iEnergy

ARTICLE

https://doi.org/10.23919/IEN.2022.0048

Safety-assured, real-time neural active fault management for resilient microgrids integration

Wenfeng Wan[1], Peng Zhang[1] ✉, Mikhail A. Bragin[2] and Peter B. Luh[2,†]

ABSTRACT

Federated-learning-based active fault management (AFM) is devised to achieve real-time safety assurance for microgrids and the main grid during faults. AFM was originally formulated as a distributed optimization problem. Here, federated learning is used to train each microgrid's network with training data achieved from distributed optimization. The main contribution of this work is to replace the optimization-based AFM control algorithm with a learning-based AFM control algorithm. The replacement transfers computation from online to offline. With this replacement, the control algorithm can meet real-time requirements for a system with dozens of microgrids. By contrast, distributed-optimization-based fault management can output reference values fast enough for a system with several microgrids. More microgrids, however, lead to more computation time with optimization-based method. Distributed-optimization-based fault management would fail real-time requirements for a system with dozens of microgrids. Controller hardware-in-the-loop real-time simulations demonstrate that learning-based AFM can output reference values within 10 ms irrespective of the number of microgrids.

KEYWORDS

Active fault management, microgrids, federated learning, real-time safety assurance, resilience.

The number of microgrids is increasing every year. 138 microgrids were added in the USA in 2013 and that number increased to 546 in 2019[1]. Microgrids are small power systems that can operate with or without connecting to the main grid. They can reduce power loss, integrate renewable energy, and increase resilience[2–5].

Active fault management (AFM) for microgrids, systematically devised in ref. [6], is important to maintain both the main grid and microgrids' reliable operation during faults and also to speed up recovery from faults[6–8]. AFM is to control microgrids, when faults happen on the hosting grid, to enable fault ride-through capability, mitigate negative impacts of microgrid interconnections, and eventually enhance grid and microgrid resilience[9–11].

Optimization has been used for fault management[6]. Compared with feedback control, which focuses on certain variables while ignoring other variables[12–14], optimization can consider multiple variables, putting them as objectives and constraints. Optimization can also include complex non-linear functions, such as power ripples during asymmetrical faults. The optimization-based fault management also has feedback and is a combination of optimization and feedback control. Optimization is used to get reference values for microgrids and feedback control is to enforce those reference values. Based on fault management with centralized optimization for a single microgrid, distributed optimization is developed for fault management of networked microgrids. Distributed fault management is implemented by assigning each subproblem to a different CPU core. Those cores are within one CPU[15]. In this way, the computation sequence of subproblems is totally decided by each core's computation capability.

One concern for integrating optimization into power grid fault management and dynamic control is real-time performance because optimization usually takes more time to get reference values than widely used PID feedback control. To address this concern, controller hardware-in-the-loop (HIL) simulation with RTDS simulators is used to demonstrate the real-time performance of distributed-optimization-based fault management algorithms[16]. In the hardware setup, one individual computer exclusively runs one microgrid or PV farm's control algorithm. Real-time simulation results demonstrate that the algorithms can output reference values within 100 ms, which can be considered good enough for fault management and dynamic control.

However, the aforementioned controller HIL simulation has only been used to demonstrate real-time performance for a system with less than 30 controllable units. More controllable units, microgrids in this paper, would lead to more computation time since more communication time is needed for updates during distributed optimization. If the optimization-based fault management algorithm needs more than 100 ms to output reference values, the fault management is expected to have compromised performance or even fail.

As a type of distributed machine learning methods operated through edge computing, federated learning enables multiple entities to collaboratively train a shared model while keeping each entity's data local[17–20]. Federated-learning-based fault management can keep distributed optimization's advantages, e.g., flexibility and privacy, and at the same time avoid long computation time for a power system with a large number of microgrids, e.g., hundreds of microgrids.

[1]Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794, USA; [2]Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA;

[†]We dedicate this paper to the loving memory of Professor Peter B. Luh. Along with this paper, his vision was to revolutionize the way difficult and important mathematical programming problems are solved through the innovative use of AI and machine learning. Yet, on 11/28/2022 he tragically left the world. Professor Luh will be dearly missed.

Address correspondence to Peng Zhang, p.zhang@stonybrook.edu

The main contribution of this work is to scale up optimization-based fault management for a large number of microgrids by federated learning. First, the training data are achieved by distributed optimization. The inputs are microgrid terminal voltages, fault currents near fault locations, and load currents. Outputs are reference currents of microgrids. After that, training data are used to train neural networks. Every microgrid has one neural network. Some of each microgrid' network parameters are coordinated with other microgrids' corresponding parameters during training. Then, the trained neural networks are used in controller HIL real-time simulation. The learning-based AFM algorithm is put on a computer, which is then connected to RTDS simulators by communication networks. The real-time simulation results demonstrate that during faults, learning-based AFM is able to output reference values to microgrids within 5 ms irrespective of the number of microgrids. This is because, after training, AFM controllers mainly comprise simple addition and multiplication computations.

The rest of this paper is organized as follows. Section 1 is the method and implementation of learning-based AFM, including problem formulation of optimization-based AFM and the process of learning-based AFM. Section 2 is a case study, including single-phase-to-ground faults, federated learning results, resilient analysis, and scalability analysis. In the case study of single-phase-to-ground faults, three methods have been compared: learning-based AFM, optimization-based AFM, and a non-AFM simple ride-through method. Section 2 presents a summary of this paper.

1 Methods and implementation of learning-based AFM

1.1 Optimization-based AFM

This section describes AFM's formulation as a distributed optimization problem. Figure 1 shows the schematic of learning-based AFM for microgrids integration. Microgrids connect to the hosting grid by interfacing converters. Three types of microgrids have been considered: microgrids with DC link batteries, microgrids with AC link batteries, and microgrids without batteries. The centralized formulation is first presented, and then the decomposition of the centralized formulation into distributed formulation is given.

