Data-driven Sparsity-promoting Optimal Control of Power Buffers in DC Microgrids

Paolo R. Massenio, David Naso, *Senior Member, IEEE*, Frank L. Lewis, *Life Fellow, IEEE*, and Ali Davoudi, *Senior Member, IEEE*

***Abstract*—A power buffer is a power electronics converter with a large capacitor that shields a weak DC grid from abrupt load changes. Distributed control solutions have been shown to be superior to the decentralized ones, however, the effects of the communication network topology on the control performance of these buffers have not yet been studied. This paper offers a data-driven optimal solution to reduce the interactions between different control loops of power buffers while minimizing a closed-loop performance function. Reinforcement learning methods deal with the optimal control of nonlinear systems, and a Tabu Search method addresses the resulting combinatorial problem. The proposed solutions are validated for a DC microgrid in a controller/hardware-in-the-loop environment.**

***Index Terms*—DC microgrid, Nonlinear optimal control, Power buffer, Reinforcement learning, Sparsity promoting.**

I. Introduction

DC microgrids are shown to be more efficient and reliable than their AC counterparts [1]. The DC distribution grid is afflicted with low damping, lack of generational inertia, and the presence of abrupt loads [2], [3]. A power buffer is an electronics converter that uses the energy stored in a large capacitor to cushion the effects of abrupt load transients [4]–[7]. High-performance control of such buffers, that tunes their stored energy and input impedance, is challenging.

Introducing a communication network among power buffers allows a collective response to load changes [7]–[10]. Distributed solutions in [7], [9] are found with policy iteration and linear distributed designs, respectively. A reinforcement learning (RL) approach in [10] overcomes linear approximations. The communication topology in [7]–[10] is inspired by physical vicinity. When the distribution grid is considered [7], [9], [10], the underlying physical interconnection reflects the fixed communication topology with no guarantees that these structures (physical and communication) are optimal with regard to control objectives. Given the limited energy available, co-optimization of control solutions and communication topologies, considering the distribution grid, is important.

Sparsity-promoting algorithms guarantee stability and performance without any *a priori* defined communication topology, i.e., few but crucial communication links are found [11], [12]. Similar to AC systems [13], [14], power buffers can benefit from reducing the interactions among feedback loops, minimizing communication costs with a limited impact on the closed-loop performance. Minimizing computational costs is appealing for battery-constrained Internet-of-Things devices [15]. Existing sparsity-promoting methods for microgrids mostly rely on linear approaches in AC systems. Based on the linear formulation in [11], decentralized controllers for AC networks with voltage-source converters are designed in [16], while sparse and block-sparse wide-control architectures for AC systems are designed in [13] and [17], respectively. By extending [11] to discrete-time systems, the sparsity-promoting controller in [18] regulates the active power flows and frequency. In [14], decentralized and sparse wide-area controllers are designed to damp inter-area oscillations in AC systems using the convex relaxation of a linear H_∞ problem. Constrained Linear Quadratic Regulator (LQR) formulation finds an optimal controller for predefined communication structures to damp inter-area oscillators in [19]. In [20], a sparsity-promoting linear optimal controller is applied to an AC power system with synchronous machines. However, such formulations are not practical for nonlinear systems as in the case of DC microgrids with power buffers.

This paper proposes a sparsity-promoting optimal design for nonlinear systems based on off-policy RL techniques. In general, nonlinear optimal control problems are solved using the Hamilton-Jacobi-Bellman (HJB) equation or the Pontryagin's Minimum Principle (PMP) [21]. The PMP method is easier to tackle and provides an open-loop controller with only the necessary condition for optimality. The HJB equation is generally intractable but provides a closed-loop control policy with both necessary and sufficient optimality conditions [22]. In this paper, the closed-loop optimal controller is found by approximating the solution of the HJB equation using a RL-based method, namely the Integral Reinforcement Learning (IRL) approach with off-policy learning [23], [24]. In particular, Neural Networks (NNs) provide approximations for the optimal control policy and value function [25]–[27]. The approximated solution of the HJB equation is learned using only system collected data and without the need for the exact knowledge of the system dynamics. Such approach is commonly categorized as *data-driven* [28]–[30]. The same set of collected data is repetitively used to find optimal controllers for different communication topologies.

Stability of optimal designs with sparsity-promoting or structural constraint is not guaranteed even for linear systems [13]. This work employs Domain-of-Attraction (DoA)

F. L. Lewis and A. Davoudi work was supported, in part, by the National Science Foundation under Grant ECCS-1839804.

P. R. Massenio was a visiting scholar at the University of Texas at Arlington. P. R. Massenio and D. Naso are with the Polytechnic University of Bari, Bari, 70126, Italy (e-mail: paoloroberto.massenio@poliba.it; david.naso@poliba.it).

F. L. Lewis and A. Davoudi are with the University of Texas at Arlington Research Institute, Fort Worth, TX, 76118, USA (e-mail: lewis@uta.edu; davoudi@uta.edu).

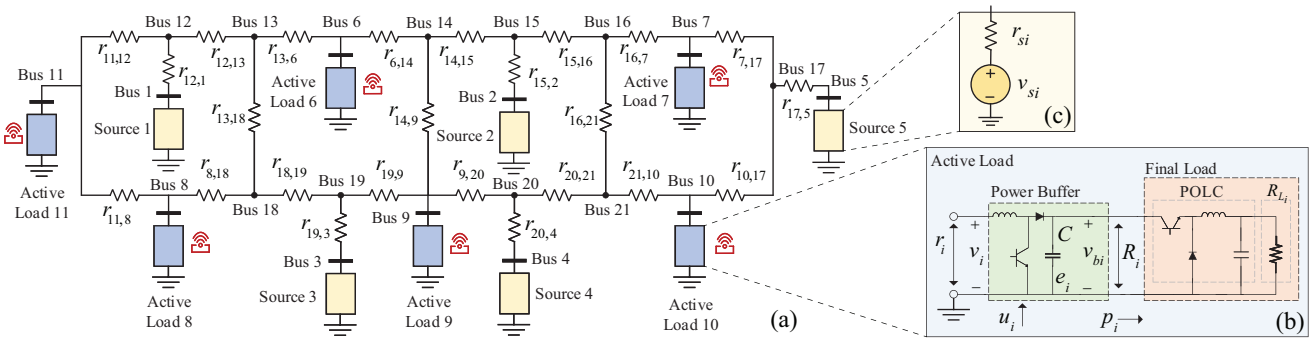

Fig. 1. DC microgrid and its elements: (a) DC microgrid, (b) active load consisting of a power buffer and a final load, and (c) model of a DC source.

estimation methods [31]–[33] to check the stability of each distributed controller. To deal with the resulting combinatorial problem, a Tabu Search (TS) approach that avoids local minima is used [34], [35]. The main contributions of this paper are

- The first attempt to solve nonlinear sparsity-promoting and structured optimal control problems using a data-driven algorithm based on RL and TS methods is presented. It can handle arbitrary positive-definite utility functions, not just simple linear quadratic functions.
- These controllers are employed in DC microgrids, with a limited impact when comparing incrementally-sparse and fully-connected communication topologies. This impact is shown to increase if existing techniques used for AC systems, e.g., [13], are applied.
- In contrast to [7], [9], [10], this paper considers the underling physical interconnection structure dictated by the distribution grid. The communication topology is considered as a free parameter subject to optimization, where the number of active communication links and a closed-loop cost function are simultaneously minimized.
- Sparsity-promoting and communication topology-related analytics for power buffers are provided. The reciprocal assistance among power buffers is shown to increase with a less sparse communication structure.

The rest of this paper is organized as follows. The distribution grid and power buffers are detailed in Section II. Section III provides the proposed data-driven sparsity-promoting algorithm. Numerical and Controller/Hardware-in-the-Loop (CHIL) studies are conducted for a DC microgrid in Section IV. Conclusion is given in Section V.

II. Nonlinear Model of a DC Microgrid

Distribution lines, *active loads*, and DC sources constitute the DC microgrid, as depicted in Fig. 1(a). $r_{i,j}$ denotes the resistance between buses i and j. A power buffer connected with a *final load*, i.e., a point-of-load converter (POLC) and a resistive load (Fig. 1(b)), constitutes an active load [9]. A resistor, r_{si}, in series with a voltage source, v_{si} (Fig. 1(c)), model a DC source. Let the number of active loads and sources be N and M, respectively, and the set of active loads be $\mathcal{L} = \{M+1, ..., M+N\}$. For the i^{th} active load, r_i, v_i, e_i, and p_i are the input impedance, input voltage, stored energy, and power supplied to the final load, respectively. Thus,

$$\dot{e}_i = \frac{v_i^2}{r_i} - p_i, \quad i \in \mathcal{L}. \tag{1}$$

The stored energy is approximated [10] as

$$e_i = \frac{1}{2} C v_{bi}^2, \quad i \in \mathcal{L}. \tag{2}$$

C is the buffer's capacitance, and v_{bi} is its output voltage. State-space model of each active load becomes [10]

$$\begin{cases} \dot{e}_i = \frac{\zeta_i(r_{M+1},...,r_i,...,r_{M+N})^2}{r_i} - \frac{2e_i}{C}\frac{1}{R_i} \\ \dot{r}_i = u_i, \end{cases} \quad i \in \mathcal{L}. \tag{3}$$

