Assistive Power Buffer Control via Adaptive Dynamic Programming

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

***Abstract*—Power buffers are power electronic converters, with large capacitors, that decouple volatile loads and a low-inertia distribution network in a DC microgrid. In this work, a set of distributed optimal control policies enable power buffers to reciprocally assist each other during abrupt load changes. While the majority of existing control paradigms are localized, enabling communication among buffers extends their effective range of assistance and helps them minimize a shared objective in a cooperative fashion. The control law's weights surfaces are learned for a mesh of reference loads of each power buffer. Hamilton-Jacobi-Bellman equation is solved by a continuous-time adaptive dynamic programming (ADP) approach with off-policy learning to directly provide a feedback controller, instead of existing approaches that obtain open-loop policies via Pontryagin's minimum principle. This paper presents the first attempt in using ADP techniques for the control of power buffers that respects their original nonlinear dynamics, overcoming the limitations of previous approaches based on small-signal analysis. Compared to the current literature, the proposed approach provides trained controllers that are known a priori, avoiding player-by-player solutions or real-time optimization procedures that could degrade performances or become computationally intensive. Hardware-in-the-loop emulations of a low-voltage DC microgrid validates the proposed approach.**

***Index Terms*—Assistive control, Adaptive dynamic programming, DC microgrid, Power buffer.**

I. Introduction

DC microgrids are efficient alternatives to their AC counterparts given the emerging DC-native sources, loads, and storage units, and to avoid issues afflicting AC systems. DC microgrids face a compound challenge of having a resistive grid with low damping/generational inertia, while handling potentially volatile source and load profiles [1], [2]. Power buffers are power electronics converters, with larger capacitors, that shield the grid from abrupt load changes, by partially compensating the transients mismatch [3], [4]. Since buffers are placed at load terminals, they exhibit faster responses compared to a central energy storage. Moreover, they provide an additional degree of freedom that can be exploited to design control laws that improve the microgrid performance.

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

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). P. R. Massenio is currently a visiting scholar at the University of Texas at Arlington.

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).

Proper control of power buffers has been an active area of research. The majority of existing solutions use a game-theoretic control framework with power buffers as players [5]–[11]. Different control objectives for power buffers are defined within the game-theoretic problem, e.g., to meet power or voltage drop requirements [5], to achieve a constant power characteristics while minimizing network loss [7], to find optimal controllers with respect to quadratic functionals at each sample time [8], to conserve as much energy as possible while preventing system collapse [10], or to simply conserve the buffer's stored energy [11]. In the absence of a closed-form solution to the game-theoretic problem, a turn-based approach is employed [5], [10] that could adversely affect the controller performance and stability as the system size increases. Alternatively, the game-theoretic solutions are found in [7], [11] using Pontryagin's minimum principle, with sliding-mode controllers used to actuate the resulting open-loop optimal trajectories. In [8], the solution is found by the means of linear optimal control approaches (i.e., by solving Riccati equations). Some of these solutions are implemented in a decentralized fashion, relying on individual objectives with non-cooperative strategies [5], [7], [8], [10], [11]. Communication-based cooperative methods are presented in [6], [9] as an alternative to non-cooperative solutions. A Policy Iteration algorithm solves the linear coupled Riccati equations in [6], where the individual objectives are defined with regards to team-aligned and selfish components. In [9], a turn-based approach implements the solution of a leader-follower Stackleberg game to prioritize leader's objective, and finds an optimum set of information to be transmitted. Finally, an assistive control strategy, based on linear distributed approaches, is presented in [12] where, as in [6], the coupling effects of the power distribution grid are considered. However, both [6] and [12] rely on small-signal approximations of power buffers, rendering them invalid to study large-signal behaviors.

This paper designs a distributed assistive control scheme for power buffers in a DC microgrid. The control scheme is *distributed* as buffers exchange information through a communication network, *cooperative* as buffers share a common objective, and *assistive* as buffers reciprocally assist each other during abrupt load changes, improving overall network performance and stability. Cooperative control techniques are already applied to other domains, e.g., unmanned aerial vehicles [13], [14], robot manipulators [15], and spacecrafts [16], and have recently been extended to DC microgrids (e.g., distributed primary/secondary control [17]). We formulate the assistive control problem of power buffers using the optimal con-

trol theory. In general, continuous-time closed-loop optimal control problems solve the Hamilton-Jacobi-Bellman (HJB) equation [18]. Adaptive dynamic programming (ADP) [19], [20] approximates the HJB solution for nonlinear systems as it could become analytically intractable [21]. ADP is usually implemented through an actor-critic learning structure, with a critic neural network (NN) to approximate the value function and an actor NN to approximate the optimal control policy [20]. Learning techniques can be categorized into on-policy and off-policy methods. On-policy learning methods update the current policy with data collected from the same policy. Off-policy learning methods permit the repeated use of the data collected from a single initial admissible and stable policy [22]. ADP techniques also provide forward-in-time approximated solutions to dynamic programming approaches for discrete-time optimal control problems [23].

This paper presents the first attempt in using ADP techniques for the real-time control of power buffers that respects their original nonlinear dynamics. ADP has also provided optimal energy management policies in smart grids approximating the solution of the Bellman equation forward-in-time [24], [25]. In [26] and [27], a discrete-time ADP algorithm solves the optimal energy management problem for microgrids with energy storage elements. In [28], the fair energy scheduling problem for a vehicle-to-grid network is solved via ADP. A self-learning ADP algorithm in [29] considers the real-time electricity price, load demand, and solar energy. Continuous-time on-policy ADP approaches provide reactive power control in wind farms [30] and improve unmatched disturbance rejection in multi-machine power systems [31]. Continuous-time ADP algorithms, based on concurrent-learning, develop droop-free control for DC microgrids [32]. The game-theoretic solution for power buffers in [6] are provided via a policy iteration algorithm.

In this paper, we solve the HJB equation employing a continuous-time ADP approach with off-policy learning for the purpose of feedback design instead of operational scheduling. The objective is to derive a set of distributed optimal control policies able to provide assistance during abrupt load changes. A communication network, spread across the distribution grid, augments the assisting range of power buffers to nearby loads. The control law's weights sets are calculated based on a mesh of reference loads for each power buffer. The control law depends on the buffer's state and those of its neighbors on a communication graph. To further reduce both computational requirements and communicated data, the controller is triggered only when a load change occurs, making it suitable for Internet-of-Things (IoT) devices. The main contributions of this paper, in contrast with existing literature, are

- The distributed controller minimizes a shared objective among power buffers in a cooperative fashion, as opposed to non-cooperative strategies in [5], [7], [8], [10], [11].
- The feedback strategy is designed according to the optimal control theory, providing a real-time controller that is known a priori and does not need a turn-based approach as in [5], [9], [10].
- Compared to the work that rely on a small-signal approximation of power buffers [6], [8], [12], the proposed nonlinear optimal control law takes into account the nonlinear dynamics of the power buffers and the coupling power grid, and is valid for large-signal variations.
- It does not solve linear-quadratic regulator (LQR) problems at each sampling instant as in [8].
- The optimal control problem is solved by approximating the solution to HJB equation, instead of employing the Pontryagin's minimum principle as in [7] and [11]. This provides both necessary and sufficient conditions for optimality instead of only the necessary condition. Moreover, it provides a closed-loop control law directly implemented without the need of other control techniques (e.g., sliding mode), offering simpler designs as well as better performances with small parameter variations or model uncertainty [18], [33].

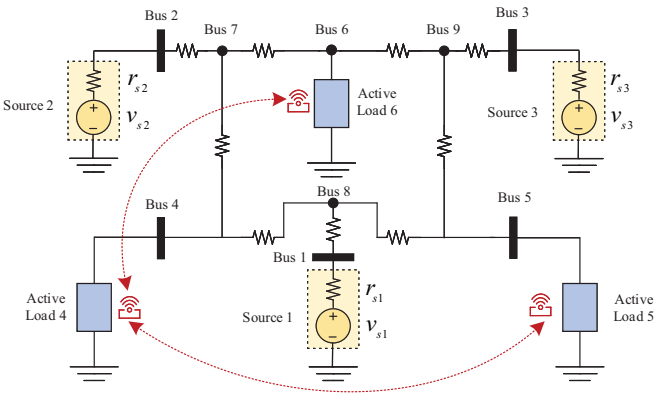

Fig. 1. A DC microgrid with sources, active loads (composed of a power buffer and a point-of-load converter), and the power grid. The assistance range of a buffer is enhanced with a communication module.

The rest of this paper is outlined as follows. Section II provides the nonlinear dynamic model of the power buffers and the power grid. The proposed control approach is explained in Section III. This controller is then verified using a Hardware-in-the-Loop (HIL) setup in Section IV. Concluding remarks are given in Section V.

II. Nonlinear Dynamic Model of a DC Microgrid

In this paper, the *final load* refers to a point-of-load converter. The *active load* is defined as the series connection of the final load and the power buffer. A localized control approach limits the assistance capabilities of a buffer to its final load. Introducing a communication network among nearby active loads, as shown in Fig. 1, allows them to collectively respond to transients. The control objectives of a power buffer are to 1) regulate its output voltage to a rated value at the steady state, and 2) vary its input impedance profile during transients according to an assistive policy. The architecture of an active load is shown in Fig. 2. The state variables of the i^{th} active load include the energy stored in the buffer, e_i, and its input impedance, r_i [6], [12]. $x_{i1} = e_i - e_i^*$ and $x_{i2} = r_i - r_i^*$ are deviations of the buffers' energy and its input impedance from their steady-state values. The set of all neighbors of the i^{th} active load is called its *neighborhood set*, and denoted by N_i. An active load transmits its own state $\mathbf{x}_i = [x_{i1}\ x_{i2}]^\intercal$, and its buffer's output resistance R_i, to its neighborhood set.

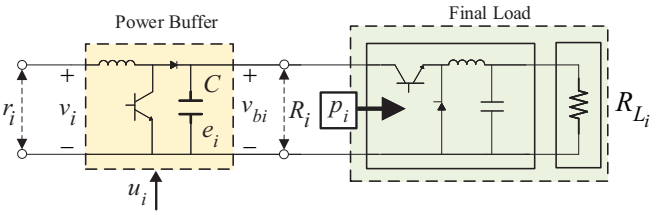

Fig. 2. An active load is composed of a power buffer and a point-of-load converter. The input resistance, r_i, is regulated by the control input, u_i.