1.1.1 Centralized optimization

The formulation of centralized AFM is given in Eqs. (1)–(7). The objective function has three parts, F_1, F_2, and F_3. F_1 are the fault current contributions, the magnitude difference between fault currents from the main grid and total fault current. F_2 is the reactive power output. F_3 is the power ripple. Fault current contributions and power ripples are to be minimized and reactive power output is to be maximized. That is why there is a negative sign for F_2 in Eq. (2).

$$\min \quad \alpha_1 F_1 + \alpha_2 F_2 + \alpha_3 F_3, \tag{1}$$

where

$$\begin{cases} F_1 \equiv \sum_{fp} \left| \dfrac{[\mathrm{Re}(\mathbf{I}_{fp}^{M} + \mathbf{I}_{fp}^{ms})]^2 + [\mathrm{Im}(\mathbf{I}_{fp}^{M} + \mathbf{I}_{fp}^{ms})]^2}{[\mathrm{Re}(\mathbf{I}_{fp}^{M})]^2 + [\mathrm{Im}(\mathbf{I}_{fp}^{M})]^2} - 1 \right| \\ F_2 \equiv -\sum_{i=1}^{N} Q_i^m \\ F_3 \equiv \sum_{i=1}^{N} \dfrac{R_i}{(P_i^m)^2} \\ R_i \equiv \{\sum_j [\mathrm{Re}(\mathbf{U}_{i,j}^m)\mathrm{Re}(\mathbf{I}_{i,j}^m) - \mathrm{Im}(\mathbf{U}_{i,j}^m)\mathrm{Im}(\mathbf{I}_{i,j}^m)]\}^2 \\ \quad + \{\sum_j [\mathrm{Im}(\mathbf{U}_{i,j}^m)\mathrm{Re}(\mathbf{I}_{i,j}^m) + \mathrm{Re}(\mathbf{U}_{i,j}^m)\mathrm{Im}(\mathbf{I}_{i,j}^m)]\}^2 \\ (i = 1, \ldots, N; j = a, b, c; fp \in P(\{a, b, c\})), \end{cases} \tag{2}$$

where subscript fp represents the faulty phases. $fp \in P(\{a, b, c\})$, power set of $\{a, b,$ and $c\}$. Subscript i indicates variables related to microgrid i. Subscript j means power systems' phases $a, b,$ and c. Superscripts $M, m,$ and ms indicate variables related to the main grid, a single microgrid, and all microgrids, respectively. $\mathbf{I}_{fp}^{M}$ and $\mathbf{I}_{fp}^{ms}$ are fault currents from the main grid and from all microgrids, respectively. $\mathbf{I}_{fp}^{M} + \mathbf{I}_{fp}^{ms}$ is the total fault current. $\mathbf{I}_{fp}^{ms} = \sum_i \mathbf{I}_{i,fp}^{m}$ is the sum of all microgrids' fault currents. Q_i^{PV} is the reactive power output from PV farm i. α_1, α_2, and α_3 are weighting factors between different parts in the objective function. The formulation is given in the rectangular coordinate systems. Symbols Re and Im are y and x coordinates, respectively, indicating the real part and the imaginary part of complex numbers, respectively.

AFM has two types of constraint: system-wide coupling constraints and local constraints. System-wide coupling constraints contain decision variables of more than one microgrid, and local constraints only involve decision variables of one microgrid.

Coupling constraints: tie line safety rating constraint. This means the sum of all microgrids' output currents, $\mathbf{I}_j^{ms} = \sum_i \mathbf{I}_{i,j}^{m}$, should be less than a safety threshold, $I^{R,ms}$

$$[\mathrm{Re}(\mathbf{I}_j^{ms})]^2 + [\mathrm{Im}(\mathbf{I}_j^{ms})]^2 \le (I^{R,ms})^2. \tag{3}$$

Local constraint 1: microgrid-wise safety rating constraint. Each microgrid's output currents during fault should be less than its own safety rating, $I_i^{R,m}$:

$$[\mathrm{Re}(\mathbf{I}_{i,j}^{m})]^2 + [\mathrm{Im}(\mathbf{I}_{i,j}^{m})]^2 \le (I_i^{R,m})^2. \tag{4}$$

Local constraint 2: zero sequence component elimination constraint. This constraint means the sum of each microgrid's three-phase currents is zero. This constraint is required if a microgrid's interface converter or nearby transformer does not allow zero-sequence currents:

$$\sum_j \mathbf{I}_{i,j}^{m} = 0. \tag{5}$$

Local constraint 3: power balance constraint, meaning a microgrid's active power output during faults is the same as before faults. This constraint is required if a microgrid has no battery installed and is not required if a microgrid has a battery. Power balance constraint is to reduce the large change in power flow and thus to reduce disturbance on the microgrid and nearby grids:

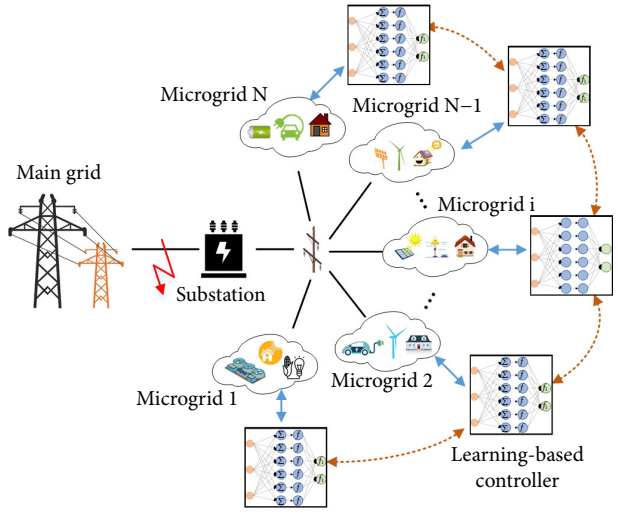

Fig. 1 Schematic of learning-based AFM for microgrids integration.

$$P_i^m = P_i^{bf}. \tag{6}$$

Local constraint 4: battery power buffer constraint. For microgrids with batteries, the power output difference between before faults and after faults should be smaller than the power rating of the PV farm's battery. Otherwise, it is beyond the battery's capability to keep a power balance, meaning the sum power output of the PV farm and its battery during the fault would be the same as before the fault:

$$(P_i^m - P_i^{bf})^2 \le (P_i^{bty})^2. \tag{7}$$

1.1.2 Distributed optimization

Distributed and asynchronous surrogate Lagrangian relaxation (DA-SLR)[21–24] is used to decompose the above centralized formulation into the distributed formulation, resulting in N subproblems with a subproblem for a microgrid. The system-wide coupling constraint Eq. (3) is multiplied by Lagrangian multiplier λ and then added to the objective function. The value of the objective function will be increased as penalties if the coupling constraint is violated. With this Lagrangian relaxation, one centralized problem is decomposed into N subproblems with each subproblem only having decision variables from one microgrid. Those subproblems are then calculated in a distributed and asynchronous way by assigning each subproblem to a single core within a multi-core CPU.