$\zeta_i(r_{M+1}, ..., r_i, ..., r_{M+N})$ relates the active load's input voltage, v_i, and input impedances of all buffers. $u_i \in \mathbb{R}$ denotes the control input tuning the buffer's input impedance, r_i. R_i is the buffer's output resistance that relates to the load resistance, R_{L_i}. Once a set of desired output resistances, R_i^*, is given, corresponding steady-state energy, input resistance, and control input are e_i^*, r_i^*, and $u_i^* = 0$, respectively. r_i^* is found by solving (3) at the steady state. Let's consider a second-order approximation of (3), for each $i \in \mathcal{L}$, around an equilibrium point, i.e.,

$$\begin{cases} \dot{x}_{i_1} = \sum_{j=M+1}^{M+N} \left(\frac{\partial}{\partial r_j} \frac{\zeta_i(r)^2}{r_i}\Big|_{r^*} x_{j_2} + \frac{1}{2}\frac{\partial^2}{\partial r_j^2}\frac{\zeta_i(r)^2}{r_i}\Big|_{r^*} x_{j_2}^2 \right) \\ \qquad -\frac{2}{CR_i^*} x_{i_1} \\ \dot{x}_{i_2} = u_i, \end{cases} \tag{4}$$

$r = [r_{M+1} \cdots r_{M+N}]^\intercal$, $x_{i_1} = e_i - e_i^*$, and $x_{i_2} = r_i - r_i^*$.

Note that both (3) and (4) are nonlinear systems. As shown in [7] and [9], first-order linearization around the half-load loading scenario provides a satisfactory performance. Better performances are obtained with nonlinear switching control policies based on target load scenarios [10]. The second-order approximation in (4) provides a good trade-off between those two approaches, as will be shown in Section IV C.

Each active load in (4) can be generally expressed as

$$\dot{x}_i = f_i(x) + g_i(x)u_i, \quad i \in \mathcal{L}. \tag{5}$$

$x_i \in \mathbb{R}^{n_i}$ is the state vector of active load i. While general expressions for n_i and $g_i(x)$ are used, for each $i \in \mathcal{L}$ it

results that $n_i = 2$, $x_i = [x_{i_1} \quad x_{i_2}]^\intercal$, and $g_i(x) = [0 \quad 1]^\intercal$. The overall system's state is $x = \left[x_{M+1}^\intercal, ..., x_{M+N}^\intercal\right]^\intercal \in \mathbb{R}^{\bar{N}}$, where $\bar{N} = \sum_{i=M+1}^{M+N} n_i$. Each function $f_i(x) : \mathbb{R}^{\bar{N}} \to \mathbb{R}^{n_i}$ is locally Lipschitz with $f_i(0) = 0$. The interconnection of the N subsystems in (5) gives the overall microgrid dynamics,

$$\dot{x} = f(x) + g(x)u. \tag{6}$$

$f(x) = [f_{M+1}(x)^\intercal, ..., f_{M+N}(x)^\intercal]^\intercal \in \mathbb{R}^{\bar{N}}$, $g(x) = \mathbf{diag}(g_{M+1}(x), ..., g_{M+N}(x)) \in \mathbb{R}^{\bar{N}\times N}$, and $u = [u_{M+1}, ..., u_{M+N}] \in \mathbb{R}^N$. Let's assume (6) partially unknown, i.e., $f(x)$ is unknown and $g(x)$ is known.

The goal is to minimize, at the same time, the number of the communication links (sparsity-promoting objective) and a closed-loop performance index (optimal control objective). To define the overall objective function, the first step is to solve the optimal control problem, whose cost function is

$$J(x, u) = \int_0^\infty U(x, u)dt, \tag{7}$$

where $U(x, u)$ is the utility function defined as

$$U(x, u) = Q(x) + \sum_{i\in\mathcal{L}} \rho_i(x)u_i^2. \tag{8}$$

$Q(x)$ is a positive definite function weighting the convergence dynamics. While any nonlinear function can be employed, usually $Q(x)$ provides a weighted sum of both single states and states products, as in Section IV. Higher weighting terms imply faster converge rates for corresponding states. $\rho_i(x)$, $i \in \mathcal{L}$, are positive definite functions weighting the control effort of each power buffer during transients. Similar to $Q(x)$, while general nonlinear expressions of states and inputs can be employed, scalar weights are used as $\rho_i(x)$. Nonlinear expressions arise when other specifications are included in the utility function, e.g., losses in distribution lines. The optimal sparsity-promoting objective function and an algorithmic procedure that optimizes it are provided in the next Section.

III. Proposed Optimal Sparsity-promoting Algorithm for Nonlinear Systems

First, a RL method approximates the structured optimal control of nonlinear systems, i.e., a set of optimal controllers for a fixed communication topology. In Section III B, the stability of structured nonlinear policies is evaluated using DoA estimation methods. This estimation is used in the objective function of the sparsity-promoting problem in Section IV C. Finally, TS handles the resulting combinatorial problem.

A. Off-policy Integral Reinforcement Learning

The optimal feedback, that minimizes (7) and drives (6) to zero [25], is

$$u^*(x) = -0.5R_\rho^{-1}(x)g^\intercal(x)\nabla V^*(x), \tag{9}$$

where $R_\rho(x) = \mathbf{diag}(\rho_{M+1}(x), ..., \rho_{M+N}(x))$. $V^*(x) = \min_u J(x, u)$ is the optimal value function found by solving the following HJB equation

$$\min_u H(x, u^*, V^*) = 0, \tag{10}$$

where H is the optimal control problem's Hamiltonian,

$$H(x, u, V) \triangleq U(x, u) + \nabla V^{*\intercal}(x)(f(x) + g(x)u). \tag{11}$$

Note that $V^*(x_0)$ provides the optimal cost when $x(0) = x_0$.

Algorithm 1 can solve the generally analytically-intractable HJB equation. Step 2a of Algorithm 1 is still a challenging task. An IRL policy iteration algorithm, featuring off-policy learning, is used [36]. For any asymptotically-stable control policy $u^{(0)}$, and for any bounded noise injected for learning and exploration purposes, $e_n(t) : \mathbb{R} \to \mathbb{R}^N$, one has

$$\begin{aligned}\dot{x} &= f(x) + g(x)(u^{(0)}(x) + e_n(t)) = \\ &= f(x) + g(x)(u^{(k)}(x) + u^{(k)'}(x)), \quad \forall k \geqslant 0,\end{aligned} \tag{12}$$

where $u^{(k)'} = u^{(0)} - u^{(k)} + e_n$. The derivative w.r.t the time of $V^{(k)}(x)$, along the state trajectory of (12), is

$$\dot{V}^{(k)}(x) = -U(x, u^{(k)}) - 2\sum_{i\in\mathcal{L}} u_i^{(k+1)}\rho_i(x)u_i^{(k)'}. \tag{13}$$

The value function, $V^{(k)}$, and the policies, $u_i^{(k+1)}$, are approximated using linear-in-parameters (LIP) approximators [37],

$$\hat{V}^{(k)}(x) = \sum_{l=1}^{N_V} \omega_l^{(k)}\gamma_l(x) = \omega^{(k)\intercal}\Gamma(x), \tag{14}$$

$$\hat{u}_i^{(k+1)}(x) = \sum_{l=1}^{N_U} \theta_{i_l}^{(k)}\xi_l(x_l^\xi) = \theta_i^{(k)\intercal}\Xi(x), \tag{15}$$

where $\gamma_l(x)$, with $l = 1, ..., N_V$, and $\xi_l(x_l^\xi)$, with $l = 1, ..., N_U$, are set of smooth linearly-independent functions returning zero at the origin, with N_V and N_U as integers. The l^{th} basis function $\xi_l(x_l^\xi)$ depends on a subset of the overall system state, e.g., if $\xi_{\bar{l}}(x_{\bar{l}}^\xi) = \xi_{\bar{l}}(x_{M+1}, x_{M+2}, x_{M+4})$, then $x_{\bar{l}}^\xi = \{x_{M+1}, x_{M+2}, x_{M+4}\}$. For each $\xi_l(x_l^\xi) \in \Xi(x)$, the set $N_l^\xi = \{j | x_j \in x_l^\xi\}$ is defined. The basis functions set $\Xi(x)$ is the same for each buffer. $\omega^{(k)} \in \mathbb{R}^{N_V}$ and $\theta_i^{(k)} \in \mathbb{R}^{N_U}$, $i \in \mathcal{L}$, are constant weights to be determined. Integrating (13) over any time interval, and replacing $V^{(k)}$ and $u_i^{(k+1)}$ with their approximations, leads to

$$\begin{aligned}&\omega^{(k)\intercal}\underbrace{[\Gamma(x(t_{n+1})) - \Gamma(x(t_n))]}_{\Delta\Gamma(t_{n+1})\in\mathbb{R}^{N_V}} = -\underbrace{\int_{t_n}^{t_{n+1}} Q(x)dt}_{Q_I(t_{n+1})\in\mathbb{R}} \\ &- \sum_{i\in\mathcal{L}} \theta_i^{(k-1)\intercal}\left(\int_{t_n}^{t_{n+1}} \Xi(x)\rho_i(x)\Xi^\intercal(x)dt\right)\theta_i^{(k-1)} \\ &- 2\sum_{i\in\mathcal{L}} \theta_i^{(k)\intercal}\underbrace{\int_{t_n}^{t_{n+1}} \Xi(x)\rho_i(x)(u_i^{(0)} + e_{n_i})dt}_{\Psi_i(t_{n+1})\in\mathbb{R}^{N_U}} \\ &+ 2\sum_{i\in\mathcal{L}} \theta_i^{(k)\intercal}\underbrace{\left(\int_{t_n}^{t_{n+1}} \Xi(x)\rho_i(x)\Xi^\intercal(x)dt\right)}_{\Phi_i(t_{n+1})\in\mathbb{R}^{N_U\times N_U}}\theta_i^{(k-1)} + \epsilon_{k_n}.\end{aligned} \tag{16}$$