Consider a microgrid with M sources numbered $1, ..., M$, and N active loads numbered $M+1, ..., M+N$. Let p_i be the power delivered to the final load. The energy-balance for the i^{th} power buffer can be estimated as

$$\dot{e}_i = \frac{v_i^2}{r_i} - p_i, \;\; i = M+1, ..., M+N, \tag{1}$$

where v_i is the bus voltage. Moreover, the energy-voltage relation can be approximated as

$$e_i = \frac{1}{2} C v_{bi}^2, \;\; i = M+1, ..., M+N, \tag{2}$$

where C is the capacitance of the power buffer, and v_{bi} is the buffer's output voltage. u_i is the control input that regulates the input impedance of the buffer, r_i. The following state-space model for the i^{th} active load is obtained

$$\begin{cases} \dot{e}_i = \frac{v_i^2}{r_i} - \frac{2e_i}{C}\frac{1}{R_i} \\ \dot{r}_i = u_i \end{cases}, \;\; i = M+1, ..., M+N. \tag{3}$$

R_i is the equivalent resistance of the buffer's output. Given a point-of-load converter (e.g., buck converter), in the steady state, R_i can be obtained from the load's resistance, R_{L_i}.

Sources are modeled as a series connection of a voltage source, v_{si}, and a resistor, r_{si}. The admittance matrix of a distribution grid relates its injected nodal currents and the bus voltages. Likewise, the active loads and sources can be related

$$\begin{aligned} \mathbf{i} = [v_{s1}/r_{s1}, ..., v_{sM}/r_{sM} \mid 0, ..., 0]^\intercal = \\ \mathbf{Y} [v_1, ..., v_M \mid v_{M+1}, ..., v_{M+N}]^\intercal, \end{aligned} \tag{4}$$

where $\mathbf{i}$ is the vector of injected currents, and $\mathbf{Y}$ is the reduced-order admittance matrix [34]. From (4), the input voltage of an active load can be related to the input impedances of all active loads:

$$\begin{aligned} v_i = \gamma_i(r_{M+1}, ..., r_i, ..., r_{M+N}), \\ i = M+1, ..., M+N. \end{aligned} \tag{5}$$

Each active load's input voltage is affected by its own input impedance, r_i, and by the impedances of all the other active loads. Let's define N_i as the set of all the indexes $k \in \{M+1, ..., M+N\}$ such that active load k is in the neighborhood of the i^{th} active load. In this paper, the neighbors set is inspired by the physical vicinity, i.e., for any value of the input resistances $(r_{M+1}, ..., r_i, ..., r_{M+N})$

$$\left| \frac{\partial \gamma_i(r_{M+1}, ..., r_{M+N})}{\partial r_j} \right| \ll \left| \frac{\partial \gamma_i(r_{M+1}, ..., r_{M+N})}{\partial r_k} \right| \tag{6}$$

for any $j \in N \backslash N_i$ and for any $k \in N_i$. Thus, the dependency of (5) on the non-neighbors can be neglected by setting the resistances of the non-neighbor loads to infinity, allowing the following approximated relation

$$v_i = \hat{\gamma}_i(r_i, \{r_j\}_{j \in N_i}), \;\; i = M+1, ..., M+N. \tag{7}$$

The resulting dynamic model for an active load becomes

$$\begin{cases} \dot{e}_i = \frac{\hat{\gamma}_i(r_i, \{r_j\}_{j \in N_i})^2}{r_i} - \frac{2e_i}{C}\frac{1}{R_i} \\ \dot{r}_i = u_i \end{cases}, \\ i = M+1, ..., M+N. \tag{8}$$

Given a set of output loads $\left[R_{L_{M+1}} \cdots R_{L_{M+N}}\right]$, the corresponding steady-state values of the buffer's output resistances $\left[R^*_{M+1} \cdots R^*_{M+N}\right]$ can be obtained. Once the output resistances are known, (8) is solved by setting its derivatives to zero and numerically finding the steady-state value of the input impedances r_i^*. Since $x_{i1} = e_i - e_i^*$ and $x_{i2} = r_i - r_i^*$, (8) can be rewritten as

$$\begin{cases} \dot{x}_{i1} = \frac{\hat{\gamma}_i(x_{i2}+r_i^*, \{x_{j2}+r_j^*\}_{j \in N_i})^2}{x_{i2}+r_i^*} - \frac{2\left(x_{i1}+e_i^*\right)}{C}\frac{1}{R_i^*} \\ \dot{x}_{i2} = u_i \end{cases}, \\ i = M+1, ..., M+N. \tag{9}$$

This can be written as

$$\dot{\bar{\mathbf{x}}}_i = f_i(\bar{\mathbf{x}}_i) + \mathbf{b} u_i, \;\; i = M+1, ..., M+N, \tag{10}$$

with $\mathbf{b} = [0 \;\; 1]^\intercal$, $\mathbf{x_i} = [x_{i1} \;\; x_{i2}]^\intercal$, $\bar{\mathbf{x}}_i = (\mathbf{x}_i^\intercal, \{\mathbf{x}_j\}_{j \in N_i})^\intercal$, and $f_i(\bar{\mathbf{x}}_i)$ defined as

$$\begin{aligned} f_i(\bar{\mathbf{x}}_i) = \frac{\hat{\gamma}_i(x_{i2}+r_i^*, \{x_{j2}+r_j^*\}_{j \in N_i})^2}{x_{i2}+r_i^*} - \frac{2\left(x_{i1}+e_i^*\right)}{C}\frac{1}{R_i^*} \\ i = M+1, ..., M+N. \end{aligned} \tag{11}$$

The dynamics of the entire DC microgrid then becomes

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}_{\mathbf{M+1}} \\ \vdots \\ \dot{\mathbf{x}}_{\mathbf{M+N}} \end{bmatrix}}_{\dot{\mathbf{x}}_{\mathbf{MG}}} = \underbrace{\begin{bmatrix} f_i(\bar{\mathbf{x}}_{M+1}) \\ \vdots \\ f_i(\bar{\mathbf{x}}_{M+N}) \end{bmatrix}}_{f_{MG}(\mathbf{x_{MG}})} + \underbrace{\begin{bmatrix} \mathbf{b} \cdots \mathbf{0} \\ \vdots \ddots \vdots \\ \mathbf{0} \cdots \mathbf{b} \end{bmatrix}}_{\mathbf{B_{MG}}} \underbrace{\begin{bmatrix} u_{M+1} \\ \vdots \\ u_{M+N} \end{bmatrix}}_{\mathbf{u}}, \tag{12}$$

with the origin as an equilibrium, and $f_{MG}(\mathbf{0}) = \mathbf{0}$.

Remark 1. For a large number of power buffers, once output resistances are fixed in value, a more computationally-tractable method would be to integrate the evolution of system (8) under a stable feedback controller, with a pre-computed equilibrium as its initial state. The feedback controller can be any stable policy that regulates the energy stored in the capacitor at its fixed steady-state value, e_i^*, independent from the operating point. The steady-state values reached by the impedance trajectories will represent the steady-state solutions r_i^* for $i = M+1, ..., M+N$.

III. Proposed Distributed Optimal Approach

A. Assistive Control as an Optimal Control Problem

The assistive control problem can be treated as finding an optimal feedback control law for (12) that minimizes a cost functional during the transient toward a given setpoint for any

initial state. Suppose that the i^{th} active load needs assistance; The cost functional is

$$J_i(\mathbf{x_{MG_0}}, \mathbf{u}) = \int_0^{\infty} U_i(\mathbf{x_{MG}}, \mathbf{u}) dt \quad i = M+1, ..., M+N. \tag{13}$$

$\mathbf{x_{MG_0}}$ is the initial state at $t = 0$. $U_i(\cdot, \cdot)$ is the corresponding *utilty function*, defined as

$$U_i(\mathbf{x_{MG}}, \mathbf{u}) = \mathbf{x_i^\intercal Q_{ii} x_i} + \rho_i u_i^2 + \sum_{j \in N_i} (\mathbf{x_i^\intercal Q_{ij} x_j} + \mathbf{x_j^\intercal Q_{jj}^{(i)} x_j} + \rho_j^{(i)} u_j^2), \tag{14}$$

where $\mathbf{Q_{ii}} \in \mathbb{R}^{2\times 2}$, $\mathbf{Q_{jj}^{(i)}} \in \mathbb{R}^{2\times 2}$, and $\mathbf{Q_{ij}} \in \mathbb{R}^{2\times 2}$ are performance matrices weighting the state of active load i, the state of its neighbors, and their product, respectively. $\rho_i^{(i)}$ and $\rho_j^{(i)}$ are scalars weighting the active load's control input and that of its neighbors, respectively. The weighting terms ensure $U_i(\mathbf{x_{MG}}, \mathbf{u}) \geqslant 0 \forall (\mathbf{x_{MG}}, \mathbf{u})$ and $U_i(\mathbf{0}, \mathbf{0}) = 0$. Note that (14) can also be written as

$$U_i(\mathbf{x_{MG}}, \mathbf{u}) = Q_i(\mathbf{x_{MG}}) + \mathbf{u^\intercal P_i u}, \tag{15}$$

Equation (13) represents a common objective shared among the active load i and its neighbors. With the utility function (14), optimization of (13) minimizes the states deviations, $\mathbf{x_i}$ and $\mathbf{x_j}$, and the control effort, u_i, of each assisting active load. A proper choice of the weights penalizes individual active load's action in favor of a collective action during transients.

Remark 2. In this paper, the relationship between the control input, u_i, and the actual switching signal for the power converter is not static as it will be clarified in the subsequent. Therefore, we consider an optimal control problem with an unconstrained input. To incorporate saturating or constrained inputs, a non-quadratic term in the control input could be considered. Interested readers may refer to [32], [35]–[37].