Three types of subproblems would be achieved, corresponding to three types of microgrids: microgrids without battery, microgrids with DC-link battery, and microgrids with AC-link battery (Figure 2). All microgrids connect to the hosting grid by power converters. DC-link battery and AC-link battery mean the battery is installed in the converter's DC side and AC side, respectively. In the following subproblems, $\boldsymbol{\lambda} = [\lambda_a, \lambda_b, \lambda_c]$ and $\mathbf{g} = [g_a, g_b, g_c]$. g_j $(j = a, b, c)$ are given in Eq. (8) after relaxation of coupling constraint Eq. (3):

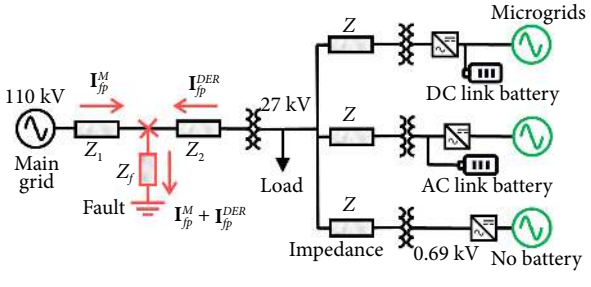

Fig. 2 Diagram of studied system.

$$g_j = [\mathrm{Re}(\mathbf{I}_j^{ms})]^2 + [\mathrm{Im}(\mathbf{I}_j^{ms})]^2 - (I^{R,ms})^2. \tag{8}$$

All microgrids have fault current contributions F_1 as objectives and the constraints of current rating Eq. (4) and zero-sequence currents elimination Eq. (5). Microgrids without a battery, have power ripples F_3 as objective functions and power balance Eq. (6) as constraints, since no battery exists to buffer power ripples and power output. Microgrids with DC-link battery, have reactive power output F_2 as objective functions and power buffer of battery Eq. (7) as constraints, since DC-link battery exists to buffer power ripples and power output, and there are limits on power flow of battery. Microgrids with an AC-link battery, have reactive power output F_2 as objective functions and power balance Eq. (6) as constraints, since an AC-link battery exists to buffer power ripples and the total active power output of the microgrid's converter and its battery is controlled to be the same as before fault.

Subproblems for microgrids without battery:

$$\min \quad \beta_{i,1}F_1 + \beta_{i,2}F_3 + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{g}, \tag{9}$$

$$\text{s.t. Eqs. (4), (5), and (6).} \tag{10}$$

Subproblems for microgrids with DC-link battery:

$$\min \quad \beta_{i,1}F_1 + \beta_{i,2}F_2 + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{g}, \tag{11}$$

$$\text{s.t. Eqs. (4), (5), and (7).} \tag{12}$$

Subproblems for microgrids with AC-link battery:

$$\min \quad \beta_{i,1}F_1 + \beta_{i,2}F_2 + \boldsymbol{\lambda}^{\mathrm{T}}\mathbf{g}, \tag{13}$$

$$\text{s.t. Eqs. (4), (5), and (6).} \tag{14}$$

1.2 Learning-based AFM.

The topology of a single microgrid's artificial neural network is feed-forward neural networks. Among machine learning's three main functions: classification, regression, and clustering[25], this neural network is trained for regression. Regression means estimating relationships between outputs and inputs. In the case of AFM, inputs are system status, such as microgrid voltages, fault voltages, fault currents, power, etc. Outputs are reference values for the microgrid's output currents. Previously, outputs were decided by optimizations used in AFM. Here, neural works are used to approximate the optimization function. The used neural network is feed-forward with a multi-layer perceptron without feedback between layers.

Figure 3 shows the schematic of federated learning used in learning-based AFM. Each microgrid has one neural work to train and all the microgrids would participate in coordinating some parameters of their neural networks. Federated learning keeps data and training localized, preserving privacy and leveraging local computation and data storage resources. Federated learning has been largely used for remote devices, e.g., mobile phones and the internet of things. In learning-based AFM, data are stored and trained locally at the microgrid side and all microgrids communicate with a central coordinator periodically for parameter updating.

Algorithm 1: Distributed and asynchronous federated learning for microgrids' active fault management (AFM)

Result: N microgrids' neural networks

1 initialization: each layer's weights $\mathbf{W}_{j,i}$ and bias $\mathbf{b}_{j,i}(i = 1, ..., N, j = 1, ..., L)$, epoch E, maximum iterations K

2 iteration $k \leftarrow 1$

3 while: $\|\mathbf{W}^{c,k} - \mathbf{W}^{c,k-1}\| > \sigma_1$ or $\|\mathbf{b}^{k} - \mathbf{b}^{k-1}\| > \sigma_2$ **do**

4 N neural networks start training

5 Neural network i' of microgrid i' finishes its E epoches training, sends to the coordinator the first layer's weights $\mathbf{W}_{1,i'}$ and bias $\mathbf{b}_{1,i'}$

6 The coordinator updates $\mathbf{W}^{c,k}$ and $\mathbf{b}^{c,k}$ by averaging, i.e., $\mathbf{W}^{c,k} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{W}_{1,i}^{c,k}$, $\mathbf{b}^{c,k} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{b}_{1,i}^{c,k}$

7 iteration $k \leftarrow k+1$

8 if: $k \ge K$ **then**

9 break

10 end

11 end

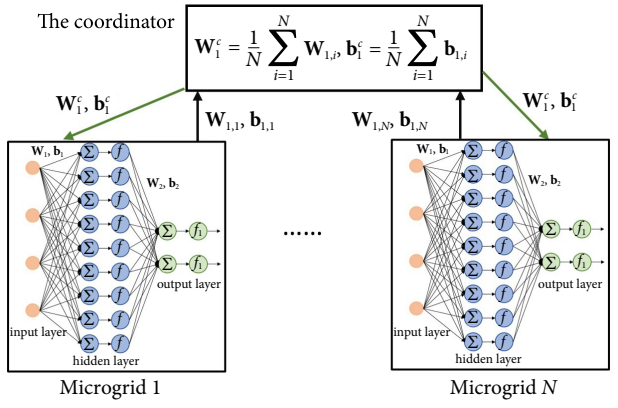

Fig. 3 Schematic of learning-based AFM for microgrids integration.

