# Reinforcement Learning based Distributed Control of Dissipative Networked Systems

K. C. Kosaraju, S. Sivaranjani, W. Suttle, V. Gupta, and J. 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 maintain voltage stability of the network using local distributed controllers at each generation unit.

### 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, multi-body mechanical systems, and microgrids [1]–[5]. The problem provides many challenges such as non-classical 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 for distributed and multi-agent control. Reinforcement Learning (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 are 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 multi-agent 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].

In this paper, 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 a wide literature on RL techniques for multi-agent 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 multi-agent 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 RLbased 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 towards 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 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. 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 paper, we develop a reinforcement learning based

K.C. Kosaraju, V. Gupta are with the Department of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA (email: {kkosaraj, vgupta2}@nd.edu).

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

W. Suttle and J. Liu are with the Department of Applied Mathematics and Statistics and the Department of Electrical and Computer Engineering at Stony Brook University, Stony Brook, NY 11794, USA (email: {wesley.suttle, ji.liu}@stonybrook.edu).

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 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 control barrier function (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 super-level set of a function (typically representing specifications like safety) is forward invariant under a given dynamics [33]–[35]. 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 multi-agent 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 paper 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 Direct-Current microgrid application. Finally, in Section V, we provide some directions for future work. Proofs of all the results in the paper, 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



Figure 1. Schematic of the system configuration

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  $M_1, M_2, \dots, M_n$ , define the matrix diag $(M_i)$  as the block diagonal matrix whose main-diagonal blocks are matrices  $M_1$ ,  $M_2, \dots, M_n$ , 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  $||f(b) - f(a)||_2 < L||b - a||_2$  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 shown in Figure 1.

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

$$\Sigma_{n}^{i}: \begin{cases} x_{t+1}^{i} = f^{i}(x_{t}^{i}, u_{t}^{i}, \nu_{t}^{i}) \\ y_{u,t}^{i} = g^{i}(x_{t}^{i}, u_{t}^{i}) \\ y_{\nu,t}^{i} = h^{i}(x_{t}^{i}, \nu_{t}^{i}) \end{cases}$$
(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  $y_{\nu,t}^i \in \mathbb{R}^{\widehat{o}_i}$  which is the output that is used to compute the inputs  $\nu_t^j$  for other subsystems *j* for whom *i* is an in-neighbor. We will define the exact relation between  $\nu_t^i$  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  $\Sigma_n^i$ .

an equilibrium point of the subsystem  $\Sigma_{i}^{i}$ . For future reference, define  $x^{\top} \triangleq [x^{1\top}, \dots, x^{N\top}] \in \mathbb{R}^{n}$ ,  $u^{\top} \triangleq [u^{1\top}, \dots, u^{N\top}] \in \mathbb{R}^{m}$ ,  $y_{u}^{\top} \triangleq [y_{u}^{1\top}, \dots, y_{u}^{N\top}] \in \mathbb{R}^{\overline{o}}$ ,  $y_{\nu}^{\top} \triangleq [y_{\nu}^{1\top}, \dots, y_{\nu}^{N\top}] \in \mathbb{R}^{\hat{o}}$ ,  $y^{i} \triangleq [y_{u}^{i\top}, y_{\nu}^{i\top}] \in \mathbb{R}^{o}$ ,  $y^{\top} \triangleq [y^{1\top}, \dots, y^{N\top}] \in \mathbb{R}^{o}$ ,  $y^{\top} \triangleq [y^{1\top}, \dots, y^{N\top}] \in \mathbb{R}^{p}$ . As stated earlier definitions of  $u^{\top}$ 

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

**Assumption 1** (Dissipative node dynamics). Each subsystem  $\Sigma_n^i$  with dynamics defined in (1) is dissipative, in the set  $S_n^i$ , with respect to the supply function

$$w_{n}^{i}(u^{i},\nu^{i},y_{u}^{i},y_{\nu}^{i}) = \underbrace{u^{i\top}S_{u}^{i\top}y_{u}^{i} - \|u^{i}\|_{R_{u}^{i}}^{2} - \|y_{u}^{i}\|_{Q_{u}^{i}}^{2}}_{= w_{u}^{i}(u^{i},y_{u}^{i})} + \underbrace{\nu^{i\top}S_{\nu}^{i\top}y_{\nu}^{i} - \|\nu^{i}\|_{R_{\nu}^{i}}^{2} - \|y_{\nu}^{i}\|_{Q_{\nu}^{i}}^{2}}_{= w_{\nu}^{i}(\nu^{i},y_{\nu}^{i})}, \quad (2)$$