B. Adaptive Dynamic Programming with Off-Policy Learning

Given any initial state $\mathbf{x_{MG_0}}$, the assistive control problem finds a feedback law $\mathbf{u}(\mathbf{x_{MG}})$ that asymptotically drives the $\mathbf{x_{MG}}$ to the origin and minimizes (13), when the active load i is in need. Firstly, the corresponding Hamiltonian is defined:

$$H(\mathbf{x_{MG}}, \mathbf{u}, V^*) \triangleq U_i(\mathbf{x_{MG}}, \mathbf{u}) + \nabla V^{*\intercal}[f_{MG}(\mathbf{x_{MG}}) + \mathbf{B_{MG}} \mathbf{u}], \tag{16}$$

where $V^*(\mathbf{x_{MG_0}}) = \min_{\mathbf{u}} J_i(\mathbf{x_{MG_0}}, \mathbf{u})$ is the optimal value function that satisfies the well-known continuous-time HJB equation, i.e.,

$$\min_{\mathbf{u}} H(\mathbf{x_{MG}}, \mathbf{u}(\mathbf{x_{MG}}), V^*) = 0. \tag{17}$$

Once V^* is found, the optimal feedback control law is

$$\mathbf{u}^*(\mathbf{x_{MG}}) = -0.5 \mathbf{P_i^{-1} B_{MG}^\intercal} \nabla V^*(\mathbf{x_{MG}}). \tag{18}$$

The generally-intractable HJB equation can be numerically solved by the Policy Iteration algorithm detailed in Algorithm 1, where sequences $\mathbf{u}_k(\cdot)$ and $V_k(\cdot)$ converge to the optimal values. The analytical solution of step 2 is still an intractable problem. To this end, an ADP algorithm with off-policy learning can be used [38]. Let's consider the following system

$$\dot{\mathbf{x}}_{\mathbf{MG}} = f_{MG}(\mathbf{x_{MG}}) + \mathbf{B_{MG}}(\mathbf{u}_0(\mathbf{x_{MG}}) + \mathbf{e}_n(t)), \tag{19}$$

where $\mathbf{u}_0$ is a feedback policy that asymptotically stabilizes the system at the origin with a finite associated cost, and $\mathbf{e}_n : \mathbb{R} \to \mathbb{R}^N$ is a bounded *exploration noise* for the learning purposes. For each iteration $k \geqslant 0$, let $\mathbf{u}'_k = \mathbf{u}_0 - \mathbf{u}_k + \mathbf{e_n}$. Then, (19) can become

$$\dot{\mathbf{x}}_{\mathbf{MG}} = f_{MG}(\mathbf{x_{MG}}) + \mathbf{B_{MG}} \mathbf{u}_k + \mathbf{B_{MG}} \mathbf{u}'_k. \tag{20}$$

The time-derivative of the function $V_k(\mathbf{x_{MG}})$, computed along the state trajectory of (20), is

$$\begin{aligned} \dot{V}_k(\mathbf{x_{MG}}) &= \nabla V_k^\intercal(\mathbf{x_{MG}})[f_{MG}(\mathbf{x_{MG}}) + \mathbf{B_{MG}}(\mathbf{u}_k + \mathbf{u}'_k)] \\ &= -U_i(\mathbf{x_{MG}}, \mathbf{u}_k) - 2 \sum_{j \in N_i \cup \{i\}} u_{j_{k+1}} \rho_j^{(i)} u'_{j_k}, \end{aligned} \tag{21}$$

where u_{j_k} and u'_{j_k} are the j^{th} elements of $\mathbf{u}_k$ and $\mathbf{u}'_k$ vectors, respectively. Using the universal approximation property [39], for each $k \geqslant 0$, the value function $V_k(\mathbf{x_{MG}})$ and the control policies $u_{j_{k+1}}$, $j \in N_i \cup \{i\}$, can be approximated in a linear-in-parameters (LIP) fashion,

$$\hat{V}_k(\mathbf{x_{MG}}) = \sum_{l=1}^{N_A} c_{k_l} \phi_l(\mathbf{x_{MG}}) = \mathbf{c}_k^\intercal \Phi(\mathbf{x_{MG}}), \tag{22}$$

and

$$\hat{u}_{j_{k+1}}(\bar{\mathbf{x}}_j) = \sum_{l=1}^{N_B^{(j)}} w_{j_{k_l}} \psi_{j_l}(\bar{\mathbf{x}}_j) = \mathbf{w}_{j_k}^\intercal \Psi_j(\bar{\mathbf{x}}_j), \quad j = M+1, ..., M+N. \tag{23}$$

Herein, $\phi_l(\mathbf{x_{MG}}) : \mathbb{R}^{2N} \to \mathbb{R}$, with $l = 1, ..., N_A$, and $\psi_{j_l}(\bar{\mathbf{x}}_j) : \mathbb{R}^{2(|N_i|+1)} \to \mathbb{R}$, with $l = 1, ..., N_B^{(j)}$ and $j = M+1, ..., N$, are sequences of linearly-independent smooth functions vanishing at the origin, and defined on compact sets containing the origin. N_A and $N_B^{(j)}$ are sufficiently-large integers. $\mathbf{c}_k$ and $\mathbf{w}_{j_k}$ are constant row vectors of weights to be determined. Note that the approximating functions in (23) depend only on the current state and that of the neighbors. In this way, it is possible to find an approximated optimal control policy that stabilizes the system and, at the same time, is distributed. Replacing V_k and $u_{j_{k+1}}$ in (21) with their

Algorithm 1 Policy Iteration Algorithm

1. **Initialization:** Let the initial iteration number be $k = 0$, and $\mathbf{u_0}(\mathbf{x_{MG}})$ be the initial control policy.
2. **Policy Evaluation:** Solve for $V_k(\mathbf{x_{MG}}) \in C^1$, with $V_k(\mathbf{0}) = 0$, from the following
$$\nabla V_k^\intercal(\mathbf{x_{MG}})[f_{MG}(\mathbf{x_{MG}}) + \mathbf{B_{MG}} \mathbf{u}_k] + U_i(\mathbf{x_{MG}}, \mathbf{u}_k) = 0.$$
3. **Policy Improvement:** Update the control policy $\mathbf{u}_{k+1}(\mathbf{x_{MG}})$ using (18) and $V_k(\mathbf{x_{MG}})$.
4. **Stopping Criterion:** if convergence is achieved then stop, else, set $k = k + 1$ and go to Step 2.

approximations, and by integrating both sides over any time interval $[t, t+T]$, the following equality is obtained,

$$\begin{aligned}\mathbf{c}_k^\intercal \underbrace{[\phi(\mathbf{x_{MG}}(t_{n+1})) - \phi(\mathbf{x_{MG}}(t_n))]}_{\mathbf{D\Phi}(n+1)} = \\ -2\sum_{j\in N_i\cup\{i\}} \mathbf{w}_{jk}^\intercal \underbrace{\int_{t_n}^{t_{n+1}} \mathbf{\Psi}_j(\bar{\mathbf{x}}_j)\rho_j^{(i)}(u_{0_j}+e_{n_j})dt}_{\mathbf{\Gamma}_j(n+1)} \\ +2\sum_{j\in N_i\cup\{i\}} \mathbf{w}_{jk}^\intercal \underbrace{\int_{t_n}^{t_{n+1}} \mathbf{\Psi}_j(\bar{\mathbf{x}}_j)\rho_j^{(i)}\mathbf{\Psi}_j^\intercal(\bar{\mathbf{x}}_j)dt}_{\mathbf{\Gamma}_j^S(n+1)}\mathbf{w}_{jk-1} \\ -\int_{t_n}^{t_{n+1}} U_i(\mathbf{x_{MG}},\hat{\mathbf{u}}_k)dt + \epsilon_{k_n},\end{aligned} \tag{24}$$

where ϵ_{k_n} is the approximation error and $\{t_n\}_{n=1}^{N_L}$ is an increasing series of time intervals, with $N_L > 0$ as a sufficiently-large number. The last integral can be written as

$$\begin{aligned}\int_{t_n}^{t_{n+1}} U_i(\mathbf{x_{MG}},\hat{\mathbf{u}}_k)dt = \underbrace{\int_{t_n}^{t_{n+1}} Q_i(\mathbf{x_{MG}})dt}_{Q_I(n+1)} \\ +\sum_{j\in N_i\cup\{i\}} \mathbf{w}_{jk-1}^\intercal \mathbf{\Gamma}_j^S(n+1)\mathbf{w}_{jk-1}.\end{aligned} \tag{25}$$

By defining $\mathbf{\Xi}_{k-1}$ as in (26) and $\mathbf{B}_{k-1}$ as follows

$$\mathbf{B}_{k-1} = -\begin{bmatrix} Q_I(1) + \sum_{j\in N_i\cup\{i\}} \mathbf{w}_{jk-1}^\intercal \mathbf{\Gamma}_j^S(1)\mathbf{w}_{jk-1} \\ \vdots \\ Q_I(N_L) + \sum_{j\in N_i\cup\{i\}} \mathbf{w}_{jk-1}^\intercal \mathbf{\Gamma}_j^S(N_L)\mathbf{w}_{jk-1}, \end{bmatrix} \tag{27}$$

with $j1, ..., jz \in N_i \cup \{i\}$, the unknown weights at iteration k can be find by solving the following

$$\mathbf{\Xi}_{k-1}\begin{bmatrix}\mathbf{c}_k^\intercal & \mathbf{w}_{j1_k}^\intercal & \dots & \mathbf{w}_{jz_k}^\intercal\end{bmatrix}^\intercal = \mathbf{B}_{k-1} \tag{28}$$

Starting from an initial stabilizable control policy $\mathbf{u_0}$, sequences $\{\hat{V}_k\}_{k=0}^{\infty}$ and $\{\hat{\mathbf{u}}_{k+1}\}_{k=0}^{\infty}$ converge to V_k and $\mathbf{u}_k$, respectively [38]. The weights $\mathbf{w}_{jk}$ and $\mathbf{c}_k$ are obtained by minimizing $\sum_{n=0}^{N_L}\epsilon_{k_n}^2$ using a least-squares method in equation (28). As commonly required in adaptive control theory [40], a persistence of excitation (PE) condition has to be met to successfully reach convergence for the weight sets. The right choice of the exploratory signal is pivotal in this case. The PE condition is given in (29).

$$rank(\mathbf{\Xi}_{k-1}) = N_A + \sum_{j\in N_i\cup\{i\}} N_B^{(j)} \tag{29}$$

Since PE condition cannot be verified analytically a priori, it requires a trial and error approach.

