$$\mathcal{V}_d^i(x_t^i) \leq \mathcal{V}_d^i(x_{t_0}^i) + \sum_{t=t_0}^{t-1} w_d^i(\nu^i, y^i). \tag{48}$$

From Assumption 2, the edge dynamics are dissipative with supply-function $w_e^k(\mu^k, \omega^k)$. Consequently, for all $k \in \{1, \ldots, M\}$, there exists a storage function $\mathcal{V}_e^i : \mathbb{R}^m \to \mathbb{R}_+$, satisfying

$$\mathcal{V}_e^k(z_t^k) \leq \mathcal{V}_e^i(z_{t_0}^k) + \sum_{t=t_0}^{t-1} w_e^k(\mu_t^k, \omega_t^k). \tag{49}$$

Consider $\mathcal{V}(s_t) = \sum_{i=1}^N \mathcal{V}_d^i(x_t^i) + \sum_{k=1}^M \mathcal{V}_e^k(z_t^k)$, consequently

$$\mathcal{V}(s_t) - \mathcal{V}(s_{t_0}) \tag{50a}$$

$$\leq \sum_{i=1}^{N} \sum_{t=t_0}^{t-1} w_d^i(\nu^i, y^i) + \sum_{k=1}^{M} \sum_{t=t_0}^{t-1} w_e^k(\mu_t^k, \omega_t^k)$$

$$= \sum_{t=t_0}^{t-1} \sum_{i=1}^{N} w_d^i(\nu^i, y^i) + \sum_{t=t_0}^{t-1} \sum_{k=1}^{M} w_e^k(\mu_t^k, \omega_t^k)$$

$$\leq \sum_{t=t_0}^{t-1} \big(\nu^\top S_\nu^\top y_\nu - \delta_{\min}\|\nu\|_2^2 - \epsilon_{\min}\|y\|_2^2 + \mu^\top S_e^\top \omega - \delta_e\|\mu\|_2^2$$

$$-\epsilon_e\|\omega\|_2^2\big) \tag{50b}$$

$$\leq \sum_{t=t_0}^{t-1} \big(\omega^\top \mathcal{B}^\top S_\nu^\top y_\nu - \delta_{\min}\|\mathcal{B}\omega\|_2^2 - \epsilon_{\min}\|y_\nu\|_2^2 - \epsilon_{\min}\|y_u\|_2^2$$

$$-y_\nu^\top \mathcal{B}^\top S_e^\top \omega - \delta_e\|\mathcal{B} y_\nu\|_2^2 - \epsilon_e\|\omega\|_2^2\big) \tag{50c}$$

$$= -\sum_{t=t_0}^{t-1} \left(\|\omega\|_{\mathcal{B}_\delta(\delta_{\min})}^2 + \|y_\nu\|_{\mathcal{B}_\epsilon(\epsilon_{\min})}^2 + \epsilon_{\min}\|y_u\|_2^2\right). \tag{50d}$$

In (50a), we use (48) and (49). In (50b), we use the interconnection laws from (8). In (50c), we use Assumption 3. This implies that the overall networked system is stable.

Furthermore, consider $\mathcal{B}_\delta(\delta_{\min}) > 0$ and $\mathcal{B}_\epsilon(\epsilon_{\min}) > 0$. Then from (50d) there exists a forward invariant set Π and by LaSalle's invariance principle, the solutions that start in Π converge to the largest invariant set contained in $\Pi \cap \{s \in \mathbb{R}^{n+p} | \, \omega = 0, \, y = 0\}$. Moreover, from (8) this implies $\mu = 0$, $\nu = 0$. From Assumption 1 and $R_u > 0$, this further implies that $u = 0$. Finally on this set, we have $(y = 0, u = 0, \nu = 0)$ and $(\omega = 0, \mu = 0)$. Given that that subsystems (1) and (5) are zero-state detectable, following [46, Corollary 4.2.2], the trajectories in Π converges asymptotically to the largest invariant set contained in $\Pi \cap \{s = 0\}$.

References

[1] M. Egerstedt and X. Hu, “Formation constrained multi-agent control,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 6, pp. 947–951, Dec. 2001.

[2] S. Sivaranjani, S. Sadraddini, V. Gupta, and C. Belta, “Distributed control policies for localization of large disturbances in urban traffic networks,” in *Proc. Amer. Control Conf.*, 2017, pp. 3542–3547.

[3] T. Dragičević, X. Lu, J. C. Vasquez, and J. M. Guerrero, “DC microgrids—Part I: A review of control strategies and stabilization techniques,” *IEEE Trans. Power Electron.*, vol. 31, no. 7, pp. 4876–4891, Jul. 2016.

[4] F. Horn and R. Jackson, “General mass action kinetics,” *Arch. Rational Mechanics Anal.*, vol. 47, no. 2, pp. 81–116, 1972.