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

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

**Remark 1.** Even though Assumption 1 states that the subsystem is dissipative, it is an assumption in the 'open loop'. Note that the design of the controller that determines the inputs  $u^i$  has not been specified. The dissipativity property required for system stability concerns the inputs  $\mu^i$  and the outputs  $y^i_{\mu}$  and this may easily be disrupted by the additional dynamics, say

of the form  $u^i = \zeta^i(x^i)$ , introduced through the design of the controller. For a simple illustration of this fact, note that from [36, Corollary 4.1.5], Assumption 1 holds if and only if the condition

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

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

$$\sum_{t=t_0}^{t-1} \sum_{i=1}^{N} w_{\nu}^{i}(\nu_{t}^{i}, y_{\nu,t}^{i}) \ge -\sum_{t=t_0}^{t-1} \sum_{i=1}^{N} w_{u}^{i}(\zeta^{i}(x_{t}^{i}), y_{u,t}^{i}) \not\ge 0, \quad (4)$$

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  $\nu_t^i$  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

$$\Sigma_{\rm e}^{k}: \begin{cases} z_{t+1}^{k} = g^{k}(z_{t}^{k}, \mu_{t}^{k}) \\ \omega_{t}^{k} = j^{k}(z_{t}^{k}, \mu_{t}^{k}) \end{cases}$$
(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  $\Sigma_e^k$ . For future reference, define  $z^\top \triangleq [z^{1\top}, \ldots, z^{M\top}] \in \mathbb{R}^q, \, \omega^\top \triangleq [\omega^{1\top}, \ldots, \omega^{M\top}] \in \mathbb{R}^s$ , and  $\mu^\top \triangleq [\mu^{1\top}, \ldots, \mu^{M\top}] \in \mathbb{R}^r$ , where M denotes the cardinality of the set  $\mathcal{E}$ .

**Assumption 2** (Dissipative edge dynamics). Each subsystem  $\Sigma_e^k$  with its dynamics defined in (5) is dissipative in the set  $S_e^k$  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  $S_e \triangleq \operatorname{diag}(S_e^k)$ ,  $R_e \triangleq \operatorname{diag}(R_e^k)$ , and  $Q_e \triangleq \operatorname{diag}(Q_e^k)$ .

**Interconnection among subsystems:** The entire networked system is defined through the interconnection of the subsystems defined by the nodes and edges by relating the inputs  $\nu$  and outputs  $y_{\nu}$  of the node subsystems with the inputs  $\mu$  and outputs  $\omega$  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

$$\begin{split} w_{u}(u, y_{u}) &\triangleq \left( u^{\top} S_{u}^{\top} y_{u} - \|u\|_{R_{u}}^{2} - \|y_{u}\|_{Q_{u}}^{2} \right), \\ w_{\nu}(\nu, y_{\nu}) &\triangleq \left( \nu^{\top} S_{\nu}^{\top} y_{\nu} - \|\nu\|_{R_{\nu}}^{2} - \|y_{\nu}\|_{Q_{\nu}}^{2} \right), \\ w_{e}(\mu, \omega) &\triangleq \left( \mu^{\top} S_{e}^{\top} \omega - \|\mu\|_{R_{e}}^{2} - \|\omega\|_{Q_{e}}^{2} \right). \end{split}$$
(7)



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



Figure 3. The topology of network considered in Example 1.