To minimize the communication links, a sparse control law that keeps the system stable and minimizes (7) needs to be found. Let's define a binary decision matrix, $A_d \in \mathbb{R}^{N\times N}$, such that $(A_d)_{ij} = 1$ if subsystem j is allowed to send its

Algorithm 1 Policy Iteration Algorithm

1. **Initialization:** Set $k = 0$, and $u^{(0)}(x)$ as the initial stable controller for the overall system.
2. **Iteration:** Repeat until convergence
   a. **Policy Evaluation:** Find $V^{(k)}(x)$, with $V^{(k)}(0) = 0$, from $H(x, u^{(k)}, V^{(k)}) = 0$.
   b. **Policy Improvement:** Update $u^{(k+1)}(x)$ using (9) and $V^{(k)}(x)$.

own state to subsystem i, otherwise, $(A_d)_{ij} = 0$. Given a fixed A_d, the matrix $P_i(A_d) \in \mathbb{R}^{N_U \times N_U}$, for each $i \in \mathcal{L}$, is defined as

$$P_i(A_d) = \mathbf{diag}\left(\prod_{j \in N_1^{\xi}} (A_d)_{ij} \quad \dots \quad \prod_{j \in N_{N_U}^{\xi}} (A_d)_{ij} \right). \quad (17)$$

Therefore, for $(A_d)_{ij} = 0$, the l^{th} diagonal element of $P_i(A_d)$ is zero if the l^{th} approximating function $\xi_l(x_l^{\xi})$ depends on x_j. Given an arbitrary A_d, Algorithm 2 makes use of the same data collected for the fully-connected communication topology to find, if there exist, approximated optimal control policies in line with the communication topology defined by A_d. An off-policy learning is implemented since the iterative stage starts after collecting the learning data.

The main advantages introduced by using the IRL approach can be summarized as follows. An approximated optimal feedback controller, that does not require the explicit solution of the HJB equation, is obtained. Using collected system data, the full knowledge of the system dynamics is not required. Finally, the same collected data can be repeatedly used to find the approximated optimal control policies for different communication topologies, hence significantly reducing the computational requirements.

B. Domain-of-Attraction Estimation

The stability of the approximated optimal policies depends on the given structure of A_d, as well as on a compact set $\Omega_L \subset \mathbb{R}^{\bar{N}}$ where the data collecting phase has been done. NNs approximate nonlinear functions on compact sets, and not on the entire $\mathbb{R}^{\bar{N}}$ [36]. The stability is verified by quantifying the DoA of the origin in the resulting closed-loop system, i.e.,

$$\mathcal{H} = \{x_0 \in \mathbb{R}^{\bar{N}} | \lim_{t \to \infty} x(t, x_0) = 0\}. \quad (18)$$

Once approximated policies are obtained, the function $\hat{V}_{A_d}(x) = \hat{\omega}_{A_d}^{\intercal} \Gamma(x)$ is employed as a candidate Lyapunov function, whose sub-level set is defined, for any $l \in \mathbb{R}$, as

$$\mathcal{H}_{\hat{V}}(l) = \{x \in \mathbb{R}^{\bar{N}} | \hat{V}_{A_d}(x) \leqslant l\}. \quad (19)$$

Given the difficulty in finding the DoA in a closed form, an estimation is found using data-driven methods. Any sublevel set provides an estimation if $\hat{V}_{A_d}(x)$ is positive definite and $\dot{\hat{V}}_{A_d}(x)$ is negative definite within the sub-level set [31]. The goal is to find the largest set, $\mathcal{H}_{\hat{V}}(l^*)$, representing the largest estimate for the DoA. This paper adopts the memory-based algorithm in [31] and modified in Algorithm 3.

Algorithm 2 Off-Policy IRL Algorithm for an Arbitrary A_d

Inputs: Initial weights $\omega^{(0)}$, $\theta_i^{(0)}$, recorded data $\Delta\Gamma(t_n)$, $\Psi_i(t_n)$, $\Phi_i(t_n)$, $Q_I(t_n)$, matrices $P_i(A_d)$, with $i \in \mathcal{L}$ and $n = 1, ..., N_L$; A stopping threshold δ.
Outputs: Near-optimal cost function and policies $\hat{\omega}_{A_d}$ and $\hat{\theta}_{i_{A_d}}$, with $i \in \mathcal{L}$.

1. **Initialization:** Set $k = 1$; Evaluate $X_\Gamma = [\Delta\Gamma^{\intercal}(t_1) \ ... \ \Delta\Gamma^{\intercal}(t_{N_L})]^{\intercal} \in \mathbb{R}^{N_L \times N_V}$, and $B_Q = -[Q_I(t_1) \ ... \ Q_I(t_{N_L})]^{\intercal} \in \mathbb{R}^{N_L}$.
2. **Data Evaluation:** Compute the following matrices
$$X_i = [2\left(\Psi_i^{\intercal}(t_1) - \theta_i^{(k-1)^{\intercal}} \Phi_i^{\intercal}(t_1)\right) P_i(A_d) \ ... \\ 2\left(\Psi_i^{\intercal}(t_{N_L}) - \theta_i^{(k-1)^{\intercal}} \Phi_i^{\intercal}(t_{N_L})\right) P_i(A_d)]$$
$$B_\Phi = -[\sum_{i \in \mathcal{L}} \theta_i^{(k-1)^{\intercal}} \Phi_i(t_1) \theta_i^{(k-1)} \ \dots \\ \sum_{i \in \mathcal{L}} \theta_i^{(k-1)^{\intercal}} \Phi_i(t_{N_L}) \theta_i^{(k-1)}].$$
3. **Policy Improvement:** Find $\omega^{(k)}$ and $\theta_i^{(k)}$, $i \in \mathcal{L}$ from the following least square problem
$$[X_\Gamma \ X_{M+1} \ ... \ X_{M+N}] \begin{bmatrix} \omega^{(k)^{\intercal}} & \theta_{M+1}^{(k)^{\intercal}} & \dots & \theta_{M+N}^{(k)^{\intercal}} \end{bmatrix}^{\intercal} \\ = B_Q + B_\Phi.$$
4. **Off-policy Iteration:** If $||\omega^{(k)} - \omega^{(k-1)}|| \geqslant \delta$, then set $k = k + 1$ and repeat Step 2. Otherwise, stop and return $\hat{\omega}_{A_d} = \omega^{(k)}$, $\hat{\theta}_{i_{A_d}} = \theta_i^{(k)}$, with $i \in \mathcal{L}$.

The candidate Lyapunov function and its derivative are evaluated on randomly-selected data during the learning phase.

Let $T_{Rs} = \{t_{R_i}, i = 1, ..., N_{Rs}\}$ be the set of randomly-selected sampling times. The following sets of sampled data are collected during the learning phase: $S_x = \{x(t_{R_i}), i = 1, ..., N_{Rs}\}$, $S_{dx} = \{\dot{x}(t_{R_i}), i = 1, ..., N_{Rs}\}$, and $S_{ue} = \{s_{ue}(t_{R_i}), i = 1, ..., N_{Rs}\}$, where $s_{ue}(t_{R_i}) = \left(u^{(0)}(x(t_{R_i})) + e_n(t_{R_i})\right)$. For each sampled state, $x(t_{R_k})$, and any weights set, ω_{A_d}, $\theta_{i_{A_d}}$, the following holds

$$\dot{\hat{V}}_{A_d}(x(t_{R_k})) = \omega_{A_d}^{\intercal} \nabla\Gamma(x(t_{R_k}))\Big[\hat{f}(x(t_{R_k})) + \\ g(x(t_{R_k}))\begin{bmatrix}\theta_{M+1_{A_d}}^{\intercal} \cdots \theta_{M+N_{A_d}}^{\intercal}\end{bmatrix}^{\intercal} \Xi(x(t_{R_k}))\Big], \quad (20)$$

where $\hat{f}(x(t_{R_k})) = \dot{x}(t_{R_k}) - g(x(t_{R_k}))s_{ue}(t_{R_k})$ is the estimated value of $f(x(t_{R_k}))$. Algorithm 3 requires the knowledge of $g(x)$ to compute (20) and $\hat{f}(x(t_{R_k}))$. The number of failed trials among the sampled data is N_F. Algorithm 3 updates the upper and lower bounds of l^*, i.e., d_U and d_L, respectively. For each sampled state, $x(t_{R_k})$, the potential estimate for the DoA, i.e., $\hat{V}_{A_d}(x(t_{R_k}))$, is stored in the memory M_E if stability conditions are verified. Then, the current estimated DoA increases or decreases its radius according to the conditions in steps 7 and 9 [31], respectively.