[5] R. H. Lasseter and P. Paigi, “Microgrid: A conceptual solution,” in *Proc. IEEE 35th Annu. Power Electron. Specialists Conf.*, 2004, vol. 6, pp. 4285–4290.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[7] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[8] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Trans. Syst., Man, Cybern., Part C. (Appl. Rev.)*, vol. 38, no. 2, pp. 156–172, Mar. 2008.

[9] K. Zhang, Z. Yang, and T. Başar, “Multi-agent reinforcement learning: A selective overview of theories and algorithms,” 2019, *arXiv:1911.10635*.

[10] K. Zhang, Z. Yang, and T. Başar, “Decentralized multi-agent reinforcement learning with networked agents: Recent advances,” 2019, *arXiv:1912.03821*.

[11] R. Cheng, G. Orosz, R. M. Murray, and J. W. Burdick, “End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks,” in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, pp. 3387–3395.

[12] L. Buşoniu, T. de Bruin, D. Toli C, J. Kober, and I. Palunko, “Reinforcement learning for control: Performance, stability, and deep approximators,” *Annu. Rev. Control*, vol. 46, pp. 8–28, 2018.

[13] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, “Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers,” *IEEE Control Syst. Mag.*, vol. 32, no. 6, pp. 76–105, Dec. 2012.

[14] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, “Safe model-based reinforcement learning with stability guarantees,” in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 908–918.

[15] M. Fazel, R. Ge, S. M. Kakade, and M. Mesbahi, “Global convergence of policy gradient methods for the linear quadratic regulator,” 2018, *arXiv:1801.05039*.

[16] K. Zhang, B. Hu, and T. Başar, “Policy optimization for $\mathcal{H}_2$ linear control with $\mathcal{H}_\infty$ robustness guarantee: Implicit regularization and global convergence,” 2019, *arXiv:1910.09496*.

[17] J. C. Willems, “Dissipative dynamical systems—Part II: Linear systems with quadratic supply rates,” *Arch. Rational Mechanics Anal.*, vol. 45, no. 5, pp. 352–393, 1972.

[18] A. J. Van der Schaft, *L_2-Gain and Passivity Techniques in Nonlinear Control*, vol. 2, Berlin, Germany: Springer, 2000.

[19] C. A. Desoer and M. Vidyasagar, *Feedback Systems: Input-Output Properties*. Philadelphia, PA, USA: SIAM, 2009.

[20] G. Niemeyer and J. E. Slotine, “Stable adaptive teleoperation,” *IEEE J. Ocean. Eng.*, vol. 16, no. 1, pp. 152–162, Jan. 1991.

[21] N. Chopra and M. W. Spong, “Passivity-based control of multi-agent systems,” in *Advances in Robot Control*, Springer, 2006, pp. 107–134.

[22] M. Arcak, C. Meissen, and A. Packard, *Networks of Dissipative Systems: Compositional Certification of Stability, Performance, and Safety*. Berlin, Germany: Springer, 2016.

[23] A. Van der Schaft and B. Maschke, “Port-Hamiltonian systems on graphs,” *SIAM J. Control Optim.*, vol. 51, no. 2, pp. 906–937, 2013.

[24] E. Agarwal, *Compositional Control of Large-Scale Cyber-Physical Systems Using Hybrid Models and Dissipativity Theory*. University of Notre Dame, 2019.

[25] E. Agarwal, S. Sivaranjani, V. Gupta, and P. J. Antsaklis, “Distributed synthesis of local controllers for networked systems with arbitrary interconnection topologies,” *IEEE Trans. Autom. Control*, vol. 66, no. 2, pp. 683–698, Feb. 2021.

[26] K. C. Kosaraju, M. Cucuzzella, J. M. A. Scherpen, and R. Pasumarthy, “Differentiation and passivity for control of Brayton-Moser systems,” *IEEE Trans. Autom. Control*, vol. 66, no. 3, pp. 1087–1101, Mar. 2021.

[27] M. J. Tippett and J. Bao, “Dissipativity based distributed control synthesis,” *J. Process Control*, vol. 23, no. 5, pp. 755–766, 2013.

[28] S. Sivaranjani, E. Agarwal, L. Xie, V. Gupta, and P. Antsaklis, “Mixed voltage angle and frequency droop control for transient stability of interconnected microgrids with loss of PMU measurements,” in *Proc. Amer. Control Conf.*, 2020, pp. 2382–2387.

[29] E. Agarwal, S. Sivaranjani, V. Gupta, and P. J. Antsaklis, “Sequential synthesis of distributed controllers for cascade interconnected systems,” in *Proc. Amer. Control Conf.*, 2019, pp. 5816–5821.

[30] B. Gao and L. Pavel, “On passivity, reinforcement learning and higher-order learning in multi-agent finite games,” *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 121–136, Jan. 2021.

[31] S. P. Nageshrao, G. A. Lopes, D. Jeltsema, and R. Babuška, “Passivity-based reinforcement learning control of a 2-DOF manipulator arm,” *Mechatronics*, vol. 24, no. 8, pp. 1001–1007, 2014.