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

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

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

# **Assumption 3.** Matrices $S_{\nu}$ 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 equation (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

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

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

**Example 1.** Consider the electrical schematic of a microgrid, containing four Distributed Generating Units (DGUs) and interconnected through four transmission lines, as shown in Figures 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  $x_t^i \triangleq \begin{bmatrix} I_t^i & V_t^i \end{bmatrix}^\top$ . The dynamics of the DGU at node  $i \in \mathcal{V} := \{1 \dots 4\}$ , which forms the *i*-th subsystem, can be written as

$$I_{t+1}^{i} = I_{t}^{i} - (T_{s}/L^{i})(R^{i}I_{t}^{i} + V_{t}^{i} - u_{t}^{i}V_{s})$$
  

$$V_{t+1}^{i} = V_{t}^{i} + (T_{s}/C^{i})(I_{t}^{i} - G^{i}V_{t}^{i} + \nu_{t}^{i}),$$
(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

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

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

where  $L_l^k$ ,  $R_l^k \in \mathbb{R}_{>0}$  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

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

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

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

$$J^{i} = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{t} r_{t}^{i}(x_{t}^{i}, u_{t}^{i})\right], \qquad (17)$$

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 a RL algorithm to design its controller. For a given control policy  $\pi^i$ , we define the value function  $V_{\pi}^i$ , and the state-action value function  $Q_{\pi}^i$ below:

$$V_{\pi}^{i}(x^{i}) = \mathbb{E}_{\pi^{i}} \left[ \sum_{t=0}^{\infty} \gamma^{t} r_{t}^{i}(x_{t}^{i}, u_{t}^{i}) \mid x_{0}^{i} = x^{i} \right], \qquad (18)$$

$$Q_{\pi}^{i}(x^{i}, u^{i}) = \mathbb{E}_{\pi^{i}} \left[ \sum_{t=0}^{\infty} \gamma^{t} r_{t}^{i}(x_{t}^{i}, u_{t}^{i}) \mid x_{0}^{i} = x^{i}, u_{0}^{i} = u^{i} \right],$$
(19)

$$A^{i}_{\pi}(x^{i}, u^{i}) = Q^{i}_{\pi}(x^{i}, u^{i}) - V^{i}_{\pi}(x^{i}).$$
(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  $\Sigma$  under consideration, with state defined as  $s_t^{\top} \triangleq \left[x_t^{\top}, z_t^{\top}\right]$  . 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 REINFORCEMENT LEARNING

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

(a) Control barrier functions for dissipativity: As stated in Remark 1, even though each subsystem i is dissipative with supply-function w<sup>i</sup><sub>u</sub>(u<sup>i</sup>, y<sup>i</sup><sub>u</sub>) + w<sup>i</sup><sub>ν</sub>(ν<sup>i</sup>, y<sup>i</sup><sub>ν</sub>), with the controller for the input u<sup>i</sup> the subsystem may no longer remain dissipative with the input-output pair w<sup>i</sup><sub>ν</sub>(ν<sup>i</sup>, y<sup>i</sup><sub>ν</sub>). Our first step is to utilize control barrier functions to characterize the set of all controllers that ensure that the closed loop subsystem i is dissipative with the supply function w<sup>i</sup> (c.f. Fig 1) with the supply function

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

where  $\delta^i_d \in \mathbb{R}$  and  $\epsilon^i_d \in \mathbb{R}$  are tuning parameters set by the designer.

- (b) **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.
- (c) **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  $\alpha = \min(\delta_d^1, \ldots, \delta_d^N)$ , and  $\beta = \min(\epsilon_d^1, \ldots, \epsilon_d^N)$ . The conditions

$$\begin{aligned}
\mathcal{B}_{\delta}(\alpha) &\geq 0, \\
\mathcal{B}_{\epsilon}(\beta) &\geq 0,
\end{aligned}$$
(22)

hold, where  $\mathcal{B}_{\delta}$ , and  $\mathcal{B}_{\epsilon}$  have been defined in (10).

## A. Control barrier functions for dissipativity

Control barrier functions (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 Control Barrier Functions). 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 C as the super-level set of this function as follows:

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

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

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

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

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

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

**Proposition 1** (Discrete-time time-varying Control Barrier Functions). Consider a time-varying zeroing control barrier function  $b(t, s_t)$  and its super level set C defined in (23). Then any Lipschitz input  $u_t \in B(t, s_t)$ , where  $B(t, s_t)$  is given in (25), will render the set C forward invariant.

