Distributed Constrained Policy Learning for Power Management of Networked Microgrids

Qianzhi Zhang, Student Member, IEEE, Kaveh Dehghanpour, Member, IEEE, Zhaoyu Wang, Member, IEEE, Feng Qiu, Senior Member, IEEE, and Dongbo Zhao, Senior Member, IEEE

Abstract—This paper presents a multi-agent constrained reinforcement learning (RL) policy gradient method for optimal power management of networked microgrids (MGs) in distribution systems. While conventional RL algorithms are black box decision models that could fail to satisfy grid operational constraints, our proposed RL technique is constrained by AC power flow equations and other operational limits. Accordingly, the training process employs the gradient information of the power management problem constraints to ensure that the optimal control policy functions generate feasible decisions. Furthermore, we have proposed a distributed primal-dual consensus-based training approach for the RL solver to maintain the privacy of MGs’ control policies. After training, the learned optimal policy functions can be safely used by the MGs to dispatch their local resources, without the need to solve a complex optimization problem from scratch. Numerical experiments have been devised to verify the performance of the proposed method.

Index Terms—Constrained reinforcement learning, distributed optimization, networked microgrids, power management, policy gradient.

I. INTRODUCTION

NETWORKED microgrids (MGs) can offer various benefits, including higher perpetration of local distributed energy resources (DERs), improved controllability, and enhancement of power system resilience and reliability [1], [2]. Solving the power management problem of networked MGs is a complex task. While previous works in this area have provided valuable insight, we have identified two shortcomings in the literature:

(1) Limitations of model-based optimization methods: In the existing literature, there are quite a few model-based methods for solving the optimal power management problem of networked MGs, such as centralized decision models [3], [4] and distributed control frameworks [5]–[7]. However, with increasing number of MGs in distribution networks, these methods have to solve large-scale optimization problems with numerous nonlinear constraints that incur high computational costs and hinder real-time decision making. Furthermore, model-based methods are unable to adapt to the continuously evolving system conditions, as they need to re-solve the problem at each time step.

(2) Potential infeasibility of model-free machine learning methods: To address the limitations of model-based methods, model-free reinforcement learning (RL) techniques have been used to learn optimal power management policies through repeated interactions between a control agent and its environment. This approach eliminates the need to solve a large-scale optimization problem at each time point and enables the control agent to provide adaptive response to time-varying system states. Existing examples of RL application in power systems include economic dispatch and energy consumption scheduling of individual MGs [8], [9], online energy management of buildings [10] and multi-area smart control of generation in interconnected power grids [11]. In previous RL methods, control agents have been designed to train black box functions to approximate the optimal solutions through trial and error. However, the trained black box functions can fail to satisfy critical operational constraints, such as nodal voltage and capacity limits, since these constraints have not been encoded in the training process. This can lead to unsafe operational states and control action infeasibility.

Inspired by recent advances in safe RL algorithms [12]–[14] and to address the shortcomings in the existing literature, we have cast the power management problem of networked MGs as a distributed constrained Markov decision process (DCMDP). Moreover, we have proposed a multi-agent RL-based policy gradient solution strategy to optimize the control policies of the networked MGs. The proposed method introduces a trade-off between model-free and model-based methods and combines the benefits offered by both sides. Hence, on one hand, MGs’ power management policy functions are modeled using black box Deep Neural Networks (DNNs); while on the other hand, to ensure decision feasibility, a constrained back-propagation training method is proposed that exploits the gradients of the constraints and objective functions of the power management problem. The training process employs these gradient factors to provide a quadratically constrained linear program (QCLP) approximation to the power management problem at each back-propagation iteration. This enables the proposed method to be both adaptable to changes in the inputs of the black box components, and feasible with respect to operational constraints, including AC power flow. Finally, a distributed consensus-based primal-dual optimization method is proposed to decompose the training task among MG agents [15]. The distributed computation offers two advantages: (i) maintaining the privacy of MG control policies, and (ii) enhancing computational efficiency as the number of learning parameters grow.

The general framework of the proposed constrained RL-based method is shown in Fig. 1. The micro-sources within
each MG is controlled by an agent that adopts a private control policy, \( \pi \). Here, an MG’s control policy is defined by a parametric probability distribution function over the dispatching action space, with parameters \( \Theta \). MG agents receive the observed variables from the grid, including nodal voltages \( V(t) \) and injection currents \( I(t) \), which are used to determine gradients of problem constraints and objectives w.r.t. to learning parameters, \( \nabla_{\Theta} J \), and update the MGs’ control policy functions. The parameter update process is a distributed constrained back-propagation-based algorithm, which employs local inter-MG communication to satisfy global operational constraints through exchanging and processing dual Lagrangian variables, \( \lambda(t) \). The learned control policy function of the \( n' \)th MG agent is used to determine the MG’s control actions, \( a_n(t) \), under the input state variables, \( S_n(t) \), defined by aggregate load and solar irradiance. The agent’s control actions are active/reactive power dispatching signals for local diesel generators (DGs), energy storage system (ESS) and solar Photo-Voltaic (PV) panels. Theoretical analysis and numerical simulations are conducted to show that the proposed policy gradient algorithm can maximize the MG agents’ payoffs and satisfy operational constraints.

II. CONSTRAINED POLICY LEARNING FOR POWER MANAGEMENT OF NETWORKED MGs

A. Optimal Power Management Problem of Networked MGs

The original centralized formulation for optimal power management of networked MGs is shown in (1)-(14). We assume that each MG consists of local DGs, ESS, solar PV panels and a number of loads. This optimization problem is solved over a moving look-ahead decision window \( t' \in [t, t + T] \), using the latest estimations of solar and load power at different instants. Here, \( n \) is the MG index \((n \in \{1, \ldots, N\})\), \( i \) and \( j \) define the node numbers \((\forall i, j \in \Omega)\), and \( l \) denotes the network branch index \((\forall l \in \mathcal{L})\). Note that the vectors are denoted in bold letters throughout the paper.