The Algorithm 2 implements the ADP algorithm once the approximating functions, learning data, and utility function are given. It involves an off-policy learning since the iteration phase is done once the training data is available. Thanks to the properties of the LIP approximators, the data collecting phase is decoupled from the evaluation of (24). The computational efforts are reduced since the same collected data solves several optimal control problems with different utility functions.

C. Learning Procedure

Algorithm 2 is exploited to obtain a set of near-optimal policies with respect to all the active loads, considering several setpoints. The obtained weight sets are then aggregated into look-up tables to compose a control scheme able to provide a near-optimal policy working in all scenarios. Algorithm 3 summarizes the learning procedure. For a set of loads for the i^{th} active load, $P_{R_{L_i}} = \{R_{L_i}^{*(1)}, .., R_{L_i}^{*(S_i)}\}$, the corresponding set $P_{R_i} = \{R_i^{*(1)}, .., R_i^{*(S_i)}\}$ can be found. A learning grid of different loads is defined as $P_{R_{M+1}} \times ... \times P_{R_{M+N}}$. For each element of this learning grid, a set of optimal control problems is defined corresponding to each active load that needs assistance for that specific setpoint. Then, given an N-tuple $(\bar{R}_{M+1}^*, ..., \bar{R}_{M+N}^*) \in P_{R_{M+1}}$, corresponding input impedances $(\bar{r}_{M+1}^*, ..., \bar{r}_{M+N}^*)$ are found by solving (9) in the steady state. The input impedances are stored in a map, $M_r(\bar{R}_{M+1}^*, ..., \bar{R}_{M+N}^*)$, to compute the states of each active load fed to the controller. Once all the reference values are given, the data collection phase can be performed. For each setpoint, N corresponding optimal control problems are solved by means of Algorithm 2. The obtained control weights for the i^{th} problem are stored in a map, $M_{w_j}^i(\bar{R}_{M+1}^*, ..., \bar{R}_{M+N}^*)$, for each $j \in N_i \cup \{i\}$. This map defines the control policy of active load j that is triggered through a load change in the active load i.

Algorithm 2 ADP Algorithm with Off-Policy Learning

Inputs: Approximating functions $\mathbf{\Phi}(\mathbf{x_{MG}})$ and $\mathbf{\Psi}_j(\bar{\mathbf{x}}_j)$, with $j = M+1, ..., M+N$; initial controller weights $\mathbf{w}_{j_0}$ for each active load; sequence $\{t_n\}_{n=1}^{N_L}$; system's collected data $\mathbf{x_{MG}}^{(L)}$ recorded by applying $(\hat{\mathbf{u}}_0 + \mathbf{e}_n)$ as input, with $\hat{u}_{j_0} = \mathbf{w}_{j_0}^\intercal \mathbf{\Psi}_j(\bar{\mathbf{x}}_j)$; weighting matrices $\mathbf{Q_{ii}}$, $\mathbf{Q_{ij}}$, $\mathbf{Q_{jj}^{(i)}}$, and scalars ρ_i, $\rho_j^{(i)}$ as in (14); stop threshold δ.

Outputs: weights $\hat{\mathbf{c}}$ and $\hat{\mathbf{w}}_j$, $j = M+1, ..., M+N$.

1. **Initialization:** Set the initial iteration number as $k = 1$.
2. **Data Evaluation:** Evaluate weights-independent terms of (24) with collected data $\mathbf{x_{MG}}^{(L)}$.
3. **Policy Improvement:** Find new weights $\mathbf{c}_k$ and $\mathbf{w}_{j_k}$ from (24) using the data evaluated at Step 2.
4. **Off Policy Iteration:** If $|\mathbf{c}_k - \mathbf{c}_{k-1}| \leqslant \delta$, then stop and return approximated optimal control policy, i.e., set $\hat{\mathbf{c}} = \mathbf{c}_k$ and $\hat{\mathbf{w}}_j = \mathbf{w}_{j_k}$, $j = M+1, ..., M+N$; else, set $k = k+1$ and repeat Step 3.

$$\mathbf{\Xi}_{k-1} = \begin{bmatrix} \mathbf{D\Phi}(1) & 2(\mathbf{\Gamma}_{j1}(1) - \mathbf{\Gamma}_{j1}^S(1)\mathbf{w}_{j1_{k-1}})^\intercal & \cdots & 2(\mathbf{\Gamma}_{jz}(1) - \mathbf{\Gamma}_{jz}^S(1)\mathbf{w}_{jz_{k-1}})^\intercal \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{D\Phi}(N_L) & 2(\mathbf{\Gamma}_{j1}(N_L) - \mathbf{\Gamma}_{j1}^S(N_L)\mathbf{w}_{j1_{k-1}})^\intercal & \cdots & 2(\mathbf{\Gamma}_{jz}(N_L) - \mathbf{\Gamma}_{jz}^S(N_L)\mathbf{w}_{jz_{k-1}})^\intercal \end{bmatrix} \tag{26}$$

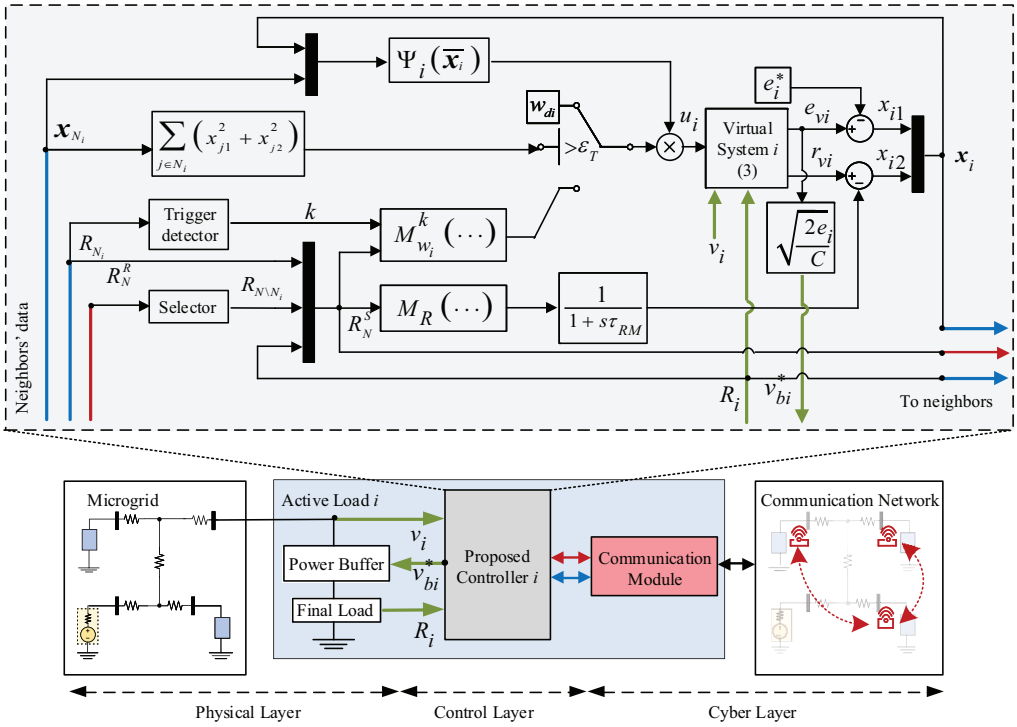

Fig. 3. Proposed control scheme. Green, blue, and red lines refer to local data, incoming/outgoing real-time data, and incoming/outgoing high-latency data.

D. Assistive Control Scheme

As the first control objective, the buffer's output voltage should be fixed, in the steady state, on the rated value of v_{bi}^*, which corresponds to e_i^* as in (2). As the second objective, the input impedance profile varies during transients according to the assistive control policy. The proposed scheme employs the voltage tracker to handle both objectives. The assistive policy acts on the software implementation of system (9), defined as the *virtual system* in Fig. 3, and is connected to the physical buffer through the input voltage, v_i. e_{vi} and r_{vi} in Fig. 3 denote the states of the virtual system synced with the physical one through the input voltage, v_i. The real-time controller uses the states of the virtual system in its feedback policy. In particular, the real-time value of r_{vi} is obtained by integrating the control input, u_i. The controller drives the input impedance of the virtual system, providing a desired energy profile translated into the reference of the voltage tracker of the power buffer.

The distributed assistive control policy is triggered when an active load detects a change in its neighborhood. Otherwise, it uses a default local stabilizing controller $u_{di}(\mathbf{x}_i) = \mathbf{w}_{di}\mathbf{\Psi}_i(\bar{\mathbf{x}}_i)$; Where $\mathbf{w}_{di}$ is chosen such that u_{di} depends only on local states. After each transient, i.e., when $\sum_{j\in N_i}(x_{j1}^2 + x_{j2}^2)$ is lower than a defined threshold ϵ_T, the control weights switch to the default ones. Thus, the communication module is used only during the assistive task. This makes the proposed method suitable for energy-constrained devices (e.g., IoT devices), and keeps the system stable in case of communication fails.

The reference and weights map in Algorithm 3 are queried by the complete N-tuple $(R_{M+1}^*, ..., R_{M+N}^*)$. To correctly query the maps, active load i has to know its resistance, the set of neighbors, and the set of non-neighbors resistances,

Algorithm 3 Assisitve Control Learning Procedure

Inputs: Buffer's output resistances set for each active load $P_{R_i} = \{R_i^{*(1)}, .., R_i^{*(S_i)}\}$; Approximating functions, sequence $\{t_n\}_{n=1}^{N_L}$, and initial controller weights $\mathbf{w}_{j_0}$ for each active load, as in Algorithm 2; weighting terms for each active load i, i.e. $\mathbf{Q_{ii}}$, $\rho_i^{(i)}$, $\mathbf{Q_{ij}}$, $\mathbf{Q_{jj}^{(i)}}$ and $\rho_j^{(i)}$, with $j \in N_i$.
Outputs: Input impedance references map $M_r(R_{M+1}, ..., R_{M+N})$; near-optimal control policies map $M_{w_j}^i(R_{M+1}, ..., R_{M+N})$, with $i = M+1, ..., M+N$ and $j \in N_i \cup \{i\}$.