Although dissipativity is a property defined by the input, and the output, we can utilize control barrier functions 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 control barrier function for each subsystem i as follows. Denote

$$\tilde{w}^{i}(u^{i},\nu^{i},y^{i}_{u},y^{i}_{\nu}) \triangleq w^{i}_{n}(u^{i},\nu^{i},y^{i}_{u},y^{i}_{\nu}) - w^{i}_{d}(\nu^{i},y^{i}).$$
(26)

Then, define the control barrier function

$$b^{i}(t, x_{t}^{i}) \triangleq -\sum_{\tau=t_{0}}^{t-1} \tilde{w}^{i}(u_{\tau}^{i}, \nu_{\tau}^{i}, y_{u,\tau}^{i}, y_{\nu,\tau}^{i}), \qquad (27)$$

whose super-level set is given by

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

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

$$D^{i}(x_{t}^{i},\nu_{t}^{i}) \triangleq \{u^{i} \in \mathbb{R}^{m_{i}} | -\tilde{w}^{i}(u_{t}^{i},\nu_{t}^{i},y_{u,t}^{i},y_{\nu,t}^{i}) + \eta^{i}b^{i}(t,x_{t}^{i}) \ge 0\}, \quad (29)$$

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

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

From Proposition 2, if the set  $D^i(x_t^i, \nu_t^i)$  is non-empty, then any control input  $u_t^i \in D^i(x_t^i, \nu_t^i)$  renders (1) dissipative with respective to the supply function  $w_d^i(\nu_t^i, y_{\nu_t}^i)$ . 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 closedloop subsystem  $\Sigma_n^i$  with respective to the supply function  $w_d^i(\nu^i, y_{\nu}^i)$ . The key idea is similar to shielded RL techniques [11], [41], [42] and uses the control barrier function 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  $\pi_k^{\mathrm{RL}_i}$  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  $\theta_k^i$  that may correspond to, e.g., the neural network being used to learn the policy. The parameterization is not relevant to our arguments and to minimize notational complexity, we suppress it in the sequel. Let  $u_k^{\mathrm{RL}_i}(x_t^i) \sim \pi_k^{\mathrm{RL}_i}(\cdot | x_t^i)$ . Our algorithm proceeds by

projecting this input on the set of dissipativity ensuring controllers. Specifically, we propose that the overall dissipativity ensuring control input in the k-th episode takes the following structure:

$$u_{k}^{\text{DEC}_{i}}(x_{t}^{i}) = u_{k}^{\text{FF}_{i}}(x_{t}^{i}) + u_{k}^{\text{CBF}_{i}}(x_{t}^{i}, u_{k}^{\text{FF}_{i}}), \quad (30)$$

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

$$u_{k}^{\mathrm{FF}_{i}}(x_{t}^{i}) = u_{k}^{\mathrm{RL}_{i}}(x_{t}^{i}) + \sum_{j=0}^{k-1} u_{j}^{\mathrm{CBF}_{i}}(x_{t}^{i}, u_{j}^{\mathrm{FF}_{i}}(x_{t}^{i})), \quad (31)$$

and  $u_k^{\text{CBF}_i}$  is computed using the optimization problem:

$$\begin{aligned} u_{k}^{\text{CBF}_{i}}(x_{t}^{i}, u_{k}^{\text{FF}_{i}}) &= \arg\min_{a_{t}^{i} \in \mathbb{R}^{m_{i}}} \|a_{t}^{i}\| \\ \text{s.t.} &- \tilde{w}(u_{t}^{i}, \nu_{t}^{i}, y_{u_{t}}^{i}, y_{\nu}^{i}) + \eta^{i}b^{i}(t, x_{t}^{i}) \geq 0, \\ &a_{t}^{i} + u_{k}^{\text{FF}_{i}}(x_{t}^{i}) = u_{t}^{i}. \end{aligned}$$
(32)

As in the usual control barrier function 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:  $u_k^{\text{RL}_i}(x^i)$  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  $u^{\text{CBF}_i}$  in (30) may be interpreted as the feedback part of the controller. The complete algorithm description is given in Algorithm 1.

We assume that the parameter  $Max\_Episodes$  has been chosen to be large enough that the algorithm converges. Upon convergence, denote  $u^{\text{DEC}_i}(x_t^i)$  to be the final deployed controller  $u_k^i(x_t^i)$  for  $k = Max\_Episodes$ . The following result shows that Algorithm 1 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  $u^{DEC_i}(x_t^i)$  designed with Algorithm 1 be used as the input  $u_t^i$  for the subsystem (1). If there exists a solution to the optimization problem (32) for all  $(x^i, \nu^i)$ , then the closed-loop subsystem (30) is dissipative with supply function  $w_d^i(\nu^i, y_{\nu}^i)$ .