The objective function (1), with control action vector \([p_{DG}^i, p_{Ch}^i, p_{Dis}^i, q_{PV}^i, q_{ESS}^i] \in \mathcal{X}_p, \mathcal{X}_q\) minimizes MGs’ total cost of operation, which is composed of the income/cost from power transfer with the grid and cost of running local DG. Here, \( \lambda_{Fuel}^P \) is the DG fuel price in \$/L, \( \lambda_{R}^F \) is the electricity price in \$/kWh, and \( F_{n,t}^{PCC} \) is active power transfer between grid and the \( n' \)th MG at the point of common coupling (PCC). The fuel consumption of DG, \( F_{i,n,t}' \), can be expressed as a quadratic polynomial function of its power, \( P_{DG,i,n,t}' \), with parameters, \( a_{n,t}' \), \( b_{n,t}' \), and \( c_{n,t}' \), as shown in (2).

\[
\min_{x_p, x_q} \sum_{n=1}^{N} \sum_{t'=t}^{t+T} (-\lambda_{R}^F p_{DG,i,n,t}')^2 + \lambda_{Fuel}^P p_{DG,i,n,t}') + c_{n,t}'
\]

The problem’s global constraints are defined by the limits for bus voltage amplitudes of all nodes in the network, \([V_i^m, V_i^M]\), and the maximum permissible current magnitudes \( I_{ij}^M \) of the branches, as shown below:

\[
V_i^m \leq V_i \leq V_i^M \tag{3}
\]

\[
-I_{ij}^M \leq I_{ij} \leq I_{ij}^M \tag{4}
\]

Constraints (5)-(14) are the local constraints for the \( n' \)th MG. Constraints (5) ensure that the DG active/reactive power outputs, \( P_{DG,i,n,t}' / Q_{DG,i,n,t}' \), are within the DG power capacity \( F_{DG,i,n,t}' / Q_{DG,i,n,t}' \), and (6) enforces the maximum DG ramp limit, \( p_{DG,R}^i \). Reactive power output, \( Q_{PV,i,n,t}' \), is constrained by its maximum limit \( Q_{PV,i,n,t}' \) per (8). The operational ESS constraints are described by (9)-(14), where (9) determines the state of charge (SOC) of ESSs, \( SOC_{i,n,t}' \), \( Q_{Cap}^i \) denotes the maximum capacity of ESSs. To ensure safe ESS operation, the SOC and charging/discharging power of ESS, \( p_{Ch,i,n,t}' / p_{Dis,i,n,t}' \) are constrained as shown in (10)-(14). Here, \([SOC_{i,n,t}' \leq SOC_{i,n,t}^M] \), \( p_{Ch,i,n,t}' \) and \( p_{Dis,i,n,t}' \) define the permissible range of SOC, and maximum charging and discharging power, respectively. Constraint (13) indicates that ESSs cannot charge and discharge at the same time instant. And \( \eta_{Ch} / \eta_{Dis} \) represents the charging/discharging efficiency. The reactive power of ESS, \( Q_{ESS,i,n,t}' \), is kept within maximum limit, \( Q_{ESS,M,i,n,t}' \), through constraint (14).

\[
0 \leq p_{DG,i,n,t}' \leq F_{DG,i,n,t}' \tag{5}
\]

\[
0 \leq Q_{DG,i,n,t}' \leq F_{DG,i,n,t}' \tag{6}
\]

\[
|p_{DG,i,n,t}' - p_{DG,i,n,t}' - 1| \leq p_{DG,R}^i \tag{7}
\]

\[
|Q_{PV,i,n,t}'| \leq Q_{PV,i,n,t}' \tag{8}
\]

\[
SOC_{i,n,t}' = SOC_{i,n,t}' - 1 + \Delta t \left( \frac{p_{Ch,i,n,t}' \eta_{Ch} - p_{Dis,i,n,t}' / \eta_{Dis}}{Q_{Cap}^i} \right) \tag{9}
\]

\[
SOC_{i,n,t}' \leq SOC_{i,n,t}' \leq SOC_{i,n,t}^M \tag{10}
\]

\[
0 \leq p_{Ch,i,n,t}' \leq F_{Ch,i,n,t}' \tag{11}
\]

\[
0 \leq p_{Dis,i,n,t}' \leq F_{Dis,i,n,t}' \tag{12}
\]

\[
p_{Ch,i,n,t}' p_{Dis,i,n,t}' = 0 \tag{13}
\]

\[
|Q_{ESS,i,n,t}'| \leq Q_{ESS,M,i,n,t}' \tag{14}
\]

Note that the global constraints are implicitly determined by the network AC power flow equations, which will be used to calculate the gradient factors of objective (1) and constraints (3)-(14) w.r.t. learning parameters as shown in Section II-D.
B. Distributed Constrained Markov Decision Process

In this section, the optimal power management of networked MGs is transformed into a DCMDP. The purpose of the DCMDP is to provide a framework for control agents to collaboratively find control policies to maximize their total accumulated reward while satisfying all problem constraints. To do this, we have proposed a multi-agent RL methodology to ensure that the outcome of the DCMDP also corresponds to the solution of (1)-(14). To show this, first we provide a description of the components of the DCMDP.

1) Control agents: The problem consists of $N$ autonomous control agents, where each agent is in charge of dispatching the resources within an individual MG. The MGs are collaborative, in the sense that they depend on local communication with each other to optimize their behaviors.

2) State variable set: The DCMDP state vector for the $n$'th MG agent at time $t$ is defined as $S_{n,t}$ over the time window $[t, t + T]$:

$$S_{n,t} = \left[ \hat{P}^{PV}_{n,t}, \hat{P}^{D}_{n,t} \right]_{t=t}^{t+T}$$

where, $\hat{P}^{PV}_{n,t}$ and $\hat{P}^{D}_{n,t}$ are the vectors of estimated aggregate solar irradiance and aggregate internal load power of the $n$'th MG at time $t$, respectively.