Algorithm 1 shows the process for distributed and asynchronous federated learning for AFM. The coordinator first chooses a network topology, e.g., the number of layers, the number of nodes in each layer, and activation functions. Then, the coordinator transmits network topology to all microgrids. All microgrids train their neural networks with their own data. The data are achieved by distributed optimization in Eqs. (10)–(14). In our work, the weights $\mathbf{W}_1$ and bias $\mathbf{b}_1$ between the input layer and the hidden layer in Figure 3 are to be coordinated. After a microgrid finishes its training, it sends $\mathbf{W}_1$ and $\mathbf{b}_1$ to the coordinator. The coordinator would then update $\mathbf{W}_1$ and $\mathbf{b}_1$ by averaging, i.e., $\mathbf{W}_1^c = \frac{1}{N}\sum_{i=1}^{N}\mathbf{W}_{1,i}$ and $\mathbf{b}_1^c = \frac{1}{N}\sum_{i=1}^{N}\mathbf{b}_{1,i}$. $\mathbf{W}_{1,i}$ and $\mathbf{b}_{1,i}$ are weights and bias from microgrid i. The aforementioned process continues until some predefined criteria are reached, such as the change of training accuracy being below a threshold. Theoretical discussions about the convergence of federated learning can be found in refs. [26–28].

2 Case study

The studied system has six microgrids (Figure 2). Microgrid 1 and microgrid 4 do not have batteries; microgrid 2 and microgrid 5 have DC-link batteries; microgrid 3 and microgrid 6 have AC-link batteries. Before faults, six microgrids output 2.4 MW, 1.0 MW, 2.0 MW, 1.2 MW, 1.2 MW, and 0.6 MW to the main grid, all with power factor 1.0. Those microgrids connect to the main grid by power converters.

For the setup of controller hardware-in-the-loop (HIL) real-time simulation, the learning-based controller algorithms are run on a personal computer or server, and power grids are run in RTDS simulators. The controller computer and simulators are connected to each other by internet networks.

2.1 Single-phase-to-ground faults

A single-phase-to-ground fault happens at phase *a* on the 110 kV grid, as shown in Figure 2. The fault happens at 0.1 s and clears at 0.3 s. The fault is simulated by connecting the ground and phase *a* with a resistor of 325 Ω.

2.1.1 With learning-based AFM

Neural networks trained with federated learning act as fault management algorithms for microgrids. Figure 4 shows voltages at fault locations and microgrids output voltages. At fault locations, phase *a* voltages are 59.3 kV during faults, 60.5% of pre-fault values. phase *b* and phase *c* voltages increase to 110.6% and 116.3%,

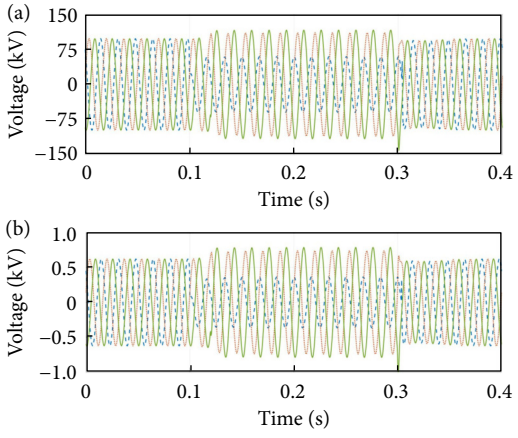

Fig. 4 Results for single-phase-to-ground faults with learning-based AFM. Voltages at (**a**) fault locations and (**b**) microgrids terminal.

respectively. The voltage increase is caused by microgrids outputting reactive power.

Figure 5(a) shows timestamp signals sent between RTDS simulators and the controller. In controller HIL experiments, the timestamp signals are sent from RTDS simulators to controllers once every 20 ms and increase by 1 after each sending. After computing reference values, controllers send reference values and the timestamp signals back to RTDS simulators. Delays between sent and received timestamp signals are controller computation time. In Figure 5(a), the AFM computation time is 3.0 ms, including 1.8 ms two-way communication time. The reason for such fast computation time is that after neural networks are trained, their computations are mainly arithmetic, e.g., addition and multiplication, without iterations. Figure 5(b) is microgrid 1's phase *a* currents sent from the controller to RTDS simulators. Abrupt changes in the reference values indicate new reference values have been sent to RTDS simulators. After 0.16 s, new reference values are sent to simulators continuously, but their values keep the same because the system status becomes unchanged.

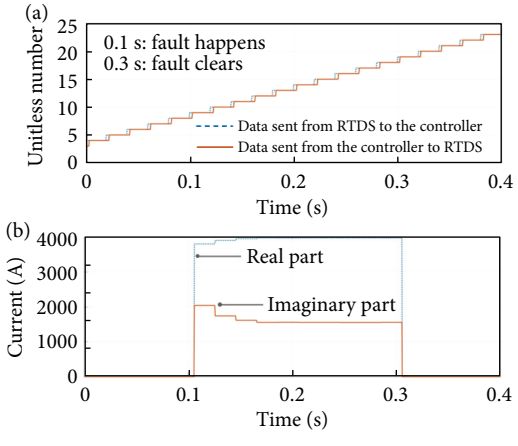

Fig. 5 Results for single-phase-to-ground faults with learning-based AFM. (**a**) Timestamp signals sent between RTDS simulators and the controller. (**b**) Microgrid 1's phase *a* currents sent from the controller to RTDS simulators.