1. **for** each $(\bar{R}_{M+1}^*, ..., \bar{R}_{M+N}^*) \in P_{R_{M+1}} \times ... \times P_{R_{M+N}}$ **do**
2. Find corresponding $(\bar{r}_{M+1}^*, ..., \bar{r}_{M+N}^*)$ by solving (9) in the steady state, with $R_i^* = \bar{R}_i^*$, and set
$$M_r(\bar{R}_{M+1}^*, ..., \bar{R}_{M+N}^*) = (\bar{r}_{M+1}^*, ..., \bar{r}_{M+N}^*).$$
3. Define system (12) by setting reference values found in Step 2, and collect corresponding learning data using the initial controller.
4. **for** each active load i **do**
5. Solve the optimal control problem using Algorithm 2 with learning data from Step 3 and terms $\mathbf{Q_{ii}}$, $\rho_i^{(i)}$, $\mathbf{Q_{ij}}$, $\mathbf{Q_{jj}^{(i)}}$ and $\rho_j^{(i)}$, with $j \in N_i \cup \{i\}$, and set
$$M_{w_j}^i(\bar{R}_{M+1}^*, ..., \bar{R}_{M+N}^*) = \hat{\mathbf{w}}_j.$$
6. **end for**
7. **end for**

i.e., R_i, $R_{N_i} = \{R_j\}_{j\in N_i}$, and $R_{N\setminus N_i} = \{R_j\}_{j\in N\setminus N_i}$, respectively. Thus, the control mechanism is enhanced with a communication protocol to broadcast each routing active load's vector $R_N^S = (R_i, R_{N_i}, R_{N\setminus N_i})$ to its neighbors. This protocol ensures consensus among active loads if the communication graph features a spanning tree [41]. Once a load change occurs, the neighbors state data is sent in real time, while the information R_N^S is sent with a higher latency. The maximum latency has to be lower than the minimum rate of load change for each active load. Once the i^{th} active load detects a change in R_{N_i}, the non-neighbors resistances are selected from R_N^R, which is the received counterpart of R_N^S. Hence, the active load can correctly query both the weights and reference maps.

Assuming that the learning procedure has been properly conducted, and given the stabilizing properties of the default policy, a switch between asymptotically-stable controllers occurs once the transient effects are dissipated. The output of the reference map is filtered to avoid states jump and preserve system stability during the switching phase [42]. This filter's time constant, τ_{RM}, is chosen faster than the communication sampling time.

IV. Hardware-in-the-Loop Validation

A. System Setup

The proposed control scheme is verified on a 48V DC microgrid, with its structure shown in Fig. 1. Microgrid parameters are adopted from [6]. Every DC source is modeled as a series connection of a $50V$ ideal voltage source and a 0.1Ω resistor. Each active load consists of a power buffer (boost converter) and a buck converter with an LC filter interposed in between. The boost converter features a fast voltage tracker to follow the voltage profile defined by the assistive control scheme in Fig. 3. Its rated output voltage is $v_{bi}^* = 100V$. The fast voltage regulator of the buck converter is regulated at 48V. Both voltage trackers employ Proportional-Integral (PI) controllers. Proportional and integral gains for the boost converter are 1 and 3.5, respectively. Proportional and integral gains for the buck converter are 0.09 and 1.08, respectively.

The relationship between the control input, u_i, and the switching state of the solid-state switch of the boost converter can be derived as follows. Given the initial value of the stored energy, e_{i0}, input impedance r_{i0}, and load value R_i, the control input profile, u_i, is translated into the energy profile, e_{vi}, by integrating the equations of the virtual system as in Fig. 3

$$e_{vi}(t) = e^{-\frac{2}{C}\frac{1}{R_i}} \left(e_{i0} + \int_0^t \frac{e^{\frac{2}{C}\frac{1}{R_i}\tau} v_i(\tau)^2}{r_{i0} + \int_0^\tau u_i(\zeta)d\zeta} d\tau \right), \quad (30)$$

where v_i is the measured input voltage of power buffer i. Using (2), e_{vi} is translated into the reference of the fast voltage tracker for the boost converter, i.e., $v_{bi}^*(t) = \sqrt{(2/C)e_{vi}(t)}$. The output of the Proportional-Integral controller of the i-th boost converter is denoted by y_i^{PI}, while its input is the error between the reference voltage, $v_{bi}^*(t)$, and the measured output voltage, v_{bi}. y_i^{PI} is used along with the measured input current, i_i, in an hysteresis-band controller to determine the switching state of the solid-state device,

$$d_i(t) = \begin{cases} 1 & if \quad y_i^{PI} - i_i > hb_i \\ 0 & if \quad y_i^{PI} - i_i < -hb_i \end{cases}, \quad (31)$$

where hb_i is the hysteresis band herein set as 0.2. The switch status is kept constant for values between the thresholds.

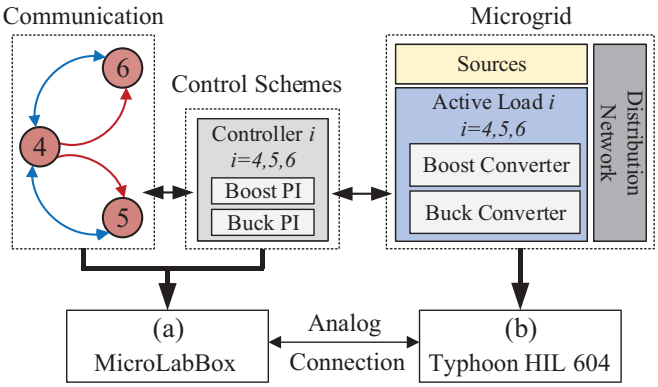

Fig. 4. Hardware-in-the-loop setup: (a) dSPACE MicroLabBox Controller (handles the control and communication routines); (b) Typhoon HIL 604 (emulates the physical components of the underlying microgrid).

The physical microgrid is emulated on a Typhoon HIL 604, and the communication network and the control scheme run on a dSPACE MicroLabBox controller board, as shown in Fig. 4. The sampling times of the controller and the communication module are $0.1ms$ and $1ms$, respectively. The time constant of the filter placed after the resistances map is $\tau_{RM} = 0.2ms$.

B. Learning Stage

The assistive control scheme requires a learning phase via Algorithm 3. According to Fig. 1, neighborhood sets are $N_4 = \{5, 6\}$, $N_5 = 4$, and $N_6 = 4$. The output load for each active load varies from 10Ω to 100Ω, in steps of 10Ω. Mixed linear-independent polynomial terms, up to 4^{th} degree, are used as approximating functions, with a corresponding $N_A = 166$, $N_B^{(4)} = 83$, and $N_B^{(5)} = N_B^{(6)} = 34$. The structure of approximating functions, for both critic and actor networks, is

$$\begin{cases} \hat{\mathbf{V}}_k(\mathbf{x_{MG}}) = \sum\limits_{\substack{l=1,\dots,166 \\ i_1,\dots,i_6\geqslant 0 \\ 2\leqslant i_1+\dots+i_6\leqslant 4}} c_l x_{4_1}^{i_1} x_{4_2}^{i_2} x_{5_1}^{i_3} x_{5_2}^{i_4} x_{6_1}^{i_5} x_{6_2}^{i_6} \\ \hat{u}_{4_{k+1}}(\bar{\mathbf{x}}_4) = \sum\limits_{\substack{l=1,\dots,83 \\ i_1,\dots,i_6\geqslant 0 \\ 1\leqslant i_1+\dots+i_6\leqslant 3}} w_{4_{k_l}} x_{4_1}^{i_1} x_{4_2}^{i_2} x_{5_1}^{i_3} x_{5_2}^{i_4} x_{6_1}^{i_5} x_{6_2}^{i_6} \\ \hat{u}_{5_{k+1}}(\bar{\mathbf{x}}_5) = \sum\limits_{\substack{l=1,\dots,34 \\ i_1,\dots,i_4\geqslant 0 \\ 1\leqslant i_1+\dots+i_4\leqslant 3}} w_{5_{k_l}} x_{4_1}^{i_1} x_{4_2}^{i_2} x_{5_1}^{i_3} x_{5_2}^{i_4} \\ \hat{u}_{6_{k+1}}(\bar{\mathbf{x}}_6) = \sum\limits_{\substack{l=1,\dots,34 \\ i_1,\dots,i_4\geqslant 0 \\ 1\leqslant i_1+\dots+i_4\leqslant 3}} w_{6_{k_l}} x_{4_1}^{i_1} x_{4_2}^{i_2} x_{6_1}^{i_3} x_{6_2}^{i_4} \end{cases} \quad (32)$$

The learning sequence $\{t_n\}$, with $N_L = 10000$ intervals of $10ms$ and 3 filtered white noises, are used as exploration

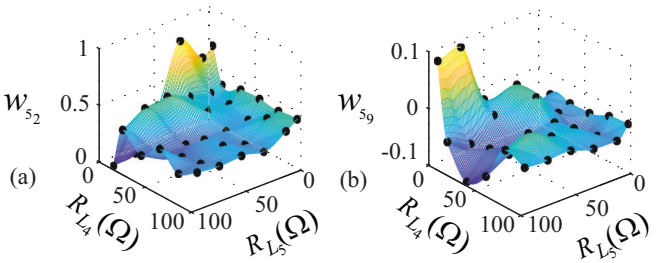

Fig. 5. Two policy weights of the active load 5, when the active load 4 is in need: (a) weights for approximating function x_{5_1}; (b) weights for approximating function $x_{4_1}x_{5_1}$.

signals. For each active load, the initial stabilizing controller is $\hat{u}_{0_j} = 2x_{j1}, j = 4, 5, 6$. A trial and error approach finds the initial controller whose stability has been checked through simulations conducted over several loading scenarios. The same steady-state stored energy were considered for all the buffers. The weighting terms are set as

$$\begin{cases} \mathbf{Q_{44}} = \mathbf{Q_{55}} = \mathbf{Q_{66}} = diag(8, 8), \\ \mathbf{Q}^{(5)}_{\mathbf{44}} = \mathbf{Q}^{(6)}_{\mathbf{44}} = \mathbf{Q}^{(4)}_{\mathbf{55}} = \mathbf{Q}^{(6)}_{\mathbf{55}} = diag(1, 1), \\ \mathbf{Q_{45}} = diag(-2, 0), \\ \mathbf{Q_{46}} = diag(-1, 0), \\ \mathbf{Q_{54}} = \mathbf{Q_{64}} = diag(-5, 0), \\ \rho_4 = \rho_6 = \rho_6 = 1, \rho^4_5 = \rho^4_6 = \rho^5_4 = \rho^6_4 = 0.1. \end{cases} \tag{33}$$

where *diag* stands for a diagonal matrix. Once the learning phase is complete, the near-optimal control policy maps are interpolated to obtain different control weights surfaces for each active load with respect to each neighbor in need. As an example, Fig. 5 shows two surfaces actuated by the active load 5 and triggered when the active load 4 needs assistance. Note that the weights depend on the desired setpoint (here, $R_{L_6} = 70\Omega$).