3) Control action set: The control action vector for the $n$'th agent at time $t$ is denoted as $A_{n,t} \in \mathbb{R}^{D_n}$ and consists of the dispatching decision variables for the $n$'th MG over the time window $[t, t + T]$:

$$A_{n,t} = [P_{D}^{n,t}, P_{C}^{Ch}, P_{C}^{Dis}, P_{Sw}^{agine}, Q_{D}^{n,t}, Q_{PV}^{n,t}, Q_{ESS}^{n,t}]_{t=t}^{t+T}$$

4) Observation variable set: The observation vector for the agents at time $t$ is denoted as $O_t$, and includes grid’s nodal voltages $V_t$ and current injections $I_t$ at that time. $O_t = [V_t, I_t]$. Note that the observations are implicitly determined by the agents’ control actions, and thus, cannot be predicted independently of the agents’ policies. However, unlike the observation variables, the state variables are independent of the agents’ control actions and can be predicted for the whole decision window without the need to consider agents’ policies [13]. In the power management problem, nodal sensors or distribution grid’s state estimation module will provide the latest values of observations.

5) Control policy: In this paper, the control policy for the $n$'th agent, denoted as $\pi_n$, is defined as a parameterized $D_n$-dimensional multivariate Gaussian distribution function over $A_{n,t}$, which determines the probability of the agent’s optimal control action after training, $a^*_n$:

$$a^*_n \sim \pi_n(S_{n,t}|\theta_n) = \frac{1}{\sqrt{|\Sigma_n|} \sqrt{2\pi}^D_n} e^{-\frac{1}{2}(a_n-\mu_n)^T \Sigma_n^{-1}(a_n-\mu_n)}$$

where, $\mu_n \in \mathbb{R}^{D_n \times 1}$ is the mean vector and $\Sigma_n \in \mathbb{R}^{D_n \times D_n}$ is the covariance matrix of multivariate normal Gaussian distribution for the $n$'th agent. The parameter vector of the $n$'th agent, $\theta_n$, consists of two parameter subsets $\theta_{\mu_n}$ and $\theta_{\Sigma_n}$ that parameterize the mean vector and the covariance matrix of the agent’s policy function. To do this, two DNNs are used for each MG agent as parametric learning functions to represent agents’ control policy components. These DNNs receive the local states as input and output the mean vectors and covariance matrices of the agents’ control policy functions. Hence, $\theta_{\mu_n}$ and $\theta_{\Sigma_n}$ are the weights and biases of these DNNs, as shown below:

$$\mu_n = DNN(S_{n,t}|\theta_{\mu_n})$$

$$\Sigma_n = DNN(S_{n,t}|\theta_{\Sigma_n})$$

The set of control policies and learning parameters of all agents are denoted by $\pi = \{\pi_1, ..., \pi_N\}$ and $\theta = \{\theta_1, ..., \theta_N\}$, respectively.

6) Reward function: The reward function at time $t$ for the $n$'th MG is defined as the negative accumulated operational cost of networked MGs over the decision window, obtained from the objective function of the power management problem, $\Pi$, as follows:

$$J_{R_n}(\pi_n) = E_{\pi_n} \left[ \sum_{t'=t}^{t+T} \left( \lambda R_{t',C}^F \right) - \lambda R_{t',P}^F \right]$$

where, $E_{\pi_n}\{\}$ is the expectation operation over the control policy of the agent, to take into account the inherent uncertainty of state and observation variables.

7) Constraint return: The DCMDP consists of a total of $M$ constraints, including $M^L$ local and $M^G$ global constraints, defined by (3)-(14), respectively, and denoted as $C_m \leq d_m, \forall m \in \{1, ..., M\}$, where $d_m$ is the upper-boundary of the $m$’th constraint. Note that equality constraint [13] can be transformed into two inequality constraints. Constraint satisfaction is encoded into the DCMCP using the constraint return values of agents’ policies $\pi$ as:

$$J_{C_m}(\pi) = E_{\pi_n}(C_m) \leq d_m, \forall m \in \{1, ..., M\}$$

where, expectation operation has been leveraged in [21] to handle state and observation uncertainties.

C. Constrained Policy Gradient Formulation

Given the definitions of the components of the DCMDP (Section II-B), the power management problem of the networked MGs is cast as an iterative policy gradient problem, where the control policy of the agents are updated at time $t$ by maximizing the reward function around the previous policies, while satisfying constraint return criteria:

$$\pi_{t+1} = \arg \max_{\pi_1, ..., \pi_N} \sum_{n=1}^{N} J_{R_n}(\pi_n)$$

$$s.t. \quad J_{C_m}(\pi) \leq d_m, \forall m \in \{1, ..., M\}$$

$$\Delta(\pi_n, \pi_{n-1}) \leq \delta, \forall n$$

where, $\Delta(\cdot, \cdot)$ is the KL-divergence function [12] that serves as a distance measure between the previous policy, $\pi_{n-1}$, and the updated policy, $\pi_{n+1}$, and is constrained by a step size, $\delta$. Note that [24] ensures that consecutive policies are within close distance from each other.

It has been shown in [12] that using a trust region method the policy gradient problem can be transformed into an iterative QCLP through Taylor expansion. This is done by
leveraging the linear approximations of the objective and constraint returns around $\theta^l$:

$$\theta^{l+1} = \arg\max_{\theta_1, ..., \theta_N} \sum_{n=1}^{N} g_n^T (\theta_n - \theta^l_n)$$  \hspace{1cm} (25)

subject to

$$J_{cm}(\theta^l) + b_n^T (\theta - \theta^l) \leq d_m, \ \forall m$$  \hspace{1cm} (26)

$$\frac{1}{2}(\theta_n - \theta^l_n)^T H_n(\theta_n - \theta^l_n) \leq \delta, \ \forall n$$  \hspace{1cm} (27)

where, $g_n = \nabla_{\theta} J_R$ and $b_n = \nabla_{\theta} J_{cm}$ are the gradient factors of the reward and constraint return functions w.r.t. the learning parameters. Constraint (24) is transformed into (27) using the Fisher Information Matrix (FIM) of the policy functions, $\pi_n$, denoted by $H_n$. The $(c,d)'th$ entry of the FIM for policy functions with Gaussian structure is determined as follows [16]:

$$H_n(c,d) = \varepsilon \left( \frac{\partial \log p(X_n, \theta_n)}{\partial \theta_n(c)} \frac{\partial \log p(X_n, \theta_n)}{\partial \theta_n(d)} \right)$$

$$= 2 \left( \frac{\partial \mu_n^H}{\partial \theta_n(c)} (\Sigma_n^{-1} \frac{\partial \mu_n}{\partial \theta_n(d)}) + \text{Tr} \left\{ \Sigma_n^{-1} \frac{\partial \Sigma_n}{\partial \theta_n(c)} \Sigma_n^{-1} \frac{\partial \Sigma_n}{\partial \theta_n(d)} \right\} \right)$$  \hspace{1cm} (28)

Note that (25)–(27) provides a constrained gradient-based method for training the policy functions of the MG agents. Given that a Gaussian structure has been selected for representing the policy functions, (17), the gradient factors have to be determined considering the two sets of learning parameters, $[\theta_{g_n}, \theta_{s_n}]$. This process is outlined in Section II-D.