**Remark 2.** Computing  $u_k^{\text{FF}_i}(x)$  requires the solution of the optimization problem k times; further, the knowledge of all  $u_0^{\text{RL}_i}, \ldots, u_{k-1}^{\text{RL}_i}$  is required. Consequently, for large k, the proposed algorithm can become memory intensive and computationally expensive. However, we need not compute  $u_k^{\text{FF}_i}(x)$  very accurately because of the presence of the feedback term  $u_k^{CBF_i}$ . This raises the possibility of approximating  $u_k^{\text{FF}_i}(x)$  by using a feed-forward neural network  $u_{\phi_k}^{\text{bar}}$  to learn the term  $\sum_{j=0}^{k-1} u_j^{\text{CBF}_i}$ . In this case, (31) should be replaced by

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

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

#### Algorithm 1: RL-DEC algorithm.

for i = 1, ..., N do Initialize RL input  $\pi_0^{\mathrm{RL}_i}$ , and arrays  $\hat{D}^i$  and  $\hat{A}^i$ . end for t = 0, ..., T do for i = 1, ..., 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, ..., 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, ..., 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, ..., 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, ..., T do for i = 1, ..., N do  $\begin{array}{l} \text{Compute the feed-forward term } u_k^{\text{FF}_i}(x_t^i) = \\ u_k^{\text{RL}_i}(x_t^i) + \sum_{j=0}^{k-1} u_j^{\text{CBF}_i}(x_t^i, u_j^{FF_i}(x_t^i)) \\ \text{Use (32) solve for } u_k^{\text{CBF}_i}(x_t^i, u_k^{FF_i}) \end{array}$ Deploy controller  $u_k^i(x_t^i) = u_k^{\text{FF}_i}(x_t^i) + u_k^{\text{CBF}_i}(x_t^i, u_k^{\text{FF}_i}(x_t^i))$ Store state-action pairs  $(x_t^i, u_k^i(x_t^i))$ end for i = 1, ..., 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 + 1end

The following is the main result of the paper, which shows that the controller calculated using Algorithm 1 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  $u^{\text{DEC}_i}(x_t^i)$  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  $\mathcal{B}_{\delta}(\alpha) > 0$ ,  $\mathcal{B}_{\epsilon}(\beta) > 0$ , and  $R_u \triangleq \text{diag}(R_u^i) > 0$ . 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  $v_t$ . From (8), computing  $v_t$  requires information from its neighbours. Then, the proposed RL algorithm is distributed. However, in the event when the desired supply-function  $w_d$  is equal to  $w_v$ , then  $\tilde{w} = w_u$ . Consequently, the RL algorithm takes a decentralized form.

### IV. CASE STUDY: DC MICROGRID

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

$$\mathcal{C}_{i}^{n} = \left\{ (\overline{I}^{i}, \overline{V}^{i}, \overline{u}^{i}, \overline{\nu}^{i}) \in \mathbb{R}^{4} | R^{i} \overline{I}^{i} + \overline{V}^{i} - \overline{u}^{i} V_{s}^{i} = 0, \quad (34) \\ \overline{I}^{i} - G \overline{V}^{i} + \overline{\nu}^{i} = 0 \right\},$$

and

$$\mathcal{C}_{k}^{e} = \left\{ (\overline{I}_{l}^{i}, \overline{\mu}^{i}) \in \mathbb{R}^{2} | R_{l}^{i} \overline{I}_{l}^{i} + \overline{\mu}^{i} = 0 \right\},$$
(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 non-trivial operating point  $(\overline{I}^i, \overline{V}^i, \overline{u}^i, \overline{\nu}^i, \overline{I}^i_l, \overline{\mu}^i) \in C_i^n \times C_k^e$ , 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  $\nu$ , 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

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

and

$$w_e^k(\tilde{\mu}^k, \tilde{\omega}^k) = \tilde{\mu}^{k\top} \tilde{\omega}^k - R_l^k \|\tilde{\omega}^k\|_2^2,$$
(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 equation (22) in Assumption 4. Consequently, using (26) we compute the resulting control barrier function as

$$b^{i}(t, x_{t}^{i}) = -\sum_{\tau=t_{0}}^{t-1} \left( \tilde{u}^{i\top} \tilde{y}_{u}^{i} - G^{i} \| \tilde{y}_{\nu}^{i} \|_{2}^{2} \right), \ t \ge t_{0} \ge 0, \quad (38)$$

and its super-level is defined as in (28). Finally, we define the



Figure 4. (Top) time evolution of control barrier function, (bottom) voltage across the load of each DGU, considering a load variation of 5% at time t = 0.05 seconds.