Figure 6 includes fault currents at fault locations, and the magnitude difference between total fault currents and fault current from the main grid, i.e., $\left||\mathbf{I}_a^M + \mathbf{I}_a^{ms}| - |\mathbf{I}_a^M|\right|$. Total fault current and fault current from the main grid are 184.0 A and 176.1 A, respec-

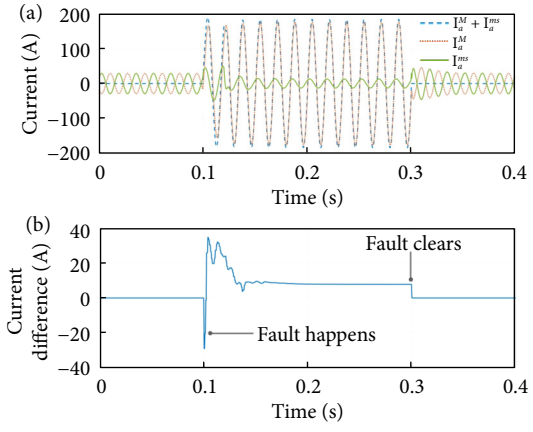

Fig. 6 Results for single-phase-to-ground faults with learning-based AFM. (**a**) phase a currents at the fault location. $\mathbf{I}_a^M + \mathbf{I}_a^{ms}$, $\mathbf{I}_a^M$, and $\mathbf{I}_a^{ms}$ are total fault currents to the ground, fault currents from the main grid, and fault currents from microgrids, respectively. (**b**) The magnitude difference between $\mathbf{I}_a^M + \mathbf{I}_a^{ms}$ and $\mathbf{I}_a^M$, i.e., $\left| |\mathbf{I}_a^M + \mathbf{I}_a^{ms}| - |\mathbf{I}_a^M| \right|$, which are also fault current contributions from microgrids.

tively and thus microgrids' current contributions are 7.9 A or 4.5%. The magnitude difference should be as close to 0 as possible. All microgrids have this fault current contributions in their sub-problem formulation. Figure 7 shows the power outputs of microgrid 1, microgrid 2, and microgrid 3, including instantaneous power, active power, and reactive power. The three microgrids are of three types. microgrid 1 doesn't have a battery and is controlled to output the same amount of active power as before the fault and to minimize power ripples in instantaneous power. Microgrid 2 has a DC-link battery and is controlled to output reactive power as much as possible. microgrid 3 has an AC-link battery and is controlled to output the same amount of active power as before the fault and to output reactive power as much as possible. The results in Figure 7 show microgrids function as required.

2.1.2 *With Optimization-based AFM*

Distributed optimization acts as a fault management algorithm for microgrids. Figure 8 shows the controller's computation sequence and transferred reference values to RTDS simulators. The fault happens at 0.1 s and the fault detection signal was received by the controller after 8.8 ms. The first set of reference values is sent to RTDS simulators at 0.156 s since it takes the optimization-based controller 47.2 ms to finish the first optimization computation. From 0.1 s to fault clearance at 0.3 s, the controller has sent reference values to RTDS simulators three times, taking 47.2 ms, 58.3 ms and 56.4 ms, respectively.

Figure 9 shows currents at fault locations. The total fault currents $\mathbf{I}_a^M + \mathbf{I}_a^{ms}$ and fault currents from the main grid $\mathbf{I}_a^M$ have magnitudes of 166.2 A and 173.4 A, respectively. Microgrids' current contributions are −7.2 A or −4.3%. Figure 10 shows the power output of microgrid 1, microgrid 2, and microgrid 3, including instantaneous power, active power, and reactive power. In Figure 9 and Figure 10, RTDS simulators received reference value 56 ms later after fault happens, so from 0.1 s to 0.156 s, microgrids are using reference values from pre-fault controller. At 0.156 s, microgrids start to use reference values from the optimization-based AFM controller. That is why current difference and power outputs have abrupt changes at 0.156 s.

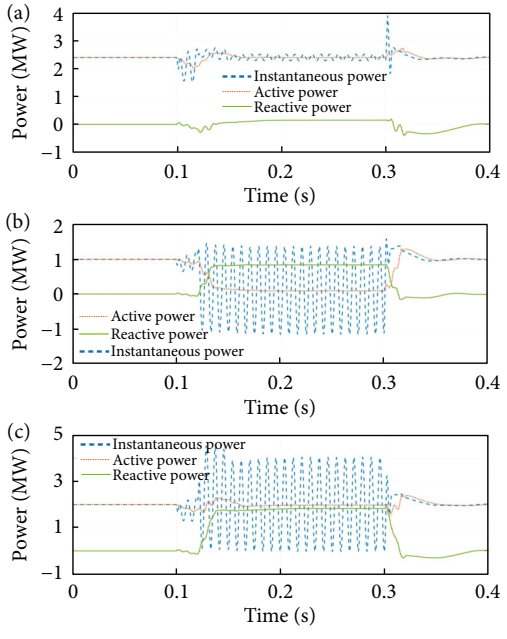

Fig. 7 Results for single-phase-to-ground faults with learning-based AFM. Power outputs of (**a**) microgrid 1, (**b**) microgrid 2, and (**c**) microgrid 3, including instantaneous power, active power, and reactive power.

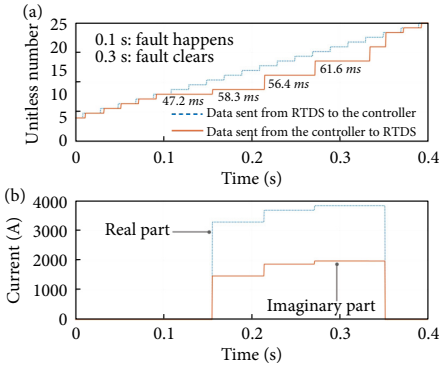

Fig. 8 Results for single-phase-to-ground faults with optimization-based AFM. (**a**) Timestamp signals are sent between RTDS simulators and the controller. (**b**) microgrid 1's phase a currents are sent from the controller to RTDS simulators.