### D. Gradient Factor Determination

To determine gradient factors, the following information are used: (i) observation variables, $O_k$, including nodal voltages $V$ and current injections $I$; (ii) the latest system states $S_k$ for each MG agent; (iii) the latest control actions $a_{n,k}$ of each MG agent; (iv) the latest learning parameters $\theta_{g_n}$; (v) network parameters, including the nodal admittance matrix, $Y$. Using these information and chain rule, $g_n$ and $b_n$ in (25) and (26) can be written as:

$$g_{\mu_n} = \frac{\partial J_{Re}}{\partial \mu_n} \frac{\partial \pi_n}{\partial \mu_n} + \frac{\partial J_{Di}}{\partial \mu_n} \frac{\partial \pi_n}{\partial \mu_n}$$  \hspace{1cm} (29)

$$g_{s_n} = \frac{\partial J_{Re}}{\partial \pi_n} \frac{\partial \pi_n}{\partial \mu_n} + \frac{\partial J_{Di}}{\partial \pi_n} \frac{\partial \pi_n}{\partial \mu_n}$$  \hspace{1cm} (30)

where, each gradient factor, $g_{\mu_n}$, $g_{s_n}$, $b_{\mu_n}$, and $b_{s_n}$ consists of four elements. These elements are determined as follows:

1) $\partial J_{Re}/\partial a_n$ and $\partial J_{cm}/\partial a_n$: As can be seen in (5)-(14), the local constraint returns are trivial functions of the control actions. Hence, for these constraints, $\partial J_{cm}/\partial a_n$ can be directly calculated. For example, the constraint return value for (5) is $J_{cm} = P_{n,t}$ which induces a simple gradient element w.r.t. control action $P_{DG}^{n,t}$:

$$\frac{\partial J_{cm}}{\partial P_{DG}^{n,t}} = 1.$$  \hspace{1cm} (31)

| Table I | Derivations of $I_{Re}^{n,t}$ and $I_{Im}^{n,t}$ w.r.t. $a_n$ |
|--------|----------------------------------------------------------|
| $I_{Re}^{n,t}$ | $V_{Re}^{n,t}$ | $V_{Re}^{n,t}$ | $V_{Re}^{n,t}$ | $V_{Re}^{n,t}$ | $V_{Re}^{n,t}$ | $V_{Re}^{n,t}$ | $V_{Re}^{n,t}$ | $V_{Re}^{n,t}$ | $V_{Re}^{n,t}$ |
| $I_{Im}^{n,t}$ | $V_{Im}^{n,t}$ | $V_{Im}^{n,t}$ | $V_{Im}^{n,t}$ | $V_{Im}^{n,t}$ | $V_{Im}^{n,t}$ | $V_{Im}^{n,t}$ | $V_{Im}^{n,t}$ | $V_{Im}^{n,t}$ | $V_{Im}^{n,t}$ |

The gradients of returns w.r.t. control actions for the remaining constraints, (6)-(13), can be obtained in a similar way. The major difficulty in determining $\partial J_{Re}/\partial a_n$ and $\partial J_{cm}/\partial a_n$ pertains to the agents’ reward functions and global constraint returns, (1)-(4), which are only implicitly related to the control actions. Since the reward and all the global constraint returns are functions of the observation variables, $V$ and $I$, the gradients of these variables w.r.t. control actions are obtained and used to quantify $\partial J_{Re}/\partial a_n$ and $\partial J_{cm}/\partial a_n$. To do this, a four-step process is proposed that leverages the current injection-based AC power flow equations:

**Step 1** - First, the gradients of real and imaginary parts of nodal current injection w.r.t. control actions are derived (denoted as $\partial I_{Re}^{n,t}/\partial a_n$ and $\partial I_{Im}^{n,t}/\partial a_n$, respectively). To achieve this, the nodal power balance and nodal current injection relationships in the network are employed:

$$I_{Re}^{n,t} = \frac{p_{i,n,t} V_{Re}^{n,t} + q_{i,n,t} V_{Im}^{n,t} V_{Re}^{n,t}}{V_{i,t}^2}$$  \hspace{1cm} (32)

$$I_{Im}^{n,t} = \frac{p_{i,n,t} V_{Im}^{n,t} - q_{i,n,t} V_{Re}^{n,t}}{V_{i,t}^2}$$  \hspace{1cm} (33)

$$p_{i,n,t} = p_{DG}^{n,t} - p_{PV}^{n,t} - p_{Ch}^{n,t} - p_{Dis}^{n,t}$$  \hspace{1cm} (34)

$$q_{i,n,t} = q_{DG}^{n,t} - q_{PV}^{n,t} - q_{Ch}^{n,t} - q_{Dis}^{n,t}$$  \hspace{1cm} (35)

where, $I_{Re}^{n,t}$ and $I_{Im}^{n,t}$ denote the real and imaginary parts of nodal voltage and current injection at node $i$. Using these equations, $\partial I_{Re}^{n,t}/\partial a_n$ and $\partial I_{Im}^{n,t}/\partial a_n$ are derived and shown in Table I. Note that the entries of this table can be calculated using the real and imaginary parts of nodal voltages, which in practice are either measured or estimated [17].