Example studies from Algorithm 2 in Fig. 6 show how the near-optimal control policy provides assistance among neighboring power buffers. Using formulation (12) for the underlying DC microgrid, a single control policy, $\hat{\mathbf{u}}$, was obtained to assist power buffer 5 during transients with the same weighting terms described above. In this example, R_5 changes from 80Ω to 10Ω at $t = 0$, while R_4 and R_6 are set as 40Ω and 30Ω, respectively. Figures 6(a) and 6(b), respectively, show the trajectories of e_4, e_5 and r_4, r_5 both with the initial control policies u_{0_4}, u_{0_5}, and with the near-optimal control policies $\hat{u}_4$, $\hat{u}_5$. These control policies are compared in Fig. 6(c). The initial control policy of power buffer 4 did not provide assistance to the power buffer 5, while the near-optimal control policy of power buffer 4 uses its stored energy to help power buffer 5 during transients, reducing both the energy and input impedance variations for power buffer 5.

The controller stability depends on the approximation domain of the employed neural networks in the learning stage [38]. The exploration signal allows the system states to span the region for the considered loading scenarios. Thus, the near-optimal control policy becomes stable, providing an approximated optimal value function $\hat{V}$, that acts as a Lyapunov

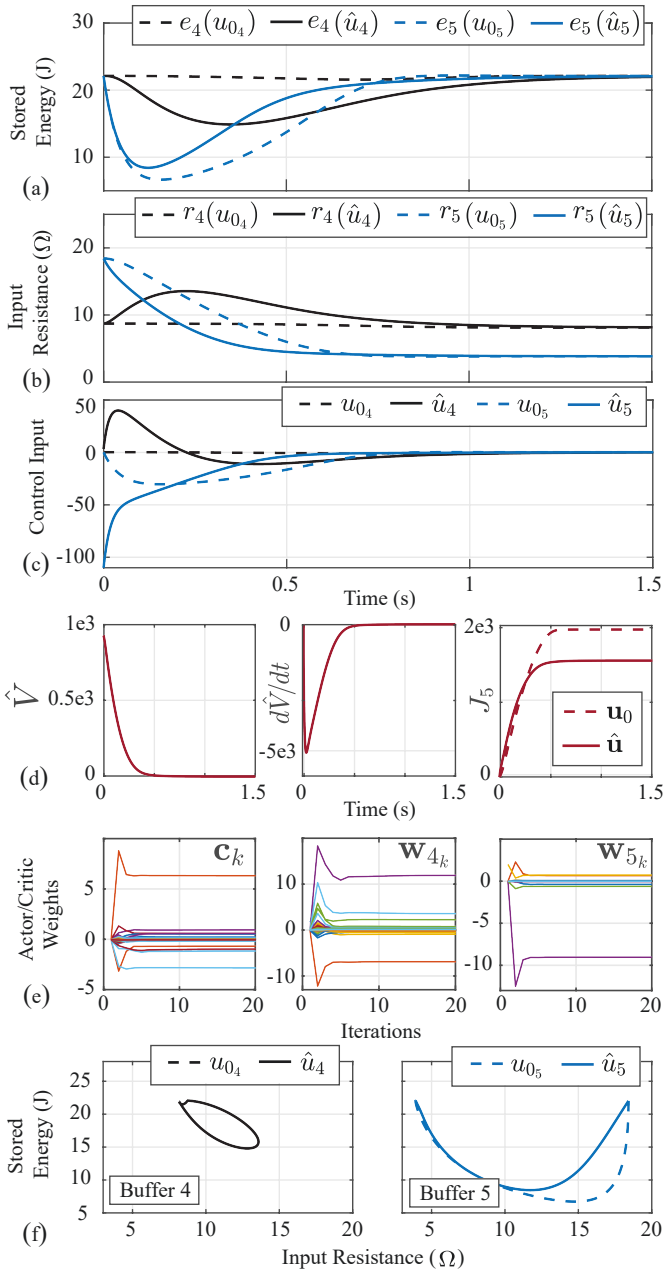

Fig. 6. Learning results when power buffer 5 is in need: (a) stored energies for initial and near-optimal controllers, (b) input resistances for initial and near optimal controllers, (c) initial and near-optimal control inputs, (d) time trajectories of the learned value function (left), derivative of the learned value function (center), and performance comparison (right), (e) weights convergence for the critic network (left), actor network of power buffer 4 (center), and actor network of power buffer 5 (right), and (f) energy-impedance trajectories for the initial and near-optimal control policies.

function, as shown in the left and central parts of Fig. 6(d). The performances of the two control policies, in minimizing the cost function, is shown in the right part of Fig. 6(d). Clearly, the near-optimal controller, $\hat{\mathbf{u}}$, provides a lower value for the shared objective function, J_5. The weights convergence for this scenario is depicted in Fig. 6(e). Finally, Fig. 6(f) shows the energy-impedance trajectories for the two power buffers, with both the initial controller and the near-optimal one. As seen,

the initial control policy for power buffer 4 doesn't change its stored energy, while its approximated optimal policy assists power buffer 5, by using the buffering capabilities of power buffer 4.

C. HIL Implementation - Deactivated Power Buffers

Figure 7 shows the system performance when power buffers are inactive. The initial loads of active loads 4, 5 and 6 are 80Ω, 100Ω and 70Ω, respectively. The load attached to the power buffer 5 changes to 20Ω at $t = 2s$. The load attached to the power buffer 4 goes to 15Ω at $t = 9s$. Loads 4 and 5 regain their original values at $t = 15s$ and $t = 25s$, respectively. Bus voltages and source currents exhibit step-change behaviors in Figs. 7(a) and 7(d), respectively. Note that when slow or stochastic (renewable) sources are present, such abrupt demands on the source currents are highly undesired. The energy-impedance trajectories of active loads are shown in Fig. 7(h). The trajectories corresponding to the first, second, third, and fourth load changes are represented by red, green, orange, and violet lines, respectively. The operating points of buffers 4 and 5 form an almost straight line, while buffer 6 doesn't show any change.

D. HIL Implementation - Activated Power Buffers with Communication Delays

The proposed control scheme is activated, and the communication network links the neighboring active loads. Some studies report IEEE 802.11 (WiFi) or Bluetooth Low Energy (BLE) as communication protocols mostly suited for low-power IoT devices [43]. During the assistive task, a data packet with 3 doubles (R_{L_i}, x_{i1}, and x_{i2}) is communicated, which would require a single link capacity of 192 kbps. Maximum data rates for WiFi and BLE are 54Mbps and 1 Mbps, respectively [44]. Thus, BLE is suitable for microgrids with up to 5 neighbors for each active load; Otherwise, WiFi is preferred. For both protocols, the maximum transport delay is less than $100ms$ [45], [46]. Communication delays of $125ms$, $120ms$, and $130ms$ are introduced in the links between active loads 4 and 5, active loads 5 and 4, and active loads 4 and 6, respectively.

For $0 < t < 2s$, all the power buffers run a default control law, the same as the u_{0i} in Section IV.B. At $t = 2s$, the active load 5 changes from 100Ω to 20Ω, while active loads 4 and 6 stay at 80Ω and 70Ω, respectively. The active load 4 receives the load-change signal after $120ms$, triggering the assistive control law by querying the references map and the weight surfaces. Once the transient is over, active load 4 switches to the default control law and updates the active load 6. Thus, the active load 6 can correctly query the references map at the next event. At $t = 9s$, the active load 4 changes to 15Ω, triggering its own near-optimal policy. After $125ms$ and $130ms$, respectively, active loads 5 and 6 receive the information and trigger their control policies to assist the active load 4. At $t = 15s$ and $t = 25s$, active loads 4 and 5 are changed back to their initial values, respectively.

After the first load change event, the active load 4 is only supporting the active load 5. The second event requires that