Algorithm 3 returns the parameters d_L, η_V, η_F, and $\bar{V}$. d_L provides a conservative estimation of the DoA, while η_V is the ratio between d_L and the maximum evaluated cost function.

Algorithm 3 DoA Estimation Algorithm modified from [31]

Inputs: Approximated optimal cost function and control policies, $\hat{V}_{A_d}(x) = \hat{\omega}_{A_d}^{\intercal}\Gamma(x)$, $\hat{u}_{i_{A_d}}(x) = \hat{\theta}_{i_{A_d}}^{\intercal}\Xi(x)$, $i \in \mathcal{L}$; Sampled data S_{ue}, S_x, and S_{dx}; Function $g(x)$.
Outputs: DoA estimation d_L; Maximum/estimated ratio η_V; Failed ratio η_F; Average cost $\bar{V}$.

1. **Initialization:** Set $d_L = 0$, $d_U = \infty$, $N_F = 0$, $M_E = \{0\}$.
2. **for** $k = 1, ..., N_{Rs}$ **do**
3. Compute $\tau_k = \hat{V}_{A_d}(x(t_{R_k}))$, and $\dot{\tau}_k = \dot{\hat{V}}_{A_d}(x(t_{R_k}))$ as in (20).
4. **if** $\dot{\tau}_k < 0$ **and** $\tau_k \geqslant 0$ **then**
5. store $\hat{V}_{A_d}(x(t_{R_k}))$ in M_E; **else** $N_F = N_F + 1$
6. **end if**
7. **if** $\dot{\tau}_k < 0$ **and** $0 \leqslant d_L < \tau_k < d_U$ **then**
8. $d_L = \hat{V}_{A_d}(x(t_{R_k}))$
9. **else if** $\dot{\tau}_k \geqslant 0$ **and** $0 \leqslant \tau_k < d_U$ **then**
10. $d_U = \hat{V}_{A_d}(x(t_{R_k}))$
11. **if** $d_L \geqslant d_U$ **then** $d_L = argmax\{e \in M_E | e < d_U\}$
12. **end if**
13. **end for**
14. Compute V_{max} and $\bar{V}$ as the maximum and the average value of M_E, respectively.
15. **return** d_L, $\eta_V = d_L/V_{max}$, $\eta_F = N_F/N_{Rs}$, and $\bar{V}$.

It provides a measure of how small the resulting DoA is compared to the state space spanned during the training phase; E.g., if $\eta_V = 1$, then the DoA is the whole training space. η_F is the ratio between the failed and total trials. Finally, the average cost, $\bar{V}$, provides a performance measure of the resulting controllers in terms of (7).

C. Sparsity Promoting and Tabu Search

The sparsity-promoting problem can now be defined as

$$\underset{A_d}{\text{minimize}} \quad \beta||A_c \circ A_d||_F^2 + \alpha(A_d) \tag{21}$$

where $A_c \in \mathbb{R}^{N\times N}$, $(A_c)_{ij} > 0$ is the cost of the communication link between buffers i and j, $\circ$ denotes the Hadamard product, $||\cdot||_F$ is the Frobenius norm, β is a weighting factor, and $\alpha(A_d)$ is defined in Algorithm 4. Due to possible numerical errors in (20), the DoA obtained by Algorithm 3 may be still valid if η_F is below a given threshold, δ_F, which is a design parameter. If $\eta_F > \delta_F$ or Algorithm 2 does not converge, the control policy is considered unstable, with the penalty set to ∞. Otherwise, $\alpha(A_d)$ provides a penalty term proportional to the average performance, $\bar{V}$, and to the reduction of the DoA regarding its maximum span, i.e., $\alpha(A_d) = \bar{V}/\eta_V$. Thus, the compromise between the resulting averaged performances and the number of active communication links is minimized. Note that $\alpha(A_d)$ is computed for every A_d using the data collected for the fully-connected structure.

The TS algorithm reported in Algorithm 5 solves the combinatorial optimization problem in (21). TS uses a flexible search history to avoid local minimum entrapment [34], [35]. The main features of TS are the moves and the tabu list. Each move m in the moves set, $\mathcal{M}$, generates a new solution when applied to the current one. Herein, swap, reversion, and insertion moves are implemented. In summary, the best solution is initialized with a fully-connected feedback. Each TS iteration seeks the best non-tabu move that improves the current best solution. Then, the best move is inserted in the tabu list whose length provides the number of TS iterations in which the move is forbidden, allowing better exploration and escaping the local minimum. Finally, the relationships between the algorithmic components of the proposed approach are graphically represented in Fig. 2.

Algorithm 4 $\alpha(A_d)$ Function

Inputs: Matrix A_d; Threshold parameter δ_F.
Outputs: Cost $\alpha(A_d)$.

1. **Off-Policy IRL Convergence Check:** Run Algorithm 2 and, if converges, obtain approximated optimal weights and go to Step 2; Otherwise, return $\alpha(A_d) = \infty$.
2. **DoA Estimation:** Run Algorithm 3 and obtain η_F, η_V, and $\bar{V}$ parameters. If $\eta_F < \delta_F$, go to Step 3; Otherwise, return $\alpha(A_d) = \infty$.
3. **Cost Evaluation:** Return $\alpha(A_d) = \bar{V}/\eta_V$.

Algorithm 5 Tabu Search Algorithm

Inputs: Initial solution S_0; Tabu length T_L; Set of moves $\mathcal{M}$.
Outputs: Best solution S_B^*.

1. **Initialization:** For every move $m \in \mathcal{M}$, initialize the corresponding tabu counter, $T_C(m)$, to zero; Set the initial best solution $S_B^* = S_0$; Set the best candidate solution $S_B = S_0$.
2. **Best candidate solution evaluation:**
   a. **for each** $m \in \mathcal{M}$ **do**
   b. **if** $T_C(m) = 0$ **then**
   c. Apply move m to S_B and obtain solution $S_{B,m}$
   d. **if** $S_{B,m}$ is better than S_B **then** Set $S_{B,m} = S_B$ and set the best move, m_B, to m.
   e. **end if**
   f. **end for**
3. **Best solution evaluation:** **if** S_B is better than S_B^* **then** update $S_B^* = S_B$.
4. **Tabu list update:** Add m_B to the tabu list by setting $T_C(m_B) = T_L$. For each $m \in \mathcal{M}$, $m \neq m_B$, decrease $T_C(m)$ by 1 if greater than 0.
5. **Stopping criterion:** Go to Step 2 until the maximum number of iteration is reached.

IV. Case Studies

A. DC Microgrid Setup

Verification studies are conducted on the 48V DC microgrid depicted in Fig. 1(a), where $M = 5$ and $N = 6$, with $v_{si} = 50V$ and $r_{si} = 0.1\Omega$. Line resistances are $r_{18,19} = 0.2\Omega$, $r_{13,6} = r_{6,14} = r_{19,3} = r_{20,21} = 0.3\Omega$, $r_{12,1} = r_{11,8} = r_{15,2} = r_{7,17} = 0.4\Omega$, $r_{19,9} = r_{14,9} = r_{10,17} = 0.7\Omega$, $r_{11,12} = r_{14,15} = r_{20,4} = r_{15,16} = r_{16,7} = r_{17,5} = 0.5\Omega$, $r_{12,13} = r_{8,18} = r_{9,20} = r_{16,21} = r_{21,10} = 0.6\Omega$, and $r_{13_{18}} = 0.9\Omega$. Each active load uses a boost converter as a power buffer, with $C = 4.4mF$, and a buck converter

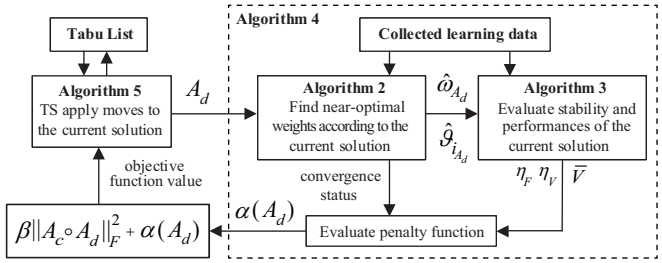

Fig. 2. Information flow between algorithmic components in the proposed sparsity-promoting approach.

as a POLC, with an LC filter placed in between them. Communication network and control schemes are emulated on a dSpace MicroLabBox system, the physical microgrid is emulated on a Typhoon HIL604 hardware. The two platforms physically separate the control loops from the real microgrid, as it happens in real-world implementations. The CHIL setup allows a realistic evaluation of the proposed approach, providing a better fidelity if compared with computer simulations [38]. In fact, simulation-only experiments do not consider a real-time implementation, which is an important aspect when testing distributed controllers [39]. Communication and controller sampling times are $1ms$ and $0.1ms$, respectively.

The control objectives are two fold: 1) Regulate the output voltage of each buffer in the steady state at $v_{bi}^* = 100V$, with a corresponding $e_i^* = 22J$; 2) Vary the input impedance, r_i, according to the sparse distributed policy. Both objectives are addressed using the fast voltage tracker of each boost converter. The resulting scheme is shown in Fig. 3. The i^{th} active load receives the states $\{x_j\}_{j\in N_i}$, where N_i denotes the set of other buffers that communicate with the buffer i, i.e., $N_i = \{j|(A_d)_{ij} = 1\}$. Note that in Fig. 3, $x = \{x_i \cup \{x_j\}_{j\in N_i}\}$. The near-optimal control policy u_i is applied to (3) whose integral provides $\bar{e}_i$ and $\bar{r}_i$.