**Step 2** - Using $\partial I_{Re}^{n,t}/\partial a_n$ and $\partial I_{Im}^{n,t}/\partial a_n$ from Step 1 (Table I), $\partial V_{Re}/\partial a_n$ and $\partial V_{Im}/\partial a_n$ are obtained employing the network-wide relationship between nodal voltages and current injections:

$$\begin{bmatrix} \partial V_{Re} / \partial a_n \\ \partial V_{Im} / \partial a_n \end{bmatrix} = \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix}^{-1} \begin{bmatrix} \partial I_{Re} / \partial a_n \\ \partial I_{Im} / \partial a_n \end{bmatrix}$$  \hspace{1cm} (36)

where, the modified network bus admittance sub-matrices are determined as follows:

$$Y_{11} = Y_{Re} - Y_{D}^{Re,Re}, \ Y_{12} = -Y_{Im} - Y_{D}^{Re,Im}$$  \hspace{1cm} (37)

$$Y_{21} = Y_{Im} - Y_{D}^{Im,Re}, \ Y_{22} = Y_{Re} - Y_{D}^{Im,Im}$$  \hspace{1cm} (38)

here, $Y_{Re}$ and $Y_{Im}$ are the real and imaginary parts of the original bus admittance matrix. The elements in diagonal
matrices $Y_{D}^{(Re,Re)}$, $Y_{D}^{(Re,Im)}$, $Y_{D}^{(Im,Re)}$ and $Y_{D}^{(Im,Im)}$ are calculated using the following equations \[17\]:

\[
Y_{D}^{(Re,Re)}(i,i) = \frac{p_{i,i} V_{i,i}^{4}}{i,i} - 2V_{i,i}^{Re}(p_{i,i} V_{i,i}^{Re} + q_{i,i} V_{i,i}^{Im})
\]

(39)

\[
Y_{D}^{(Re,Im)}(i,i) = \frac{q_{i,i} V_{i,i}^{4}}{i,i} - 2V_{i,i}^{Im}(p_{i,i} V_{i,i}^{Re} + q_{i,i} V_{i,i}^{Im})
\]

(40)

\[
Y_{D}^{(Im,Re)}(i,i) = -\frac{q_{i,i} V_{i,i}^{4}}{i,i} - 2V_{i,i}^{Re}(p_{i,i} V_{i,i}^{Re} + q_{i,i} V_{i,i}^{Im})
\]

(41)

\[
Y_{D}^{(Im,Im)}(i,i) = \frac{p_{i,i} V_{i,i}^{4}}{i,i} - 2V_{i,i}^{Im}(p_{i,i} V_{i,i}^{Re} + q_{i,i} V_{i,i}^{Im})
\]

(42)

**Step 3** - Noting that the current flow constraint returns and the rewards are also functions of branch current flows, the gradients of branch current flows are required to obtain $J_{Re}/\partial a_{n}$ and $J_{Im}/\partial a_{n}$. Using the branch current flow equations, these gradients are determined as a function of the derivatives of nodal voltages and current injections, as follows:

\[
\begin{align*}
\frac{\partial I_{ij,t}^{Re}}{\partial a_{n,t'}} &= \frac{\partial V_{i,t}^{Im}}{\partial a_{n,t'}} - \frac{\partial V_{i,t}^{Im}}{\partial a_{n,t'}} - \frac{\partial V_{i,t}^{Re}}{\partial a_{n,t'}} + \frac{\partial V_{i,t}^{Re}}{\partial a_{n,t'}} \\
\frac{\partial I_{ij,t}^{Im}}{\partial a_{n,t'}} &= \frac{\partial V_{i,t}^{Im}}{\partial a_{n,t'}} - \frac{\partial V_{i,t}^{Im}}{\partial a_{n,t'}} + \frac{\partial V_{i,t}^{Re}}{\partial a_{n,t'}} - \frac{\partial V_{i,t}^{Re}}{\partial a_{n,t'}}
\end{align*}
\]

(43)

(44)

where, $I_{ij,t}^{Re}$ and $I_{ij,t}^{Im}$ are the real and imaginary parts of branch currents, $V_{i,t}^{Re}$ and $V_{i,t}^{Im}$ are the real and imaginary parts of branch admittance.

**Step 4** - Finally, using the derivatives obtained from Steps 1, 2, and 3, $J_{Re}/\partial a_{n}$ and $J_{Im}/\partial a_{n}$ are determined through straightforward algebraic manipulations. As an example, the gradient of reward function w.r.t. $P_{D,n,t'}$ is calculated as:

\[
\frac{\partial J_{Re}}{\partial P_{D,n,t'}} = \sum_{t'=t}^{t+T} \left( \lambda_{F,n,t'} (2a_{f} + b_{f}) - \lambda_{Re,n,t'} \frac{\partial PCC_{n,t'}}{\partial P_{D,n,t'}} \right)
\]

(45)

where, $\partial PCC_{n,t'}/\partial P_{D,n,t'}$ is obtained using the outcomes of Steps 2 and 3, as follows:

\[
\begin{align*}
\frac{\partial PCC_{n,t'}}{\partial P_{D,n,t'}} &= \frac{\partial V_{i,t'}^{Re}}{\partial P_{D,n,t'}} V_{i,t'}^{Re} + \frac{\partial V_{i,t'}^{Im}}{\partial P_{D,n,t'}} V_{i,t'}^{Im} \\
&+ \frac{\partial V_{i,t'}^{Re}}{\partial P_{D,n,t'}} I_{ij,t'}^{Re} + \frac{\partial V_{i,t'}^{Im}}{\partial P_{D,n,t'}} I_{ij,t'}^{Im}
\end{align*}
\]

(46)

Furthermore, $J_{Im}/\partial a_{n}$ for the global constraints (3) and (4) can be calculated using the outcomes of Steps 2 and 3:

\[
\begin{align*}
\frac{\partial V_{i,t'}}{\partial a_{n,t'}} &= \frac{V_{i,t'}^{Re}}{\partial a_{n,t'}} + \frac{V_{i,t'}^{Im}}{\partial a_{n,t'}} \\
\frac{\partial I_{ij,t'}}{\partial a_{n,t'}} &= \frac{I_{ij,t'}^{Re}}{\partial a_{n,t'}} + \frac{I_{ij,t'}^{Im}}{\partial a_{n,t'}}
\end{align*}
\]

(47)

(48)

2) $\partial a_{n}/\partial \pi_{n}$: Using the latest values for parameters $\mu_{n}$, $\Sigma_{n}$, and actions $a_{n}$, the gradient of control actions w.r.t. $\pi_{n}$ is obtained from \[17\], as follows \[18\]:

\[
\frac{\partial a_{n}}{\partial \pi_{n}} = -\frac{\Sigma_{n}^{-1}(\alpha_{n} - \mu_{n})}{\sqrt{\Sigma_{n}((2\pi)^{D_{n}})}} e^{-\frac{1}{2} A}
\]

(49)

where, $A = (\alpha_{n} - \mu_{n})^{T} \Sigma_{n}^{-1}(\alpha_{n} - \mu_{n})$.