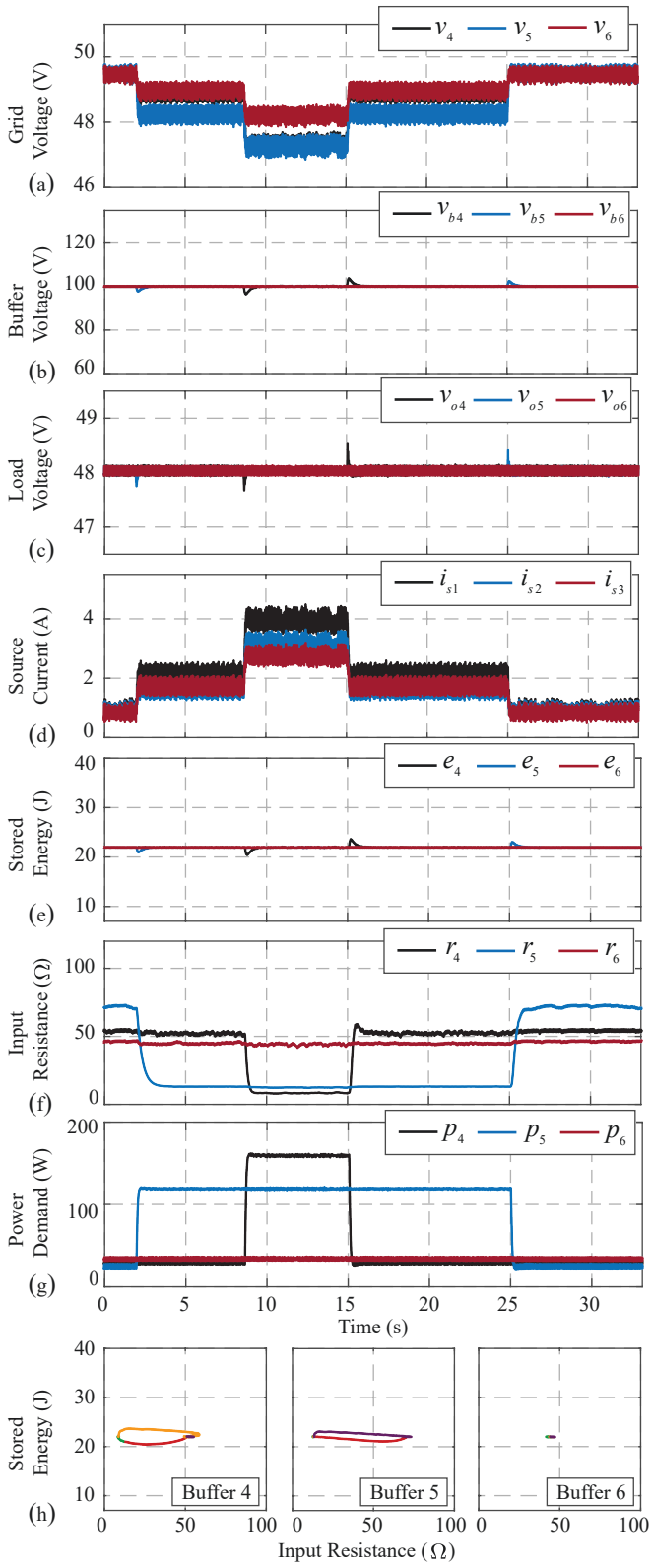

Fig. 7. Microgrid performance in response to two load changes at terminal 5 and terminal 6 with deactivated power buffers: (a) distribution bus voltages observed at the load terminals, (b) output voltage of the power buffers, (c) output voltage across the resistive loads, (d) source currents, (e) stored energy in power buffers, (f) input impedance of the power buffers, (g) output of the active loads, (h) energy-impedance trajectories of the power buffers.

both active loads 5 and 6 help smooth the transients. As shown in Fig. 8(e), once active load 5 abruptly changes, the stored energy of the active load 4 changes according to the assistive control law. The same happens to the stored energies of active loads 5 and 6, after the active load 4 changes. Red curves in Fig. 8(h) show impedance-energy trajectories during the first event, the green curves show those trajectories after the second event. In both cases, assistance is provided by dropping the stored energy and increasing the input impedance of the corresponding buffer. Orange and violet curves in Fig. 8(h) refer to the third and fourth load change events, respectively. Energy-impedance trajectories of buffers 4 and 5 go back to their initial points. Violet trajectory of buffer 4, and orange trajectories of buffers 5 and 6, denote how the stored energy and impedance exhibit smaller variations. This asymmetric behavior is due to the non-linearity of the control law as well as the choice of weighting terms. As shown in Fig. 8(a), Fig. 8(d), and Fig. 8(g), the group action of power buffers smooth, respectively, the input bus voltages, source currents, and power demands.

Remark 3. Communication delays affect the performance and stability of distributed control architectures. Herein, we have considered delay values in the range of the most commonly-used transmission technologies. Delay effects are negligible since the controller dynamics is slower then the the dynamics of the communication network, in line with the analysis done in [6], [12]. In fact, Fig. 6 and Fig. 8 (after the controller is triggered) show controller dynamics in the order of 250 ms or higher. More rigorously, theoretical stability analysis for nonlinear interconnected systems with time delays could be done using Lyapunov-Krasovskii or Lyapunov-Razumikhin approaches [47]. These methods would need suitable Lyapunov functions for each global system defined according to the loading grid used in the learning phase. Constructing such functions presents a more challenging task when compared with delay-free systems [48], and stability results are available only for certain classes of nonlinear systems [49]. Most control work in microgrids, that consider time delays, have relied on linear analysis [50]–[54], quasi-linear analysis, [55], or feedback-linearizing control structures [56].

A comparison with the distributed algorithm presented in [12] is shown in Fig. 9. At $t = 0.7s$, buffer 4 observes an abrupt change of its load from 80Ω to 15Ω. Using ADP, the energy stored in the buffer 4 recovers faster, as seen in Fig. 9(a). The energies stored in buffers 5 and 6 show higher deviation with comparable (active load 6) or slower (active load 5) settling times. This shows how the proposed method penalizes the individual action of the active load 4, enhancing the collective assistance provided by the active loads 5 and 6. Power demands are kept smooth, with a smaller initial derivative, as seen in Fig. 9(b). A faster dynamic response could be attained by adjusting the control gains in [12]. Therein, the controller design was based on a small-signal approximation of power buffers, making the controller valid only for a single operating point without guaranteeing its performance for larger load variations. By contrast, the method proposed here is based on a nonlinear formulation of the microgrid in (12). So long as

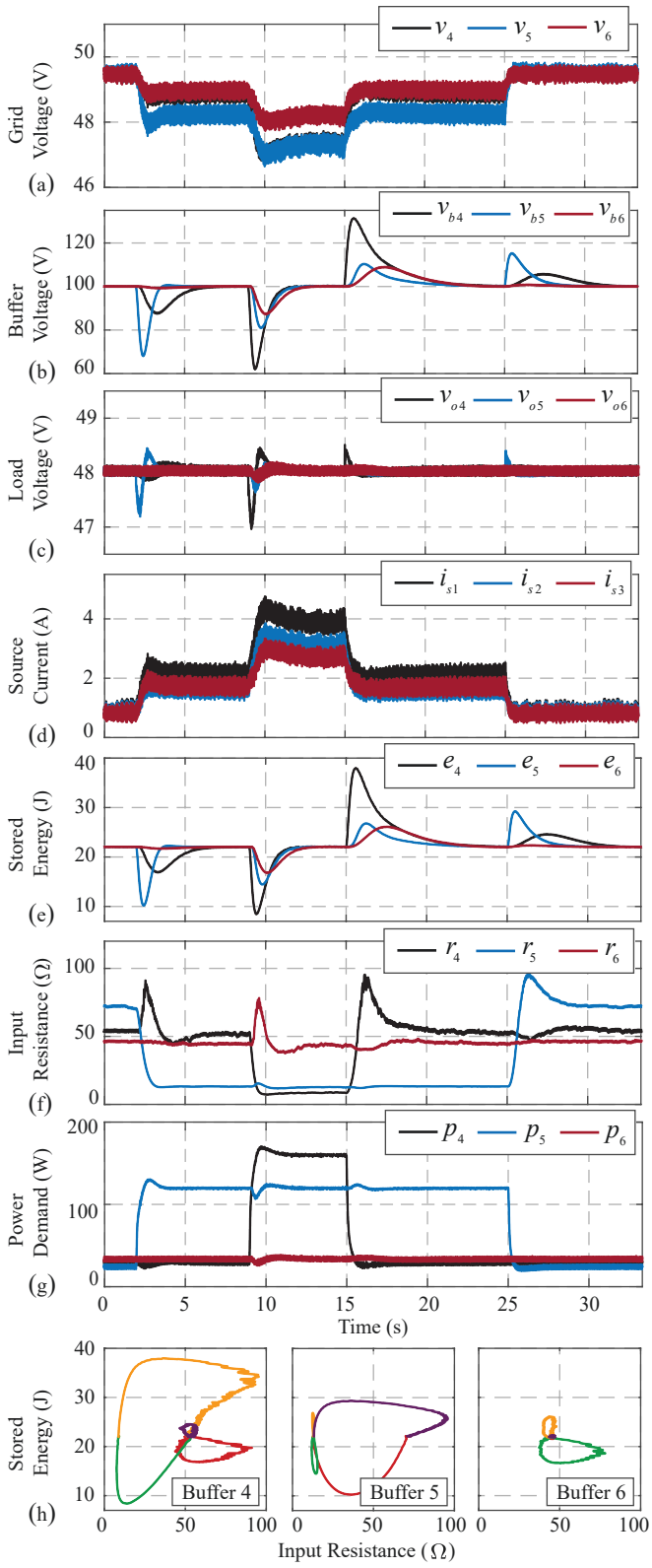

Fig. 8. Microgrid performance in response to two load changes at terminals 5 and 6 with activated power buffers: (a) distribution bus voltages observed at the load terminals, (b) output voltage of the power buffers, (c) output voltage across the resistive loads, (d) source currents, (e) stored energy in power buffers, (f) input impedance of the power buffers, (g) output of the active loads, and (h) energy-impedance trajectories of the power buffers.

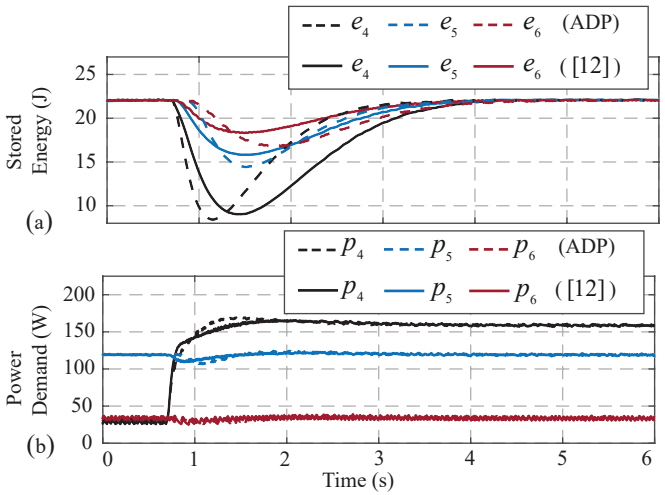

Fig. 9. Proposed controller performances against the linear controller in [12].

the learning phase spans a sufficiently-large loading space, and the PE condition is valid, controller stability is guaranteed for higher deviations. Finally, individual controllers are designed through Algorithm 3 for each load variation. The optimal control formulation guarantees semi-optimal performance in every learned scenario.