A control policy designed around the half-load operating condition is used and validated for other operating points, as done in [7] and [9]. The resulting feedback controller requires the knowledge of local states, x_{i_1} and x_{i_2}, which represent the deviations with respect to the target operating point. The target stored energy is fixed at e_i^*, thus, the local state x_{i_1} is easily obtained as $x_{i_1} = \bar{e}_i - e_i^*$. Instead, to obtain x_{i_2}, the unknown value of r_i^*, that depends on the overall operating point, is required. In [7] a low-frequency filter extrapolates the quiescent part of the input resistance, i.e., r_i^* to determines the corresponding actual deviation. However, this filter could introduce delays, distortions, and computational demand. Alternatively, in this paper, the following approximation is adopted

$$x_{i_2} \approx \bar{r}_i - \frac{C}{2}\frac{R_i^* v_i^2}{e_i^*}, \tag{22}$$

where v_i is the measured input voltage and $\bar{e}_i$ represents the energy profile to be tracked by the power buffer. The knowledge of the target load, R_i^*, is needed in (22). Assuming an ideal buck onverter, R_i^* is easily related to the desired load R_{L_i} as $R_i^* = (v_{bi}^*/v_{oi}^*)^2 R_{L_i}$, where v_{oi}^* is the fixed output voltage of the buck converter. The comparison between

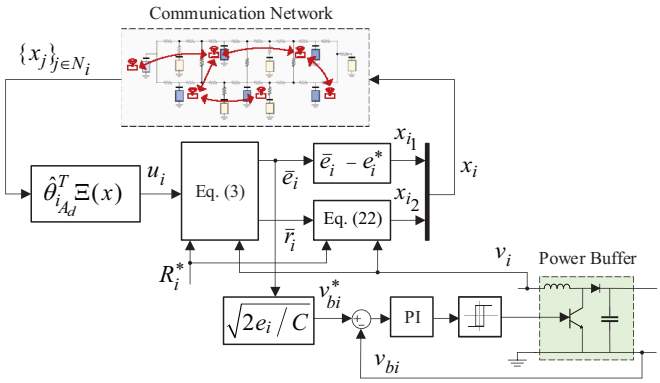

Fig. 3. Control scheme of the i^{th} power buffer.

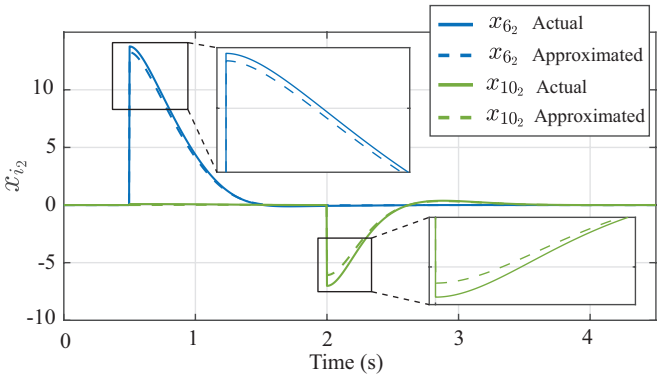

Fig. 4. Comparison of actual states and approximated ones using (22)

the actual and approximated states x_{i_2} in Fig. 4 shows the effectiveness of (22). The depicted scenario uses the local feedback policy $u_i = 2x_{i_1}$, $i = 6, ..., 11$, and varies the final resistive loads of buffers 6 and 10 from 20Ω to 10Ω in $t = 0.5s$ and from 10Ω to 16Ω in $t = 2s$, respectively.

Finally, by translating $\bar{e}_i$ into the reference of the voltage tracker, using (2), the two control objectives mentioned above are attained. The voltage tracker of each boost converter implements a Proportional-Integral (PI) regulator with proportional and integral gains set to 1.2 and 3.7, respectively, followed up with a hysteresis-band controller (band set to 0.2). The buck converter's output voltage is set at 48V with a PI controller, with proportional and integral gains set to 0.09 and 1.08, respectively.

B. Optimizing the Communication Topology

Algorithm 5 solves problem (21) for different values of β. Starting from a fully-connected controller, the TS procedure modifies the current communication topology by applying a set of moves and defining the solutions to visit. For each visited solution, characterized by a specific communication topology, the off-policy IRL procedure in Algorithm 2 finds the corresponding optimal controller. A set of learning data, i.e., $\Delta\Gamma(t_n)$, $\Psi_i(t_n)$, $\Phi_i(t_n)$, $Q_I(t_n)$, with $i \in \mathcal{L}$ and $n = 1, ..., N_L$, is previously collected and used in every run of Algorithm 2. Such data collecting phase is conducted in the Simulink environment on the interconnection of the N subsystems (4) with half-loads values, i.e. $R_i^* = 50\Omega$,

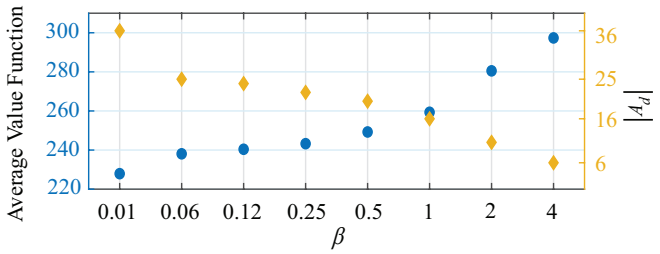

Fig. 5. Optimal average value function and cardinality of A_d for several β.

$i = 6, ..., 11$. Second-order polynomial terms in the 12 states are considered as approximating functions in $\Gamma(x)$, while $\Xi(x) = x$. The learning time intervals are $N_L = 5000$ of $0.01s$ length. The initial controller is $u_i^{(0)} = 2x_{i_1}$, $i = 6, ..., 11$, the stopping threshold is $\delta = 10^{-4}$, and filtered white noises are used as exploration signals.

The stability and performance, i.e., the average value function, of each visited solution are evaluated using Algorithm 3 (see Fig. 2). In particular, each visited solution could be: 1) Unstable if Algorithm 2 does not converge or if the ratio of the failed stability checks, η_F, is higher than the δ_F threshold, herein set to 0.01; 2) Stable with a DOA smaller than the training space, i.e., $\eta_V < 1$; 3) Stable on the full training space, i.e., $\eta_V = 1$. The DoA is estimated using $N_{Rs} = 6000$ randomly-sampled data during the learning stage. Finally, the utility function is defined with $\rho_i(x) = 2$, $i = 6, ..., 11$, and $Q(x) = x^\intercal Q_U x$, where

$$Q_U = \begin{bmatrix} Q_d & Q_2 & Q_6 & Q_6 & Q_2 & Q_4 \\ Q_2 & Q_d & Q_2 & Q_3 & Q_6 & Q_2 \\ Q_6 & Q_2 & Q_d & Q_4 & Q_2 & Q_6 \\ Q_6 & Q_3 & Q_4 & Q_d & Q_4 & Q_2 \\ Q_2 & Q_6 & Q_2 & Q_4 & Q_d & Q_2 \\ Q_4 & Q_2 & Q_6 & Q_2 & Q_2 & Q_d \end{bmatrix}. \quad (23)$$

$Q_d = \mathbf{diag}(30, 15)$ and $Q_k = \mathbf{diag}(-k, 0)$. All entries of matrix A_c are 1. The decision variables are the extra-diagonal elements of A_d, i.e., non-symmetric communication links are allowed while self-loops are present. Each trial of Algorithm 5 uses a tabu length of 15 and a maximum number of iterations of 100.

The average-value function, i.e., $\bar{V}$ in Algorithm 4, and the resulting cardinality, $|A_d|$, of the optimal solutions obtained by eight different trials of Algorithm 5 for increasing values of the weight β in (21), are reported in Fig. 5. Greater values of β promote sparsity with a decreasing number of active communication links. For $\beta = 0.01$, a fully-connected pattern is obtained, i.e., $|A_d| = 36$. For $\beta = 4$, only local controllers are obtained, i.e., $|A_d| = 6$. As expected, increasing sparsity leads to a lower performance evaluated within the randomly-sampled data during the learning phase.