3) $\partial \pi_{n}/\partial \mu_{n}$ and $\partial \pi_{n}/\partial \Sigma_{n}$: using the latest values for parameters $\mu_{n}$, $\Sigma_{n}$ and actions $a_{n}$, the gradients of control policies, w.r.t. $\mu_{n}$ and $\Sigma_{n}$ are determined using \[17\]:

\[
\frac{\partial \pi_{n}}{\partial \mu_{n}} = \frac{\Sigma_{n}^{-1}(\alpha_{n} - \mu_{n})}{\sqrt{\Sigma_{n}((2\pi)^{D_{n}})}} e^{-\frac{1}{2} A}
\]

(50)

\[
\frac{\partial \pi_{n}}{\partial \Sigma_{n}} = -\frac{1}{2} \frac{\Sigma_{n}^{-1}(\alpha_{n} - \mu_{n})(\alpha_{n} - \mu_{n})^{T} \Sigma_{n}^{-1}}{\sqrt{\Sigma_{n}((2\pi)^{D_{n}})}} e^{-\frac{1}{2} A}
\]

(51)

4) $\partial \mu_{n}/\partial \theta_{\mu}$ and $\partial \Sigma_{n}/\partial \theta_{\Sigma}$: A back-propagation process \[19\] is performed on the two DNNs within each MG agent’s control policy function, \(13\) and \(19\), to determine the gradients of DNNs’ outputs w.r.t. their parameters. In each iteration, the latest values of DCMDP state variables are employed as inputs of the DNNs. The back-propagation process exploits a chain-rule-based method for stage-by-stage spreading of gradient information through layers of the DNNs, starting from the output layer and moving towards the input \[19\]. To enhance the stability of the back-propagation process, a sample batch approach is adopted, where the gradients obtained from several sampled actions are averaged to ensure robustness against outliers.

**III. DISTRIBUTED CONSENSUS-BASED RL POLICY GRADIENT**

Using the gradient factors \[29\] and \[30\], the QCLP, \(25\)-\(27\), is fully specified and can be solved at each policy update iteration for training the agents’ RL frameworks. However, we have identified two challenges for this problem: (i) the size of the decision variable $\theta$ can be extremely large, which results in high computational costs during training. (ii) The control policy privacy of the MG agents needs to be preserved during training, which implies that the agents might not have access to each others’ control policy parameters. Centralized solvers can be both time-consuming and lack guarantees for privacy maintenance.

In order to address these two challenges, we have developed a consensus-based distributed constrained optimization algorithm \[15\], which is both scalable and does not require sharing control policy parameters among agents. Thus, the proposed algorithm is able to efficiently solve the QCLP \(25\)-\(27\), while relying only on local inter-MG communication. The purpose of inter-MG interactions is to satisfy global constraints, \(3\) and \(4\). To do this, the agents repeatedly estimate and communicate dual variable $\lambda_{n}$, corresponding to the Lagrangian multiplier of global constraints. Furthermore, a local primal-dual gradient step is included in the algorithm to move the primal and dual parameters towards their global optimum. The proposed distributed algorithm consists of four stages that are performed iteratively, as follows:

**Stage I. Initialize ($k \leftarrow 1$):** Gradient factors $g_{n}$ and $b_{m}$ are obtained from Section II-D. The previous values of learning parameters are input to the QCLP, $\theta^{l} = 0 \leftarrow \theta^{l-1}$, Lagrangian multipliers are initialized as zero for each MG agent.

**Stage II. Weighted averaging operation:** MG agent $n$ receives the Lagrangian multiplier $\lambda_{n'}$ (for global constraints)
from its neighbouring MG agents \( n' \in \{1, ..., N_n \} \) and combines the received estimates using weighted averaging:

\[
\hat{\lambda}_n(k) = \sum_{n'=1}^{N_n} w_{n}(n') \lambda_{n'}(k) \tag{52}
\]

where, \( w_{n}(n') \) is the weight that MG agent \( n \) assigns to the incoming message of the neighbouring MG agent, \( n' \). To guarantee convergence to consensus, the weight matrix, composed of the agents’ weight parameters \( w_{n}(n') \), is selected as a doubly stochastic matrix \([15]\).

**Stage III. Primed gradient update:** The \( n' \)th MG agent updates its primal parameters \( \theta_{n'}^t \) employing a gradient descent operation, using the gradients of the agent’s reward and the global constraint returns, \( n' \in M^G \), and step size \( \rho_1 \):

\[
\bar{\theta}_n(k) = \theta_{n}^t(k) - \rho_1(g_n(\theta_{n}^t(k)) + b_m(\theta_{n}^t(k))\hat{\lambda}_n(k)) \tag{53}
\]

**Stage IV. Projection on local constraints:** The agent projects the local learning parameters to the feasible region defined by the gradients of the local constraints \([5]-[14]\):

\[
\theta_{n}^t(k + 1) = \arg \min \| \bar{\theta}_n(k) - \theta \| \tag{54}
\]

s.t. \( J_{c,n}(\theta_n(0)) + b_m^T(\theta_n(0) - \theta) \leq d_m, \forall m \in M^L \tag{55} \)

\[
\frac{1}{2}(\theta_{n}^t(0) - \theta)^TH_n(\theta_{n}^t(0) - \theta) \leq \delta, \forall n \tag{56}
\]

**Stage V. Dual gradient update:** Each agent’s estimations of dual variables \( \lambda_n \) for the global constraints, \([3] \) and \([4]\), will be updated using a gradient ascent process over \( \bar{\lambda}_n \):

\[
\lambda_n(k + 1) = [(\bar{\lambda}_n(k) + \rho_2(b_m\theta_n^t(k + 1) - d_m))]^+, \forall m' \in M^G \tag{57}
\]

where, \( \rho_2 \) is a penalty factor for global constraints violation, and the operator \([\cdot]^+\) returns the non-negative part of its input.