V. Conclusion

The stability properties of DC microgrids can be improved by augmenting volatile loads with a power buffer. Proper impedance-energy adjustment of such buffers can smooth transients during abrupt load changes. In this work, a communication network collectively groups the power buffers. To guarantee both performances and stability over a wide range of load variations, the full nonlinear dynamics of the power buffers and the distribution network has been taken into account. An assistive control law is designed using an adaptive dynamic programming algorithm with off-policy learning. The resulting control law is computed off-line and triggered at each load change, making the proposed approach particularly suitable for constrained IoT devices. Finally, hardware-in-the-loop simulations validates the proposed approach.

References

[1] M. Hamzeh, M. Ghafouri, H. Karimi, K. Sheshyekani, and J. M. Guerrero, "Power oscillations damping in DC microgrids," *IEEE Trans. Energy Convers.*, vol. 31, pp. 970–980, Sep. 2016.

[2] S. Sanchez and M. Molinas, "Degree of influence of system states transition on the stability of a DC microgrid," *IEEE Trans. Smart Grid*, vol. 5, pp. 2535–2542, Sep. 2014.

[3] D. Logue and P. T. Krein, "The power buffer concept for utility load decoupling," in *Proc. IEEE 31st Annual Power Electronics Specialists Conf.*, 2000, pp. 973–978.

[4] W. W. Weaver and P. T. Krein, "Mitigation of power system collapse through active dynamic buffers," in *Proc. IEEE 35th Annual Power Electronics Specialists Conf.*, 2004, pp. 1080–1084.

[5] W. Weaver and P. T. Krein, "Game-theoretic control of small-scale power systems," *IEEE Trans. Power Del.*, vol. 24, pp. 1560–1567, Jul. 2009.

[6] 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.

[7] N. C. Ekneligoda and W. W. Weaver, "Game-theoretic cold-start transient optimization in DC microgrids," *IEEE Trans. Ind. Electron.*, vol. 61, pp. 6681–6690, Dec. 2014.

[8] A. M. Dissanayake and N. C. Ekneligoda, "Online game theoretic feedback control of DC microgrids," in *Proc. IEEE Power & Energy Society Innovative Smart Grid Technologies Conf.*, 2018.

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

[10] W. W. Weaver, "Dynamic energy resource control of power electronics in local area power networks," *IEEE Trans. Power Electron.*, vol. 26, pp. 852–859, Mar. 2011.

[11] B. Banerjee and W. W. Weaver, "Generalized geometric control manifolds of power converters in a DC microgrid," *IEEE Trans. Energy Convers.*, vol. 29, pp. 904–912, Dec. 2014.

[12] 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.

[13] Y. Kuriki and T. Namerikawa, "Experimental validation of cooperative formation control with collision avoidance for a multi-UAV system," in *Proc. IEEE 6th Conf. on Automation, Robotics and Applications (ICARA)*. IEEE, 2015.

[14] J. Hu and Z. Xu, "Distributed cooperative control for deployment and task allocation of unmanned aerial vehicle networks," *IET Control Theory & Applications*, vol. 7, no. 11, pp. 1574–1582, 2013.

[15] L. Jin, S. Li, L. Xiao, R. Lu, and B. Liao, "Cooperative motion generation in a distributed network of redundant robot manipulators with noises," *IEEE Trans. on Syst., Man, and Cybern.*, vol. 48, no. 10, pp. 1715–1724, 2018.

[16] S. Li, H. Du, and P. Shi, "Distributed attitude control for multiple spacecraft with communication delays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 50, no. 3, pp. 1765–1773, 2014.

[17] V. Nasirian, S. Moayedi, A. Davoudi, and F. L. Lewis, "Distributed cooperative control of DC microgrids," *IEEE Trans. Power Electron.*, vol. 30, no. 4, pp. 2288–2303, 2015.

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

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

[20] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand, 1992, pp. 493–525.

[21] 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.

[22] 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.

[23] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Lear. Syst.*, vol. 25, no. 3, pp. 621–634, Mar. 2014.

[24] Q. Wei, F. L. Lewis, G. Shi, and R. Song, "Error-tolerant iterative adaptive dynamic programming for optimal renewable home energy scheduling and battery management," *IEEE Trans. Ind. Electron.*, vol. 64, no. 12, pp. 9527–9537, Dec. 2017.

[25] G. K. Venayagamoorthy, R. K. Sharma, P. K. Gautam, and A. Ahmadi, "Dynamic energy management system for a smart microgrid," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1643–1656, Aug. 2016.

[26] M. Boaro, D. Fuselli, F. D. Angelis, D. Liu, Q. Wei, and F. Piazza, "Adaptive dynamic programming algorithm for renewable energy scheduling and battery management," *Cogn. Comput.*, vol. 5, no. 2, pp. 264–277, Sep. 2012.

[27] D. Fuselli, F. D. Angelis, M. Boaro, S. Squartini, Q. Wei, D. Liu, and F. Piazza, "Action dependent heuristic dynamic programming for home energy resource scheduling," *Intern. Journal of Electr. Power & Energy Syst.*, vol. 48, pp. 148–160, Jun. 2013.

[28] S. Xie, W. Zhong, K. Xie, R. Yu, and Y. Zhang, "Fair energy scheduling for vehicle-to-grid networks using adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 8, pp. 1697–1707, Aug. 2016.

[29] Q. Wei, G. Shi, R. Song, and Y. Liu, "Adaptive dynamic programming-based optimal control scheme for energy storage systems with solar renewable energy," *IEEE Trans. Ind. Electron.*, vol. 64, no. 7, pp. 5468–5478, Jul. 2017.

[30] Y. Tang, H. He, J. Wen, and J. Liu, "Power system stability control for a wind farm based on adaptive dynamic programming," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 166–177, Jan. 2015.

[31] T. Bian, Y. Jiang, and Z.-P. Jiang, "Decentralized adaptive optimal control of large-scale systems with application to power systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2439–2447, Apr. 2015.

[32] A. M. Dissanayake and N. C. Ekneligoda, "Droop free optimal feedback control of distributed generators in islanded DC microgrids," *IEEE Trans. Emerg. Sel. Topics Power Electron.*, 2019.

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

[34] F. Dorfler and F. Bullo, "Kron reduction of graphs with applications to electrical networks," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 60, pp. 150–163, Jan. 2013.

[35] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, 2005.

[36] H. Modares and F. L. Lewis, "Optimal tracking control of nonlinear partially-unknown constrained-input systems using integral reinforcement learning," *Automatica*, vol. 50, no. 7, pp. 1780–1792, 2014.

[37] B. Luo, H.-N. Wu, T. Huang, and D. Liu, "Reinforcement learning solution for HJB equation arising in constrained optimal control problem," *Neural Networks*, vol. 71, pp. 150–158, 2015.

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

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

[40] M. Krsti, I. Kanellakopoulos, and V. Petar, *Nonlinear and adaptive control design*. Wiley New York, 1995.

[41] D. Xu, M. Chiang, and J. Rexford, "Link-state routing with hop-by-hop forwarding can achieve optimal traffic engineering," *IEEE/ACM Trans. Netw.*, vol. 19, pp. 1717–1730, Dec. 2011.

[42] D. Liberzon and A. S. Morse, "Basic problems in stability and design of switched systems," *IEEE Control Syst. Mag.*, vol. 19, pp. 59–70, Oct. 1999.

[43] M. Collotta and G. Pau, "A novel energy management approach for smart homes using bluetooth low energy," *IEEE J. Sel. Areas Commun.*, vol. 33, pp. 2988–2996, Dec. 2015.

[44] C. J. Hansen, "Internetworking with bluetooth low energy," *GetMobile: Mobile Comp. Comm.*, vol. 19, pp. 34–38, Aug. 2015.

[45] R. Rondón, M. Gidlund, and K. Landernäs, "Evaluating bluetooth low energy suitability for time-critical industrial IoT applications," *Int. Journal of Wireless Information Netw.*, vol. 24, pp. 278–290, May 2017.

[46] T. K. Refaat, R. M. Daoud, H. H. Amer, and E. A. Makled, "WiFi implementation of wireless networked control systems," in *Proc. ASM 7th Int. Conf. on Networked Sensing Systems*, 2010.

[47] K. Gu and S.-I. Niculescu, "Survey on recent results in the stability and control of time-delay systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 125, no. 2, pp. 158–165, Jun. 2003.

[48] S. Tiwari and Y. Wang, "Razumikhin-type small-gain theorems for large-scale systems with delays," in *49th IEEE Conference on Decision and Control*. IEEE, Dec. 2010.

[49] E. Fridman, "Tutorial on lyapunov-based methods for time-delay systems," *European Journal of Control*, vol. 20, no. 6, pp. 271–283, Nov. 2014.

[50] D. H. Nguyen and J. Khazaei, "Multiagent time-delayed fast consensus design for distributed battery energy storage systems," *IEEE Trans. Sustainable Energy*, vol. 9, no. 3, pp. 1397–1406, 2018.

[51] C. Dong, H. Jia, Q. Xu, J. Xiao, Y. Xu, P. Tu, P. Lin, X. Li, and P. Wang, "Time-delay stability analysis for hybrid energy storage system with hierarchical control in DC microgrids," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6633–6645, 2018.

[52] S. Liu, X. Wang, and P. X. Liu, "Impact of communication delays on secondary frequency control in an islanded microgrid," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2021–2031, Apr. 2015.

[53] C. Dou, D. Yue, J. M. Guerrero, X. Xie, and S. Hu, "Multiagent system-based distributed coordinated control for radial DC microgrid considering transmission time delays," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2370–2381, Sep. 2017.

[54] H. Yan, X. Zhou, H. Zhang, F. Yang, and Z.-G. Wu, "A novel sliding mode estimation for microgrid control with communication time delays," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 1509–1520, Mar. 2019.

[55] S. Kotpalliwar, S. Satpute, S. Meshram, F. Kazi, and N. Singh, "Modelling and stability of time-delayed microgrid systems," *IFAC-PapersOnLine*, vol. 48, no. 30, pp. 294–299, 2015.

[56] R. Zhang and B. Hredzak, "Nonlinear sliding mode and distributed control of battery energy storage and photovoltaic systems in AC microgrids with communication delays," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5149–5160, Sep. 2019.