2.1.3 *Without AFM*

The subsection presents results when AFM is not implemented. Instead, microgrids are in power control mode during faults, outputting the same active power and reactive power while maintaining currents within safety ratings. Figure 11 shows results without AFM, including currents at fault locations, current contributions, and power output of microgrid 2. Since all microgrids have the same power control methods, other microgrids have similar power curves with microgrid 2 but with different values. The total to-ground fault currents $\mathbf{I}_a^M + \mathbf{I}_a^{ms}$ and fault currents from the main grid $\mathbf{I}_a^M$ have magnitudes of 183.1 A and 152.4 A, respectively. Microgrids' current contributions are 30.7 A or 20.1%.

Table 1 lists comparisons of the above-mentioned three control methods during faults: learning-based AFM, optimization-based AFM, and without AFM. Metrics for comparison include response time, fault current contributions, power ripples of type 1 microgrid, and total reactive power output of microgrids. Except

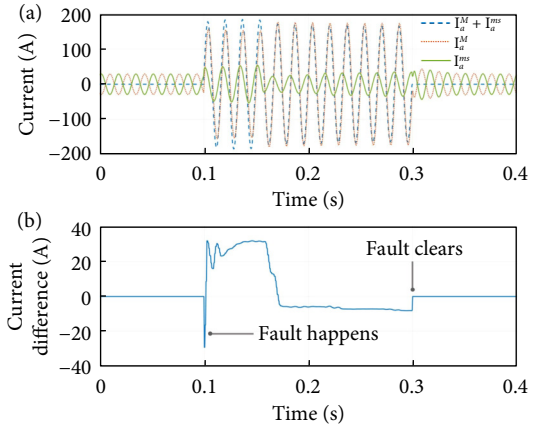

Fig. 9 Results for single-phase-to-ground faults with optimization-based AFM. (**a**) phase a currents at the fault location. $\mathbf{I}_a^M+\mathbf{I}_a^{ms}$, $\mathbf{I}_a^M$, and $\mathbf{I}_a^{ms}$ are total fault currents to the ground, fault currents from the main grid, and fault currents from microgrids, respectively. (**b**) The magnitude difference between $\mathbf{I}_a^M+\mathbf{I}_a^{ms}$, and $\mathbf{I}_a^M$, i.e., $|\mathbf{I}_a^M+\mathbf{I}_a^{ms}|-|\mathbf{I}_a^M|$, which are also fault current contributions from microgrids.

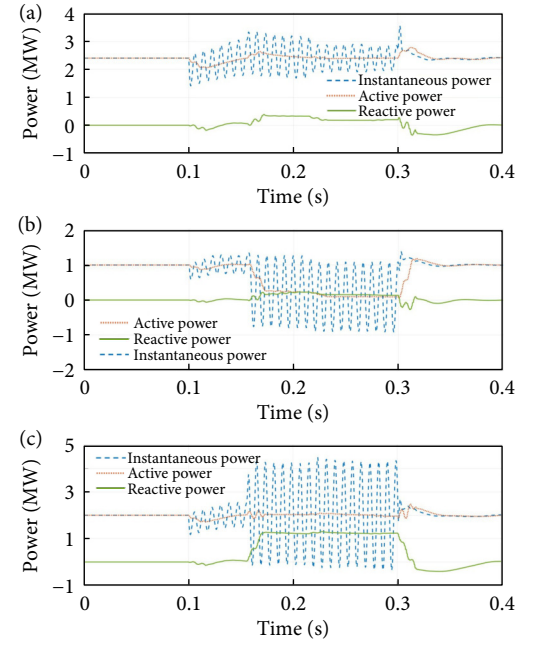

Fig. 10 Results for single-phase-to-ground faults with optimization-based AFM. The power output of (**a**) microgrid 1, (**b**) microgrid 2, and (**c**) microgrid 3, including instantaneous power, active power, and reactive power.

for response time, other metrics of learning-based AFM and optimization-based AFM are expected to be close. Yet, learning-based AFM performs better than optimization-based AFM with respect to power ripples and reactive power. The reason is that the number of optimization rounds is reduced to ensure the real-time performance of optimization-based AFM. Optimization would terminate before optimal values are reached. By contrast, during generating training data from distributed optimization in learning-based AFM, optimization only ends after optimal values are reached.

2.2 Federated learning

The analysis of federated learning is for the case of single-phase-to-ground faults. The learning-based controller in Ssection 2.1 is the

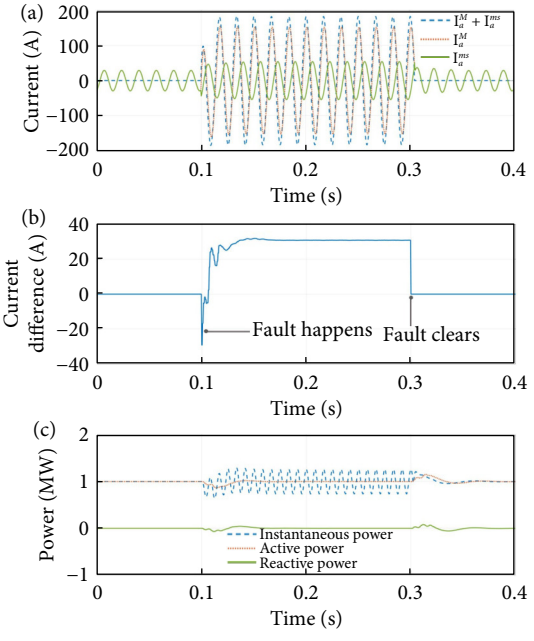

Fig. 11 Results for single-phase-to-ground faults without AFM. (**a**) phase a currents at the fault location. $\mathbf{I}_a^M+\mathbf{I}_a^{ms}$, $\mathbf{I}_a^M$, and $\mathbf{I}_a^{ms}$ are total fault currents to the ground, fault currents from the main grid, and fault currents from microgrids, respectively. (**b**) The magnitude difference between $\mathbf{I}_a^M+\mathbf{I}_a^{ms}$, and $\mathbf{I}_a^M$, i.e., $|\mathbf{I}_a^M+\mathbf{I}_a^{ms}|-|\mathbf{I}_a^M|$, which are also fault current contributions from microgrids. (**c**) The power output of microgrid 2.