**Stage VI. Stopping criteria:** Check algorithm convergence using the changes of \( \theta_{n'}^t(k) \): stop when the changes in parameters falls below the threshold value \( \Delta \theta_n \); otherwise, go back to Stage II.

The overall flowchart of the RL policy gradient process using the proposed distributed consensus-based optimization algorithm is shown in Algorithm 1.

### Algorithm 1 Distributed Constrained Policy Learning

1. Select \( T, \rho_1, \rho_2 \)
2. Initialize \( \theta_{n}^0 \)
3. for \( t \leftarrow 1 \) to \( t_{\text{max}} \) do
4. \( S_{n} \leftarrow (S_n(t), ..., S_n(t + T)) \)
5. \( \mu_{n} \leftarrow [18] \) [Parameter Insertion]
6. \( \Sigma_{n} \leftarrow [19] \) [Parameter Insertion]
7. \( \alpha_n \sim \pi_n(S_n(\theta_{n}^t)) \leftarrow [17] \) [Action selection]
8. \( \partial J_{R_n}/\partial a_n \leftarrow (45) - (46) \)
9. \( \partial J_{C_{mn}}/\partial a_n \leftarrow (31), (47)-(48) \)
10. \( \partial a_n/\partial \pi_n \leftarrow (49) \)
11. \( \partial \pi_n/\partial \mu_n \leftarrow (50) \)
12. \( \partial \pi_n/\partial \Sigma_n \leftarrow (51) \)
13. \( \mu_{n}/\partial \theta_{n} \mu_{n} \rightarrow DNN_{\mu_{n}} \) [Back-propagation]
14. \( \Sigma_{n}/\partial \theta_{n} \Sigma_{n} \rightarrow DNN_{\Sigma_{n}} \) [Back-propagation]
15. \( g_{\mu_{n}},b_{m},\mu_{n} \leftarrow (29) \) [Chain rule]
16. \( g_{\Sigma_{n}},b_{m},\Sigma_{n} \leftarrow (30) \) [Chain rule]
17. \( H_n \leftarrow (28) \) [FIM Construction]
18. Initialize \( \lambda_n(k_0) \)
19. for \( k \leftarrow 1 \) to \( k_{\text{max}} \) do
20. \( \bar{\lambda}_n(k) \leftarrow (52) \) [Averaging operation]
21. \( \bar{\theta}_n(k) \leftarrow (53) \) [Primed gradient update]
22. \( \theta_{n}^t(k + 1) \leftarrow (54)-(56) \) [Projection on \( M^L \)]
23. \( \lambda_n(k + 1) \leftarrow (57) \) [Dual gradient update]
24. if \( \| \theta_{n}^t(k + 1) - \theta_{n}^t(k) \| \leq \Delta \theta_n \) then
25. \( \theta_{n}^{t+1} \leftarrow \theta_{n}^t(k + 1); \) Break;
26. end if
27. end for
28. if \( \| \theta_{n}^{t+1} - \theta_{n}^t \| \leq \Delta \theta_n \) then
29. Output \( \theta_{n}^{t+1} \); Break;
30. end if
31. end for

IV. Simulation Results

The proposed method is tested on a modified 33-bus distribution network \([20]\). The case study consists of five MGs as shown in Fig. 2. Each MG is modeled as a modified IEEE 13-bus network at a low voltage level \([21]\). The energy price for the transferred power at the MG PCCs and the fuel price for the local DGs are adopted from \([22]\) and \([23]\), respectively. Both load demands and PV generations data have 15-minute time resolution and are obtained from smart meters to provide realistic numerical experiments \([24]\). The total demand and PV generations of the MGs are shown in Fig. 3.

#### A. System Operation Outcomes

The outcome of DG active power for each MG under the input load demand PV generation scenarios are shown in Fig. 4. It can be seen that the main demand is supplied by the local DGs due to low fuel prices. The ESS charging/discharging power outputs and SOCs for each MG are shown in Fig. 5 where as can be seen the ESSs charge during off-peak period and discharge during peak time to provide economical power to customers.

Table 1 presents a comparison between the centralized optimization method and the proposed constrained RL in one
decision window, in terms of MGs’ operational costs. It can be seen that the MGs’ operational costs from both methods are relatively close to each other, since both the centralized optimization method and the proposed constrained RL method search for the optimal solutions to maximize the social welfare. However, compared to centralized optimization our method provides instantaneous response to time-varying system conditions after completing the training process, instead of solving a new complex optimization problem at each time step. Even though the RL training process takes time, the decision time for the proposed constrained RL method is only 7.46 seconds, which is much shorter than 65.45 seconds for the centralized optimization solver.

B. Algorithm Performance

The learning parameters $\theta_\mu$ and $\theta_\Sigma$ for each MG agent are shown in Fig. 6. As can be seen, the changes in $\theta_\mu$ are relatively larger than that of $\theta_\Sigma$, since the objective functions of the MG agents show higher levels of sensitivity to the mean value of the actions rather than their variance levels in the policy functions.

To better show the performance of the proposed constrained RL method and the importance of considering constraints during the training process, we have compared three cases: (i) with DG capacity constraints in all MGs; (ii) no DG capacity constraints in MG1 and MG2; (iii) no DG capacity constraints in MG1-MG5. As can be seen in Fig. 7, in cases (ii) and (iii), the agents obtain a higher reward compared to case (i) due to constraint omission; however, this comes at the expense of decision infeasibility. In case (i), these operational constraints are satisfied, which also leads to a drop in total reward, as expected. This shows that our proposed constrained RL decision model guarantees the feasibility of the control actions w.r.t. the constraints of the power management problem.

| TABLE II |
| COMPARISON OF MGs’ OPERATIONAL COSTS IN ONE DECISION WINDOW |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | MG1             | MG2             | MG3             | MG4             | MG5             |
| Centralized opt. ($) | 22.123          | 22.339          | 20.796          | 26.378          | 21.061          |
| Constrained RL ($)  | 21.679          | 21.827          | 21.571          | 27.002          | 21.919          |
| Difference (%)     | 2.01            | 2.29            | -3.73           | -2.37           | -4.07           |

Fig. 7. Comparison of reward w/ and w/o constraints.