Figure 6 elaborates the results for $\beta = 0.5$ and $\beta = 2$. Figures 6(a), 6(b), and 6(c) show the visited solutions during the optimization procedure. Figure 6(a) presents the visited unstable solutions. The objective function has an infinite value for both $\beta = 0.5$ and $\beta = 2$ despite the gap depicted for presentation purposes only. Figure 6(b) shows the visited stable solutions with $\eta_V < 1$, which implies higher values

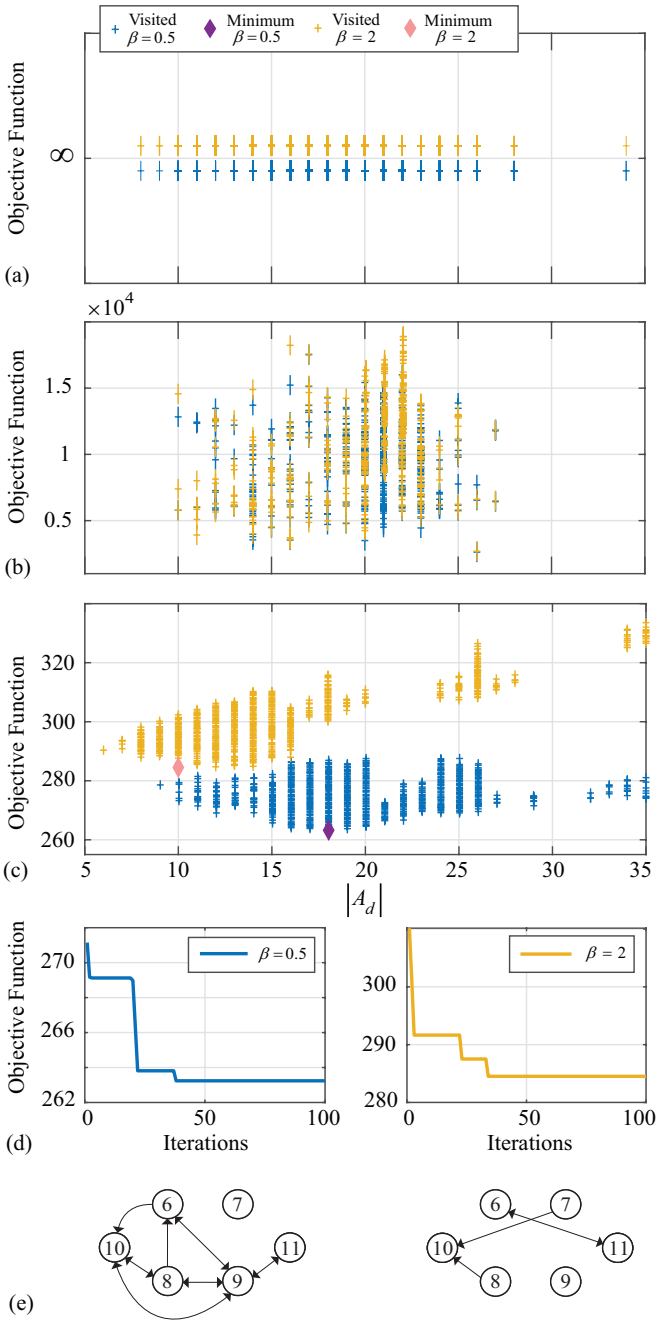

Fig. 6. Results of the optimization stage: (a) visited unstable solutions for $\beta = 0.5$ and $\beta = 2$, (b) visited stable solutions with $\eta_V < 1$ for $\beta = 0.5$ and $\beta = 2$, (c) visited and optimal solutions with $\eta_V = 1$ for $\beta = 0.5$ and $\beta = 2$, (d) best solution for each tabu-search iteration, and (e) optimal communication topologies when $\beta = 0.5$ (left) and $\beta = 2$ (right).

of the objective function, especially when the corresponding DOA is significantly smaller than the training space. The optimal and visited solutions when $\eta_V = 1$ are depicted in Fig. 6(c). Due to its greater value, $\beta = 2$ has higher values of both optimal and visited solutions compared with those of $\beta = 0.5$ in Fig. 6(c). In both cases, proper operation of TS is exhibited through intensified, i.e., more dense, searches around optima. Figure 6(d) shows the trend of best solutions during TS iterations. For $\beta = 0.5$ and $\beta = 2$, the optimum is reached in 38 and 34 iterations, respectively. Finally, Fig. 6(e) shows

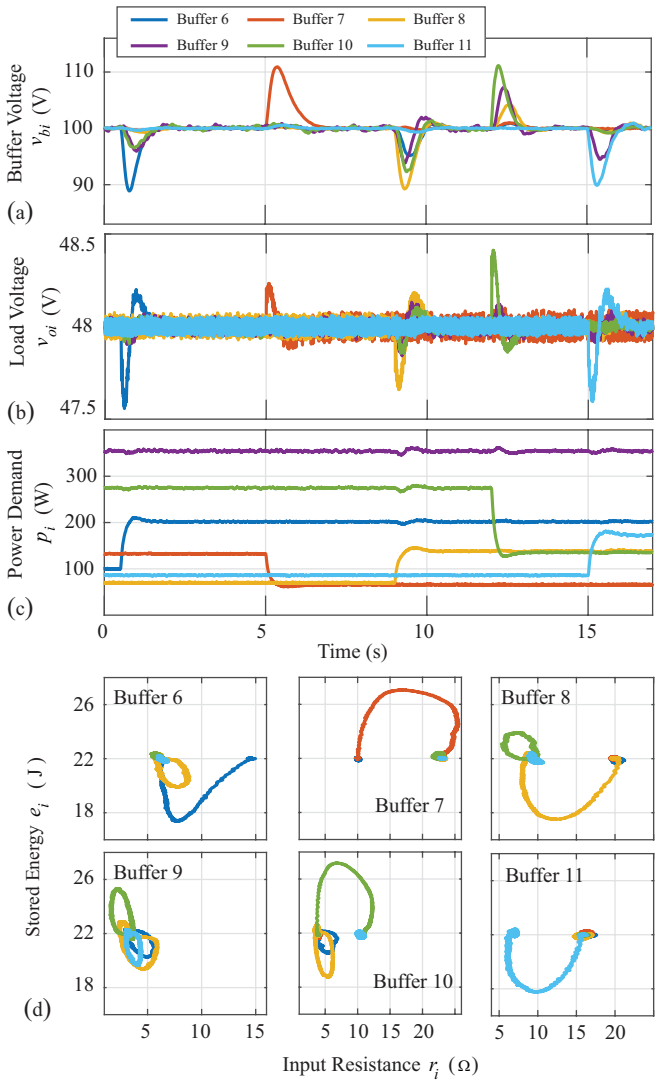

Fig. 7. CHIL validation when $\beta = 0.5$: (a) output voltage of power buffer, (b) output voltage at terminal load resistances, (c) output power of power buffers, and (d) energy-impedance trajectories.

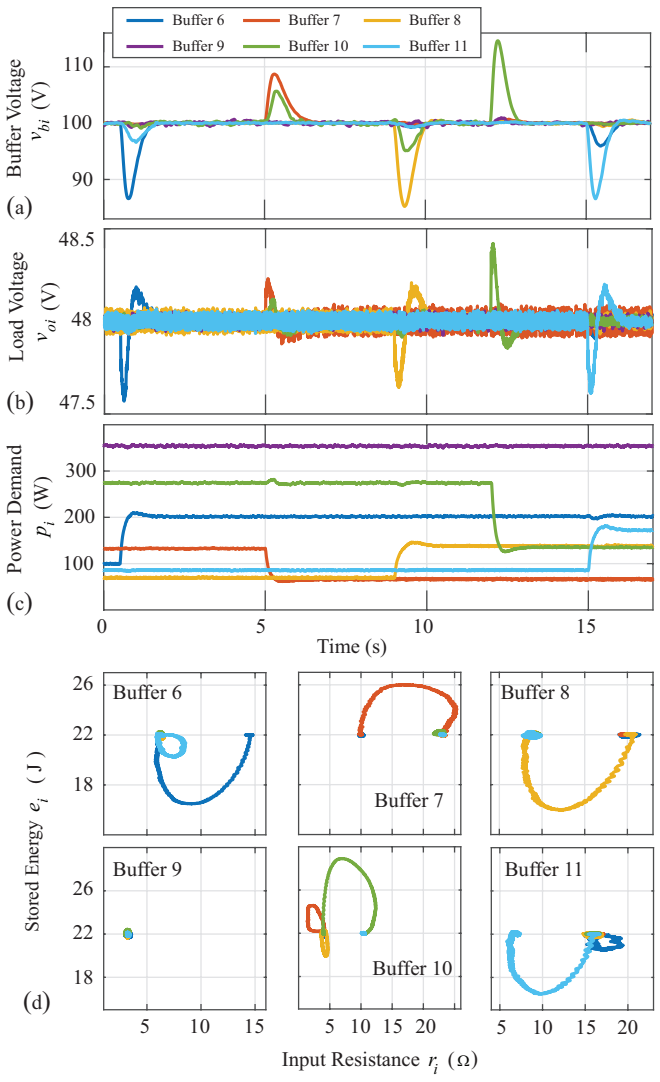

Fig. 8. CHIL validation when $\beta = 2$: (a) output voltage of power buffer, (b) output voltage at terminal load resistances, (c) output power of power buffers, and (d) energy-impedance trajectories.

the optimal communication topologies. Cardinalities of $|A_d|$ for $\beta = 0.5$ and $\beta = 2$ are 20 and 10, respectively.

C. Controller/Hardware-in-the-Loop Studies

CHIL studies for two optimal control policies, with $\beta = 0.5$ and $\beta = 2$, are reported in Fig. 7 and Fig. 8, respectively. Final load resistances are $R_{L_6} = 24\Omega$, $R_{L_7} = 18\Omega$, $R_{L_8} = 35\Omega$, $R_{L_9} = 7\Omega$, $R_{L_{10}} = 9\Omega$, and $R_{L_{11}} = 28\Omega$. As seen in Fig. 7(c) and Fig. 8(c), both scenarios consider the step change in load at $t = 1s$ when load 6 doubles its power demand, i.e., $R_{L_6} = 12\Omega$, at $t = 5s$ when load 7 halves its power demand, i.e., $R_{L_7} = 36\Omega$, at $t = 9s$ when load 8 doubles its power demand, i.e., $R_{L_8} = 17.5\Omega$, at $t = 12s$ when load 10 halves its power demand, i.e., $R_{L_{10}} = 18\Omega$, and at $t = 15s$ when load 11 doubles its power demand, i.e., $R_{L_{11}} = 14\Omega$.