[32] O. Sprangers, R. Babuška, S. P. Nageshrao, and G. A. Lopes, “Reinforcement learning for port-Hamiltonian systems,” *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1017–1027, May 2015.

[33] A. Kanellopoulos, K. G. Vamvoudakis, and V. Gupta, “Decentralized verification for dissipativity of cascade interconnected systems,” in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 3629–3634.

[34] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs for safety critical systems,” *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[35] M. Z. Romdlony and B. Jayawardhana, “Uniting control Lyapunov and control barrier functions,” in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 2293–2298.

[36] P. Wieland and F. Allgöwer, “Constructive safety using control barrier functions,” *IFAC Proc. Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

[37] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *Proc. 18th Eur. Control Conf.*, 2019, pp. 3420–3431.

[38] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, “Robustness of control barrier functions for safety critical control,” *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[39] G. Notomista and M. Egerstedt, “Persistification of robotic tasks,” *IEEE Trans. Control Syst. Technol.*, vol. 3, no. 2, pp. 758–763, Apr. 2018, doi: 10.1109/LRA.2018.2789848.

[40] G. Notomista, X. Cai, J. Yamauchi, and M. Egerstedt, “Passivity-based decentralized control of multi-robot systems with delays using control barrier functions,” in *Proc. Int. Symp. Multi-Robot Multi-Agent Syst.*, 2019, pp. 231–237.

[41] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, “Safe reinforcement learning via shielding,” in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2669–2678.

[42] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, “A general safety framework for learning-based control in uncertain robotic systems,” *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2737–2752, Jul. 2019.

[43] M. Cucuzzella, K. C. Kosaraju, and J. Scherpen, “Voltage control of DC networks: Robustness for unknown zip-loads,” 2019, *arXiv:1907.09973*.

[44] T. P. Lillicrap *et al.*, “Continuous control with deep reinforcement learning,” 2015, *arXiv:1509.02971*.

[45] E. Navarro-López, D. Cortés, and E. Fossas-Colet, “Implications of dissipativity and passivity in the discrete-time setting,” *IFAC Proc. Volumes*, vol. 35, no. 1, pp. 55–60, 2002.

[46] A. J. van der Schaft, *L_2-Gain and Passivity Techniques in Nonlinear Control*. Berlin, Germany: Springer, 2000.

Krishna Chaitanya Kosaraju received the bachelor’s degree in electronics and instrumentation from the Birla Institute of Technology and Science, Pilani, Rajasthan, India, in 2010. He received the master’s degree in control and instrumentation and the Ph.D. degree in electrical engineering from the Indian Institute of Technology Madras, Chennai, India, in 2013 and 2018, respectively.

He is currently a Postdoctoral Researcher with the University of Notre Dame, IN, USA. From 2018 to 2019, he was a Postdoctoral Researcher with the University of Groningen, Groningen, the Netherlands. His research interests include nonlinear control theory, passivity-based control and optimization theory with applications to power networks, building systems, and reinforcement learning.

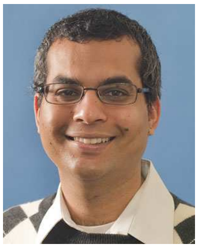
S. Sivaranjani received the B.E. degree from PES Institute of Technology, Bengaluru, India, M.S. degree from the Indian Institute of Science, Bengaluru, India, and Ph.D. degree from the University of Notre Dame, IN, USA, respectively, all in electrical engineering.

She is currently a 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.

Wesley Suttle received the B.A. degree in mathematics and philosophy from the University of Minnesota, Twin Cities, MN, USA. He is currently working toward the Ph.D. degree in applied mathematics and statistics with Stony Brook University, Stony Brook, NY, USA.

He was recently a Graduate Research Associate with the U.S. Army Research Laboratory. His research interests include reinforcement learning for ratio optimization problems and multiagent reinforcement learning.

Dr. Suttle was the winner of the ARL 2021 Summer Student Symposium Best Presentation Award.



Vijay Gupta received the B.Tech. degree from the Indian Institute of Technology, Delhi, India and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, CA, USA, all in electrical engineering, in 2001, 2002, and 2006 respectively.

He joined the faculty at the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN, USA in 2008. His research and teaching interests include the interface of communication, control, distributed computation, and human decision making.

Dr. Gupta received the 2018 Antonio J Rubert Award from the IEEE Control Systems Society, the 2013 Donald P. Eckman Award from the American Automatic Control Council, and a 2009 National Science Foundation (NSF) CAREER Award.

Ji Liu received the B.S. degree in information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 2013.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY, USA. Prior to that, he was a Postdoctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, IL, USA, and the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA. He is an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. His current research interests include distributed control and optimization, distributed reinforcement learning, multiagent systems, epidemic networks, social networks, and cyber-physical systems.