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 1. Figure 5 compares the accumulated rewards of vanilla DDPG and the proposed dissipativity-ensuring Algorithm 1 using DDPG during training. As the plot shows, Algorithm 1 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 control barrier function at each node are plotted in Figure 4. At t = 0 seconds, 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 seconds. In Figure 4 we see that, after a minor perturbation, the voltage signals again stabilized to their desired values. Furthermore, the control barrier function is positive, thus validating the dissipativity-ensuring nature of the proposed approach.

## V. CONCLUSIONS

In this paper, 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 reinforceent 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  $a \in \mathbb{R}^m$ 

$$\begin{cases} x_{t+1} = f(x_t, a_t), \\ y_t = h(x_t, a_t). \end{cases}$$
(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  $\Sigma$  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 exist a non-negative function  $S : \mathbb{R}^n \to \mathbb{R}_+$ , called as storage function, satisfying S(0) = 0 such that for all  $s_{t_0} \in X$ , all  $t > t_0 \ge 0$  and all  $a_t \in A$ ,

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

or equivalently [46],

$$\sum_{i=t_0}^{t-1} w(a_t, y_t) \ge \sum_{i=t_0}^{t-1} \mathcal{D}(x_t) \ge 0$$
(42)

where  $\mathcal{D}(x_t) \in \mathbb{R}_+$  is a non-negative 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 (42) holds with

$$w(a_t, y_t) = -\|y_t\|_Q^2 + a_t^\top S y_t - \|a_t\|_R^2$$
(43)

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

$$a_t = 0 \text{ and } y_t = 0 \implies \lim_{t \to \infty} x_t \to 0.$$

## 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  $b(0, s_0) \ge 0$ . It suffices to show that  $b(t, s_t) \ge 0$ , for all  $a_t \in \text{DEC}(t, s_t)$ . From (24) and (25), for all  $a_t \in \text{DEC}(t, s_t)$ , we have

$$b(t+1, s_{t+1}) \ge (1-\eta)b(t, s_t).$$
(44)

Now, consider the following boundary value problem:

$$X_{t+1} = (1 - \eta)X_t \tag{45}$$



Figure 5. Comparison of accumulated rewards from nodes of DC microgrid for each episode during training using DDPG and the propose Dissipative CBF approach.

with initial condition  $X_0 = b(0, s_0) \ge 0$ . Then, the solution to (45) is  $X_t = (1 - \eta)^t X_0 \ge 0$ ,  $\forall k \in \mathbb{Z}^+$ ,  $0 < \eta \le 1$ . From (44) and (45),

$$b(t, s_t) \ge X_t. \tag{46}$$

Thus 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  $u_t \in D^i(x_t^i, \nu_t)$ , it implies that  $C^i$  is forward invariant. Consequently, we have  $b^i(t, x_t^i) = -\sum_{\tau=t_0}^{t-1} \tilde{w} \ge 0$ 

$$\implies \sum_{\tau=t_0}^{t-1} \tilde{w} \le 0 \tag{47}$$

$$\implies \sum_{\tau=t_0}^{t-1} (w_n - w_d) \le 0 \tag{48}$$

$$\implies \sum_{\tau=t_0}^{t-1} w_d \ge \sum_{\tau=t_0}^{t-1} w_n. \tag{49}$$

From Assumption 1 the subsystem (1) is dissipative, which further implies

$$\implies \sum_{\tau=t_0}^{t-1} w_d \ge \sum_{\tau=t_0}^{t-1} w_n \ge 0.$$
 (50)

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 exist a storage function  $S_d^i : \mathbb{R}^n \to \mathbb{R}_+$ , satisfying

$$S_d^i(x_t^i) \le S_d^i(x_{t_0}^i) + \sum_{t=t_0}^{t-1} w_d^i(\nu^i, y^i).$$
(51)