Table 1 Comparison of different control methods during faults

Metric	Learning-based AFM	Optimization-based AFM	Without AFM
Response time (ms)	3.0	54.0	≤3.0
Fault current contributions	4.5%	−4.3%	20.1%
Power ripples of type 1 microgrid	5.3%	17.5%	26.3%
Total reactive power output of microgrids (MW)	4.7	2.3	0.0

result of this section's federated learning. Before learning, a total of 6561 sets of data are achieved by distributed optimization.

There are six neural networks to train, with one microgrid having one neural network. All six neural networks have the same topology: an input layer with eight nodes, a hidden layer with 10 nodes, and an output layer with six nodes. The six outputs are microgrids output currents with two outputs for each phase and six outputs for three phases. The number of hidden layers and the number of nodes in each hidden layer can be customized based on training performance.

Asynchronous training is adopted. It means in Figure 2, the coordinator would update weight $\mathbf{W}_i^c$ and bias $\mathbf{b}_i^c$ after any of the six microgrids finishes its current training and sends out its $\mathbf{W}_i$ and $\mathbf{b}_i$ without waiting for all six microgrids finishing their current training. Parallel computation with multiple cores within one CPU is used to realize asynchronous training. Figure 12(a) shows the difference of $\mathbf{W}_i^c$ and $\mathbf{b}_i^c$ between two consecutive rounds of training. Figure 12(b) shows the training finish sequence for the six neural networks. The computation sequence is not predefined and is decided by each core's computation capacity.

Figures 13, 14, and 15 show the training error for microgrid 1,

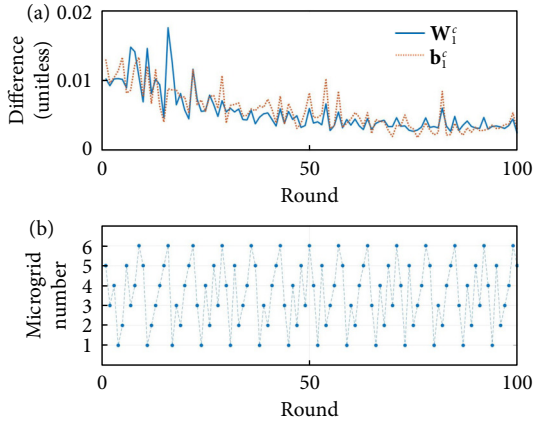

Fig. 12 Federated learning results. (**a**) Difference of W_1^c and b_1^c between two consecutive rounds of training. (**b**) Training finish sequence for the six neural networks.

microgrid 2, and microgrid 3's three phases, respectively. The *x*-axis is for currents' real parts and the *y*-axis is for imaginary parts. The error is calculated based on the following equation,

$$e = \frac{A_L - A_D}{A_R} \tag{15}$$

where A_L and A_D are values obtained from federated learning and distributed optimization, respectively. A_R is a microgrid's rated value. Root mean square (RMS) errors for microgrid 1's three phases in Figure 13 are 4.3%, 4.5%, and 3.0%. RMS errors for microgrid 2's three phases in Figure 14 are 10.9%, 9.9% and 8.8%. RMS errors for microgrid 3's three phases in Figure 15 are 6.7%, 6.2% and 3.7%.

2.3 Resilience analysis

Resilience has been defined as the ability of an entity to respond to and recover from a disturbance[29–31]. In this paper, the entity is networked microgrids and the disturbance is electrical faults. One widely used resilience metric is refs. [32] and [33],

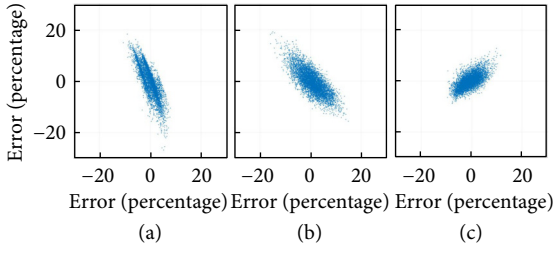

Fig. 13 Federated learning errors of currents' real parts (*x*-axis) and imaginary parts (*y*-axis) for microgrid 1's (**a**) phase *a*, (**b**) phase *b*, and (**c**) phase *c*.

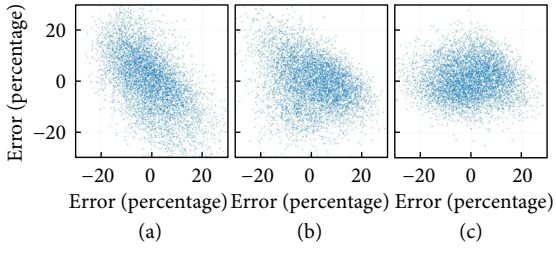

Fig. 14 Federated learning errors of currents' real parts (*x*-axis) and imaginary parts (*y*-axis) for microgrid 2's (**a**) phase *a*, (**b**) phase *b*, and (**c**) phase *c*.

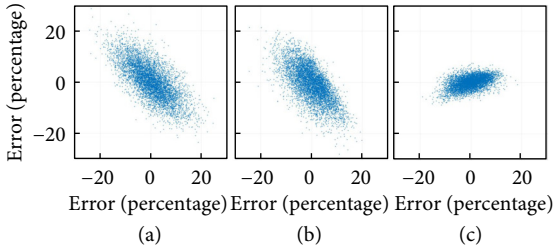

Fig. 15 Federated learning errors of currents' real parts (*x*-axis) and imaginary parts (*y*-axis) for microgrid 3's (**a**) phase *a*, (**b**) phase *b*, and (**c**) phase *c*.

$$\xi = 0.5(\xi_1 + \xi_2) \tag{16}$$

$$\xi_1 = \frac{d_1}{d_0} \tag{17}$$

$$\xi_2 = \frac{\int_{t_1}^{t_2} d_t \mathrm{d}t}{d_0(t_2 - t_1)} \tag{18}$$

where ξ is the resilience. ξ_1 is the invulnerability and ξ_2 is the recovery. d_1 is the value of the variable of interest during disturbance and d_0 is the variable's values before disturbance. t_1 and t_2 are the disturbance happening time and clearing time, respectively. ξ, ξ_1, and ξ_2 are all in the range of 0 to 1. Larger ξ means the system is more resilient. Table 2 and Table 3 show resilience metrics for voltages and current contributions, respectively. Values in the table are calculated based on Figure 16. The resilience metrics for current contributions are calculated using the worst current contributions, 54.4 A, as a base.