Different communication topologies, as in Fig. 6(e), imply different control behaviors during transients, as highlighted in Fig. 7(a) and Fig. 8(a). The buffer voltages, v_{bi}, $i = 6, ..., 11$, reflect changes in the stored energy according to the distributed control policies actuated by the scheme in Fig. 3. During the first load change, the topology obtained with $\beta = 0.5$ allows power buffers 9 and 10 to change their stored energies and actively assist load 6. With a more sparse communication topology, i.e., $\beta = 2$, the power buffer 6 is assisted only by the power buffer 11. This results in the increased usage for buffer 6 if compared with the previous topology, see Fig. 7(a) and Fig. 8(a) during the first transient. Similar considerations are made for the second load change, where assistance is provided for the case of $\beta = 2$ where buffer 10 assists buffer 7. For $\beta = 0.5$, buffer 7 does not communicate its states, with no changes in other buffer energies. For $\beta = 0.5$, power buffers 6, 10, and 9, reduce the energy usage of buffer 8 during the third transient, when compared with the case of $\beta = 2$, where buffer 8 is assisted only by buffer 10. Better performances are obtained with $\beta = 0.5$ during the fourth and fifth load changes. Figures 7(b) and 8(b) show how the load voltages do not substantially change during the transients due to the

buffering capabilities of the power buffers.

Energy-impedance trajectories are reported in Fig. 7(d) and Fig. 8(d), for two topologies, respectively. Trajectories during the first, second, third, fourth, and fifth load changes are depicted in blue, orange, red, green, and light blue, respectively. Input impedances and stored energies are modified to provide assistance according to the optimized communication topology. For instance, power buffer 9 reacts to changes in buffers 6, 8, 10, and 11, for $\beta = 0.5$. For $\beta = 2$, where only a local controller is active, the stored energy remains constant at its rated value. Less sparse topologies imply more assistance in terms of faster transient responses with lower energy usage. Note that utility functions, i.e., $Q(x)$ and $\rho_i(x)$, can enhance the performances of specified buffers, e.g., by increasing the corresponding diagonal weighting terms in (23).

The effectiveness of the proposed method is demonstrated through a comparison with two other approaches. The first comparison is made with the linear sparsity-promoting algorithm in [11] and used in [13] and [17] for AC microgrid applications. This algorithm is applied to the first-order linearization of (3). The algorithm is tuned such that the resulting optimal topology has the same number of active links as the one obtained by the proposed method. The second comparison is made with an optimal LQR obtained on the first-order linearization of (3) and truncated such that the communication topology coincides with that of the proposed approach. Comparisons are made for the scenario in Fig. 7 and Fig. 8, and for various β parameters, i.e., $\beta = 0.5$, $\beta = 1$, $\beta = 2$, and $\beta = 4$ (fully-decentralized controller). While the computational requirements of the proposed method are higher when compared with [11], the proposed approach could handle nonlinear systems. In both cases, the optimization procedure is conducted offline. Since the basis function set $\Xi(x) = x$ provides a linear feedback controller, the implementation of the proposed real-time controller requires the same computational resources as other controllers obtained via [11] and the truncated LQR.

Figure 9 compares the buffer voltages obtained with the proposed approach (continuous line), [11] (dotted line), and truncated LQR (dashed line), when $\beta = 0.5$ (Fig. 9(a)), $\beta = 1$ (Fig. 9(b)), $\beta = 2$ (Fig. 9(c)), and $\beta = 4$ (Fig. 9(d)). Note that the communication topology obtained with [11] differs from the one obtained by the proposed approach. Thus, when comparing the same load changes, the set of assistive power buffers is different. Compared with both the truncated LQR approach and [11], the proposed method provides faster recovering times for each buffer subject to the load change, i.e., the time needed to restore its initial energy level corresponding to $v_{bi} = 100V$, with a lower maximum energy utilized. The proposed approach shows higher energy drawn from the assisting buffers, to help with the faster restoration of the buffer subject to the load change, e.g., during the first load change in Fig. 9(a), during the last load change in Fig. 9(b), and during the third load change in Fig. 9(c), see corresponding zoomed parts. The proposed method always shows better performance compared with the truncated LQR method. On the other hand, due to different optimized communication topologies, [11] could sometimes show better behaviors, e.g., the second load change in Fig. 9(a), and the third load change in Fig. 9(b), where the corresponding optimized topologies obtained with the proposed method do not provide assistance for buffers 7 and 8, respectively. However, the proposed approach shows better overall performances on the majority of the loading events, with a smaller overall utility function, as shown next. It also provides better responses with fully decentralized controllers, as in Fig. 9(d).

Finally, Table I compares the proposed method against the two other approaches in terms of the resulting utility functions. The base value used to evaluate the percentage variation in the fifth column is the one obtained by applying the fully-connected optimal controller with $\beta = 0.01$. As shown in Fig. 5, greater β implies more sparsity with higher performance values. The proposed approach finds a better compromise between the resulting performance and the number of active communication links, i.e., by comparing topologies for $\beta = 0.5$ and $\beta = 2$, some communication links are activated, and other deactivated, to minimize the impact on the performance index. The proposed method outperforms other approaches, even with more sparse communication topologies (e.g., compare row 6 with rows 4 and 5 in Table I).

TABLE I
CLOSED-LOOP PERFORMANCE COMPARISON BETWEEN PROPOSED APPROACH, [11], AND TRUNCATED LQR

	β	$\lvert A_d \rvert$	Utility	Variation %
Proposed	0.01	36	7702.6	0
LQR		36	7910.3	1.4
Proposed	0.5	18	7883.5	2.3
[11]		18	8150.8	5.8
Truncated LQR		18	8187.4	6.3
Proposed	1	16	7964.7	3.4
[11]		16	8158.8	5.9
Truncated LQR		16	8128.8	5.5
Proposed	2	10	8096.5	5.1
[11]		10	8194.8	6.4
Truncated LQR		10	8227.5	6.8
Proposed	4	6	8137.6	5.7
[11]		6	8193.3	6.4
Truncated LQR		6	8292.5	7.7

V. CONCLUSION

Existing distributed solutions for power buffers in DC microgrids do not consider the effects of the communication network topologies on the controller performance. A second-order approximated model of a DC microgrid considering the physical interconnection among power buffers is developed. Sparsity-promoting optimal control of general interconnected nonlinear systems, including power buffers, are investigated. RL and TS methods find the best compromise between the minimization of a defined closed-loop performance index and the number of activated communication links. TS seeks the best solution by applying some moves on the decision variables matrix, i.e., the communication topology. Controller performance and stability, corresponding to this topology, are