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 exist a storage function  $S_e^i : \mathbb{R}^m \to \mathbb{R}_+$ , satisfying

$$S_{e}^{k}(z_{t}^{k}) \leq S_{e}^{i}(z_{t_{0}}^{k}) + \sum_{t=t_{0}}^{t-1} w_{e}^{k}(\mu_{t}^{k},\omega_{t}^{k}).$$
(52)

Consider  $S(s_t) = \sum_{i=1}^{N} S_d^i(x_t^i) + \sum_{k=1}^{M} S_e^k(z_t^k)$ , consequently

$$S(s_{t}) - S(s_{t_{0}})$$

$$\leq \sum_{i=1}^{N} \sum_{t=t_{0}}^{t-1} w_{d}^{i}(\nu^{i}, y^{i}) + \sum_{k=1}^{M} \sum_{t=t_{0}}^{t-1} w_{e}^{k}(\mu_{t}^{k}, \omega_{t}^{k})$$

$$= \sum_{t=t_{0}}^{t-1} \sum_{i=1}^{N} w_{d}^{i}(\nu^{i}, y^{i}) + \sum_{t=t_{0}}^{t-1} \sum_{k=1}^{M} w_{e}^{k}(\mu_{t}^{k}, \omega_{t}^{k})$$

$$\leq \sum_{t=t_{0}}^{t-1} (\nu^{\top} S_{\nu}^{\top} y_{\nu} - \alpha \|\nu\|_{2}^{2} - \beta \|y\|_{2}^{2} + \mu^{\top} S_{e}^{\top} \omega - \delta_{e} \|\mu\|_{2}^{2}$$

$$-\epsilon_{e} \|\omega\|_{2}^{2})$$
(53b)
$$\leq \sum_{t=t_{0}}^{t-1} (\omega^{\top} \mathcal{B}^{\top} S_{\nu}^{\top} y_{\nu} - \alpha \|\mathcal{B}_{0}\|_{2}^{2} - \beta \|y\|_{2}^{2} - \beta \|y\|_{2}^{2} - \beta \|y\|_{2}^{2}$$

$$\leq \sum_{t=t_0} \left( \omega^\top \mathcal{B}^\top S_{\nu}^\top y_{\nu} - \alpha \|\mathcal{B}\omega\|_2^2 - \beta \|y_{\nu}\|_2^2 - \beta \|y_u\|_2^2 - \gamma_{\nu}^\top \mathcal{B}^\top S_e^\top \omega - \delta_e \|\mathcal{B}y_{\nu}\|_2^2 - \epsilon_e \|\omega\|_2^2 \right)$$
(53c)

$$= -\sum_{t=t_0}^{t-1} \left( \|\omega\|_{\mathcal{B}_{\delta}(\alpha)}^2 + \|y_{\nu}\|_{\mathcal{B}_{\epsilon}(\beta)}^2 + \beta \|y_u\|_2^2 \right)$$
(53d)

In (53a) we use (51) and (52). In (53b) we use the interconnection laws from (8). In (53c), we use Assumption 3. This implies that the overall networked system is stable.

Furthermore, consider  $\mathcal{B}_{\delta}(\alpha) > 0$ , and  $\mathcal{B}_{\epsilon}(\beta) > 0$ . Then from (53d) there exists a forward invariant set  $\Pi$  and by LaSalle's invariance principle, the solutions that start in  $\Pi$ converge to the largest invariant set contained in

$$\Pi \cap \left\{ s \in \mathbb{R}^{n+p} | \ \omega = 0, \ y = 0 \right\}.$$
(54)

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, the trajectories in  $\Pi$  converges asymptotically to the largest invariant set contained in

$$\Pi \cap \{s = 0\},\tag{55}$$

following [36, Corollary 4.2.2].

#### REFERENCES

 M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE transactions on robotics and automation*, vol. 17, no. 6, pp. 947– 951, 2001.