To further demonstrate this, the constraint returns for the DG capacity constraint in MG1 is given along the training process.
for a time window with length 4 in Fig. 8, where each curve in the figure represents constraint return at one time instant. As can be seen, the conventional unconstrained RL solver violates the upper boundary for the constraint return limit for all time instants; on the other hand, the proposed constrained RL solver satisfies the DG generation capacity constraints, which implies that the constraint return values do not violate the DG capacity limits.

![Fig. 8. Return value comparisons w/ and w/o constraints.](image)

The distributed optimization convergence process is shown in Fig. 9 for one policy gradient update step. As can be seen, the Lagrangian multipliers $\lambda_n$ reach zero over iterations of the proposed distributed training algorithm, which indicates that the global constraints are satisfied and optimal feasible solutions are obtained.

![Fig. 9. The performance of the distributed training method.](image)

V. CONCLUSION

Conventional model-based optimization methods suffer from high computational costs when solving large-scale multi-MG power management problems. On the other hand, the conventional model-free methods are black-box tools, which may fail to satisfy the operational constraints. Motivated by these challenges, in this paper, a distributed multi-agent constrained RL-based method has been proposed for power management of networked MGs. Our proposed method exploits the gradients of the decision problem to learn control policies that achieve both optimality and feasibility. Furthermore, to enhance computational efficiency and maintain the policy privacy of the control agents, a distributed consensus-based optimization solver is implemented to train the agents’ policy functions using local communication. The proposed approach takes the advantages of both model-based and model-free methods.

REFERENCES

[1] Z. Li, M. Shahidehpour, F. Aminifar, A. Alabdulwahab, and Y. Al-Turki, “Networked microgrids for enhancing the power system resilience,” Proc. of the IEEE, vol. 105, no. 7, pp. 1290–1310, July 2017.

[2] M. N. Alam, S. Chakrabarti, and A. Ghosh, “Networked microgrids: State-of-the-art and future prospectives,” IEE Trans. Ind. Informat., vol. 15, pp. 1238–1250, March 2019.

[3] Z. Wang, B. Chen, J. Wang, M. Becovic, and C. Chen, “Coordinated energy management of networked microgrids in distribution systems,” IEE Trans. Smart Grid, vol. 6, no. 1, pp. 45–53, Jan. 2015.

[4] M. Ban, M. Shahidehpour, J. Yu, and Z. Li, “A cyber-physical energy management system for optimal sizing and operation of networked nanogrids with battery swapping stations,” IEE Trans. Sustain. Energy, vol. 10, no. 1, pp. 491–502, Jan 2019.

[5] Z. Wang, B. Chen, J. Wang, and J. Kim, “Decentralized energy management system for networked microgrids in grid-connected and islanded modes,” IEE Trans. Smart Grid, vol. 7, no. 2, pp. 1097–1105, March 2016.

[6] D. Shi, X. Chen, Z. Wang, X. Zhang, Z. Yu, X. Wang, and D. Bian, “A distributed cooperative control framework for synchronized reconnection of a multi-bus microgrid,” IEE Trans. Smart Grid, vol. 9, no. 6, pp. 4664–4665, Nov 2018.

[7] B. Zhao, X. Wang, D. Lin, M. M. Calvin, J. C. Morgan, R. Qin, and C. Wang, “Energy management of multiple microgrids based on a system of systems architecture,” IEE Trans. Power Syst., vol. 33, no. 6, pp. 6410–6421, Nov 2018.

[8] J. Duan, Z. Yi, D. Shi, C. Lin, X. Lu, and Z. Wang, “Reinforcement-learning-based optimal control for hybrid energy storage systems in hybrid ac/dc microgrids,” IEE Trans. Ind. Informat., Early Access 2019.

[9] P. Zeng, H. Li, H. He, and S. Li, “Dynamic energy management of a microgrid using approximate dynamic programming and deep recurrent neural network learning,” IEE Trans. Smart Grid, vol. 10, no. 4, pp. 4435–4445, 2019.

[10] E. Mocanu, D. Constantin, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, and J. Slootweg, “On-line building energy optimization using deep reinforcement learning,” IEE Trans. Smart Grid, vol. 10, no. 4, pp. 3698–3708, July 2019.

[11] T. Yu, H. Z. Wang, B. Zhou, K. W. Chan, and J. Tang, “Multi-agent correlated equilibrium Q learning for coordinated smart generation control of interconnected power grids,” IEE Trans. Power Syst., vol. 30, no. 4, pp. 1669–1679, July 2015.

[12] J. Achiam, D. Held, A. Tamar, and P. Abbeel, “Constrained policy optimization,” in Proc. of 34th International Conference on Machine Learning, 2017.

[13] P. Zeng, H. Li, and S. Li, “Dynamic energy management of a microgrid using approximate dynamic programming and deep recurrent neural network learning,” IEE Trans. Smart Grid, vol. 10, no. 4, pp. 4435–4445, 2019.

[14] Y. Chow, O. Nachum, M. Gharavazadeh, and E. Duen, “A lyapunov-based approach to safe reinforcement learning,” in Proc. of 32nd Conference on NeurIPS, 2018.

[15] M. Men and T. Ufuk, “Constrained cross-entropy method for safe reinforcement learning,” in Proc. of 32nd Conference on NeurIPS, 2018.

[16] J. Li, J. Cheng, J. Shi, and F. Huang, “Brief introduction of back propagation neural network algorithm and its improvement,” Advances in computer science and information engineering, pp. 553–558, 2012.

[17] M. E. Baran and F. F. Wu, “Network reconfiguration in distribution systems for loss reduction and load balancing,” IEEE Trans. Power Del., vol. 4, pp. 1401–1407, Apr. 1989.

[18] W. Kersting, “Radial distribution test feeder,” in Proc. of 2001 IEEE PES Winter Meeting, vol. 2, pp. 908–912, 2001.

[19] Wholesale electricity and natural gas market data. [Online]. Available: https://www.eia.gov/electricity/wholesale/history

[20] Iowa distribution test systems. [Online]. Available: http://wzy.ece.iastate.edu/Testsystem.html