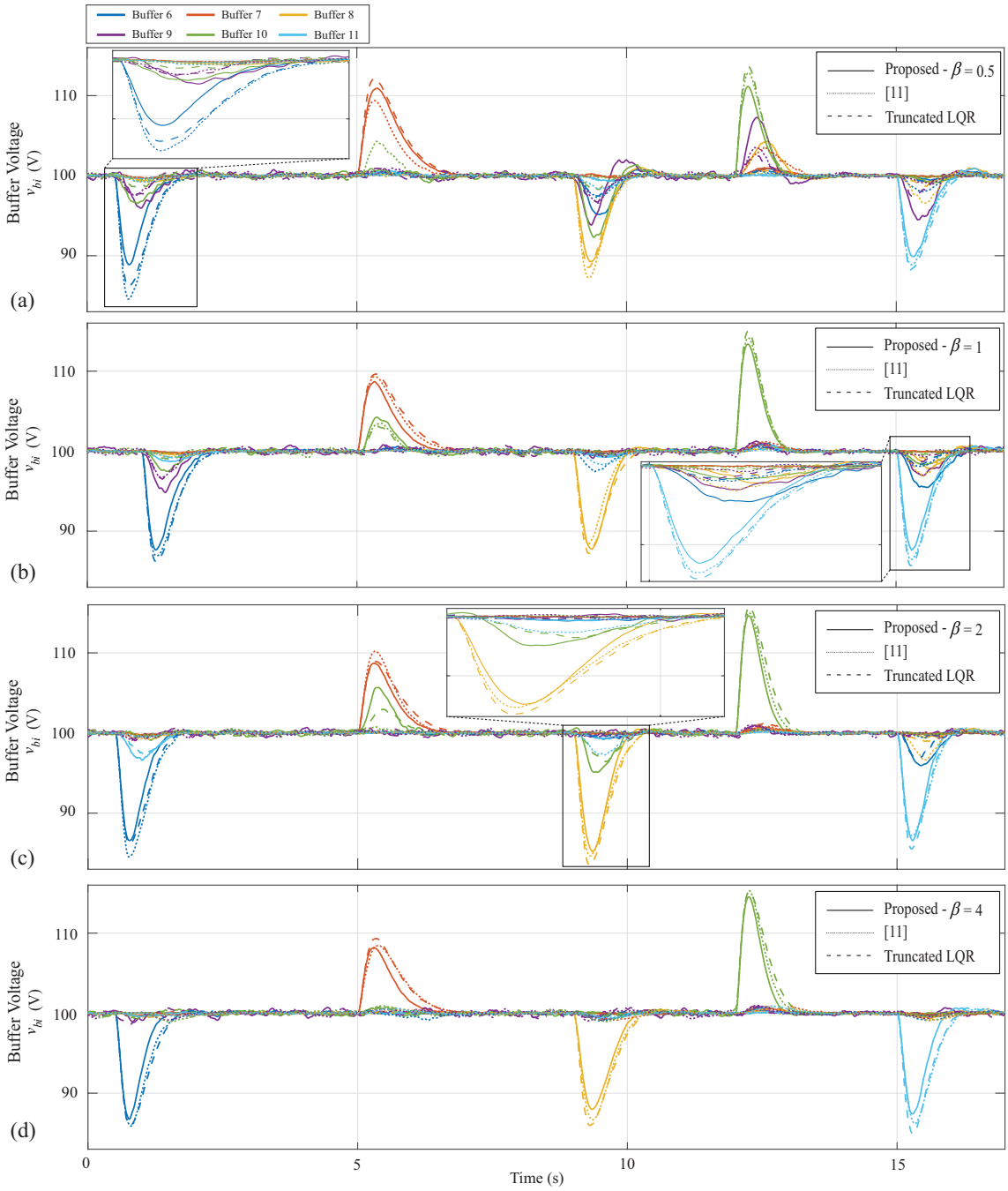

Fig. 9. Comparison of the buffer voltages using the proposed approach, [11], and the truncated LQR: (a) $\beta = 0.5$, (b) $\beta = 1$, (c) $\beta = 2$, and (d) $\beta = 4$.

evaluated using an off-policy learning algorithm and a DoA estimation algorithm. While appearing intuitive, showing that less sparse communication topologies provide better performances is not trivial for nonlinear systems such as DC microgrids. Through CHIL studies, performance improvement following the use of a less sparse communication topology is reflected in a better mutual assistance among the buffers, i.e., faster transient responses with less stored energy utilized. Quantitative comparisons show that the proposed approach outperforms existing methods.

REFERENCES

[1] U. Vuyyuru, S. Maiti, and C. Chakraborty, "Active power flow control between DC microgrids," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5712–5723, Sep. 2019.

[2] M. Hamzeh, M. Ghafouri, H. Karimi, K. Sheshyekani, and J. M. Guerrero, "Power oscillations damping in DC microgrids," *IEEE Trans. Energy Convers.*, vol. 31, no. 3, pp. 970–980, Sep. 2016.

[3] A. Kwasinski and C. N. Onwuchekwa, "Dynamic behavior and stabilization of DC microgrids with instantaneous constant-power loads," *IEEE Trans. on Power Electron.*, vol. 26, no. 3, pp. 822–834, Mar. 2011.

[4] W. W. Weaver and P. T. Krein, "Mitigation of power system collapse through active dynamic buffers," in *Proc. IEEE Power Electron. Spec. Conf.*, 2004, pp. 1080–1084.

[5] X. Wang, D. Vilathgamuwa, and S. Choi, "Decoupling load and power

system dynamics to improve system stability," in *Proc. Int. Conf. Power Electron. Drives Syst.*, 2005, p. 268273.

[6] W. W. Weaver and P. T. Krein, "Optimal geometric control of power buffers," *IEEE Trans. Power Electron.*, vol. 24, no. 5, pp. 1248–1258, May 2009.

[7] L.-L. Fan, V. Nasirian, H. Modares, F. L. Lewis, Y.-D. Song, and A. Davoudi, "Game-theoretic control of active loads in DC microgrids," *IEEE Trans. Energy Convers.*, vol. 31, pp. 882–895, Sep. 2016.

[8] N. C. Ekneligoda and W. W. Weaver, "Game-theoretic communication structures in microgrids," *IEEE Trans. Power Del.*, vol. 27, pp. 2334–2341, Oct. 2012.

[9] V. Nasirian, A. P. Yadav, F. L. Lewis, and A. Davoudi, "Distributed assistive control of power buffers in DC microgrids," *IEEE Trans. Energy Convers.*, vol. 32, pp. 1396–1406, Dec. 2017.

[10] P. R. Massenio, D. Naso, F. L. Lewis, and A. Davoudi, "Assistive power buffer control via adaptive dynamic programming," *IEEE Trans. Energy Convers.*, vol. 35, pp. 1534–1546, Sep. 2020.

[11] F. Lin, M. Fardad, and M. R. Jovanovic, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2426–2431, Sep. 2013.

[12] S. Schuler, P. Li, J. Lam, and F. Allgöwer, "Design of structured dynamic output-feedback controllers for interconnected systems," *Int. J. Control*, vol. 84, no. 12, pp. 2081–2091, Dec. 2011.

[13] F. Dorfler, M. R. Jovanovic, M. Chertkov, and F. Bullo, "Sparsity-promoting optimal wide-area control of power networks," *IEEE Trans. Power Syst.*, vol. 29, no. 5, pp. 2281–2291, Sep. 2014.

[14] S. Schuler, U. Münz, and F. Allgöwer, "Decentralized state feedback control for interconnected systems with application to power systems," *J. Process Control*, vol. 24, no. 2, pp. 379–388, Feb. 2014.

[15] V. Tanyingyong, R. Olsson, J. woo Cho, M. Hidell, and P. Sjodin, "IoT-grid: IoT communication for smart DC grids," in *Proc. IEEE Global Comm. Conf.* IEEE, Dec. 2016.

[16] Y. Tian and J. A. Taylor, "Sparsity-promoting controller design for VSC-based microgrids," in *Proc. IEEE Global Signal Information Processing Conf.* IEEE, Dec. 2016.

[17] X. Wu, F. Dorfler, and M. R. Jovanovic, "Input-output analysis and decentralized optimal control of inter-area oscillations in power systems," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2434–2444, May 2016.

[18] A. Al-Digs, S. V. Dhople, and Y. C. Chen, "Measurement-based sparsity-promoting optimal control of line flows," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5628–5638, Sep. 2018.

[19] A. Jain, A. Chakrabortty, and E. Biyik, "An online structurally constrained LQR design for damping oscillations in power system networks," in *Proc. IEEE American Control Conf.* IEEE, May 2017.

[20] N. Gaeini, A. M. Amani, M. Jalili, and X. Yu, "Optimization of communication network topology in distributed control systems subject to prescribed decay rate," *IEEE Trans. Cyber.*, pp. 1–9, 2019.

[21] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control*. Hoboken: John Wiley & Sons, Inc., Jan. 2012.

[22] D. E. Kirk, *Optimal Control Theory: An Introduction*. Dover Publications, 2004.

[23] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, pp. 32–50, 2009.

[24] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems," *Automatica*, vol. 50, no. 1, pp. 193–202, 2014.

[25] K. G. Vamvoudakis and F. L. Lewis, "Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.

[26] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comput. Intell. Mag.*, vol. 4, pp. 39–47, May 2009.

[27] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling." Elsevier, 2014, vol. 50, no. 1, pp. 493–525.

[28] B. Luo, D. Liu, H.-N. Wu, D. Wang, and F. L. Lewis, "Policy gradient adaptive dynamic programming for data-based optimal control," *IEEE Trans. Cybern.*, vol. 47, pp. 3341–3354, Oct. 2017.

[29] H. Zhang, H. Jiang, Y. Luo, and G. Xiao, "Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method," *IEEE Trans. Ind. Electron.*, vol. 64, no. 5, pp. 4091–4100, 2016.

[30] G. Xiao, H. Zhang, Y. Luo, and H. Jiang, "Data-driven optimal tracking control for a class of affine non-linear continuous-time systems with completely unknown dynamics," *IET Control Theory & Applications*, vol. 10, no. 6, pp. 700–710, 2016.

[31] E. Najafi, R. Babuška, and G. A. D. Lopes, "A fast sampling method for estimating the domain of attraction," *Nonlinear Dynamics*, vol. 86, no. 2, pp. 823–834, Jul. 2016.

[32] G. Yuan and Y. Li, "Estimation of the regions of attraction for autonomous nonlinear systems," *Trans. Inst. Meas. Control*, vol. 41, no. 1, pp. 97–106, Mar. 2018.

[33] U. Topcu, A. Packard, P. Seiler, and G. Balas, "Robust region-of-attraction estimation," *IEEE Trans. Autom. Control*, vol. 55, no. 1, pp. 137–142, Jan. 2010.

[34] F. Glover, "Tabu searchpart i," *ORSA J. Comput.*, vol. 1, no. 3, pp. 190–206, 1989.

[35] ——, "Tabu searchpart ii," *ORSA J. Comput.*, vol. 2, no. 1, pp. 4–32, 1990.

[36] Y. Jiang and Z.-P. Jiang, *Robust adaptive dynamic programming*. Hoboken: John Wiley & Sons, Inc., 2017.

[37] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, Jan. 1989.

[38] Y. P. Kumar and R. Bhimasingu, "Alternative hardware-in-the-loop (HIL) setups for real-time simulation and testing of microgrids," in *Proc. IEEE Power Electronics, Intelligent Control and Energy Systems Conf.* IEEE, Jul. 2016.

[39] W. W. Weaver and G. G. Parker, "Real-time hardware-in-the-loop simulation for optimal dc microgrid control development," in *Control and Modeling for Power Electronics, IEEE 15th Workshop on*. IEEE, Jun. 2014, pp. 1–6.