- [2] S. Sivaranjani, S. Sadraddini, V. Gupta, and C. Belta, "Distributed control policies for localization of large disturbances in urban traffic networks," in 2017 American Control Conference (ACC). IEEE, 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 Transactions on power electronics*, vol. 31, no. 7, pp. 4876–4891, 2015.
- [4] F. Horn and R. Jackson, "General mass action kinetics," Archive for rational mechanics and analysis, vol. 47, no. 2, pp. 81–116, 1972.
- [5] R. H. Lasseter and P. Paigi, "Microgrid: A conceptual solution," in 2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No. 04CH37551), vol. 6. IEEE, 2004, pp. 4285–4290.
- [6] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [7] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.
- [8] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.
- [9] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," arXiv preprint arXiv:1911.10635, 2019.
- [10] —, "Decentralized multi-agent reinforcement learning with networked agents: Recent advances," *arXiv preprint arXiv:1912.03821*, 2019.
- [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 *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3387–3395.
- [12] L. Buşoniu, T. de Bruin, D. Tolić, J. Kober, and I. Palunko, "Reinforcement learning for control: Performance, stability, and deep approximators," *Annual Reviews in 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 Systems Magazine*, vol. 32, no. 6, pp. 76–105, 2012.
- [14] F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe modelbased reinforcement learning with stability guarantees," in *Advances in neural information processing systems*, 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," *arXiv* preprint arXiv:1801.05039, 2018.
- [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," *arXiv preprint arXiv:1910.09496*, 2019.
- [17] J. C. Willems, "Dissipative dynamical systems part ii: Linear systems with quadratic supply rates," *Archive for rational mechanics and analysis*, vol. 45, no. 5, pp. 352–393, 1972.
- [18] c. J. Van der Schaft, L<sub>2</sub>-Gain and Passivity Techniques in Nonlinear Control. Springer, 2000, vol. 2.
- [19] C. A. Desoer and M. Vidyasagar, Feedback systems: input-output properties. SIAM, 2009.
- [20] G. Niemeyer and J. E. Slotine, "Stable adaptive teleoperation," *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 152–162, 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. Springer, 2016.
- [23] A. Van der Schaft and B. Maschke, "Port-hamiltonian systems on graphs," *SIAM Journal on Control and Optimization*, 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.
- [27] M. J. Tippett and J. Bao, "Dissipativity based distributed control synthesis," *Journal of Process Control*, vol. 23, no. 5, pp. 755–766, 2013.

- [25] 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*, 2020.
- [26] K. C. Kosaraju, M. Cucuzzella, J. M. A. Scherpen, and R. Pasumarthy, "Differentiation and passivity for control of brayton-moser systems," *IEEE Transactions on Automatic Control*, 2020.
- [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 2020 American Control Conference (ACC), 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 2019 American Control Conference (ACC). IEEE, 2019, pp. 5816– 5821.
- [30] B. Gao and L. Pavel, "On passivity, reinforcement learning and higherorder learning in multi-agent finite games," *IEEE Transactions on Automatic Control*, 2020.
- [31] S. P. Nageshrao, G. A. Lopes, D. Jeltsema, and R. Babuška, "Passivitybased 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 transactions on cybernetics*, vol. 45, no. 5, pp. 1017–1027, 2014.
- [33] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2016.
- [34] M. Z. Romdlony and B. Jayawardhana, "Uniting control lyapunov and control barrier functions," in 53rd IEEE Conference on Decision and Control. IEEE, 2014, pp. 2293–2298.
- [35] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007.
- [36] A. J. van der Schaft, L<sub>2</sub>-gain and passivity techniques in nonlinear control. Springer, London, 2000.
- [37] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in 2019 18th European Control Conference (ECC). IEEE, 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 Transactions on Control Systems Technology*, 2020.
- [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 2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS). IEEE, 2019, pp. 231–237.
- [41] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," in *Thirty-Second* AAAI Conference on Artificial Intelligence, 2018.
- [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 Transactions on Automatic Control*, vol. 64, no. 7, pp. 2737–2752, 2018.
- [43] M. Cucuzzella, K. C. Kosaraju, and J. Scherpen, "Voltage control of dc networks: robustness for unknown zip-loads," arXiv preprint arXiv:1907.09973, 2019.
- [44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.
- [45] E. Navarro-López, D. Cortés, and E. Fossas-Colet, "Implications of dissipativity and passivity in the discrete-time setting," *IFAC Proceedings Volumes*, vol. 35, no. 1, pp. 55–60, 2002.
- [46] M. Xia, P. J. Antsaklis, V. Gupta, and M. J. McCourt, "Determining passivity using linearization for systems with feedthrough terms," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2536–2541, 2014.