Table 2 Resilience metrics for microgrid voltages

Metric	With AFM	Without AFM
Invulnerability	58.9%	54.9%
Recovery	63.9%	61.2%
Resilience	61.4%	58.1%

Table 3 Resilience metrics for current contributions

Metric	With AFM	Without AFM
Invulnerability	85.5%	43.0%
Recovery	81.0%	46.0%
Resilience	83.3%	44.5%

2.4 Scalability analysis

The motivation of this paper is to use federated learning to train optimization-based AFM controllers so that AFM controllers can output reference values fast enough when the system has more than dozens of microgrids. Figure 17 shows a diagram of a studied system with 36 microgrids for scalability analysis. The main grid, fault type, and fault location are the same as in the Section. 2.1. The total output power of the 36 microgrids is 8.4 MW before faults.

Figure 18(a) shows computation times, i.e., the time for the controller to output reference values for microgrids, is 3.0 ms, the same as the case study with 6 microgrids. After each microgrid's neural network is trained, each microgrid's learning-based controller mainly consists of simple addition and multiplication com-

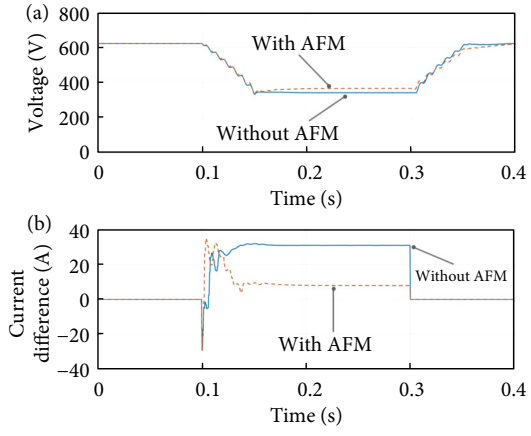

Fig. 16 Resilience curves for (**a**) voltages at microgrids' point of connection and (**b**) fault current contributions.

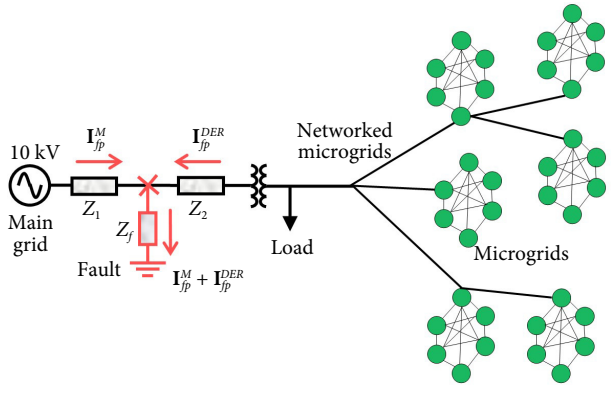

Fig. 17 Diagram of studied system with 36 microgrids for scalability analysis.

putations. This would result in similar computation time irrespective of the number of microgrids. Figure 18(b) is microgrid 1's phase *a* currents sent from the controller to RTDS simulators. The fault happens at 0.1 s and RTDS simulators send the fault detection signal to controllers at 0.105 s. Controllers sent out the first reference values at 0.108 s, 3.0 ms after the fault detection signal and system status are sent to controllers. Figure 18(c) and Figure 19 show fault current contributions and power output of three types of microgrids, respectively. The curves in those four sub-figures correspond to the formulation in (9)–(14).

3 Conclusions

Active fault management (AFM) based on federated learning is established to realize ultra integration of hundreds of microgrids, enabling them to output reference values fast enough during fault ride-through. AFM is first formulated as a distributed optimization problem. Then federated learning is used to train each microgrid's neural network. The trained data are from the distributed optimization and trained neural networks are used as microgrids' controllers. Learning-based AFM can output reference values much faster than distributed optimization because neural networks mainly contain arithmetic computations, such as addition and multiplication while computation of distributed optimization has iterations. The case study demonstrates that the computation time of learning-based AFM is around 3 ms and the computation time would not increase as the number of microgrids increases. The

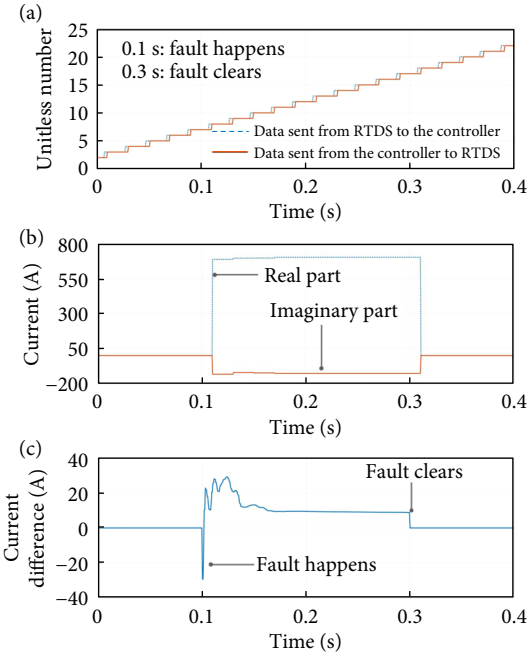

Fig. 18 Results for scalability analysis with 36 microgrids. (**a**) Timestamp signals are sent between RTDS simulators and the controller. (**b**) Microgrid 1's phase *a* currents are sent from the controller to RTDS simulators. (**c**) The magnitude difference between $\mathbf{I}_a^M + \mathbf{I}_a^{ms}$, and $\mathbf{I}_a^M$, i.e., $|\mathbf{I}_a^M + \mathbf{I}_a^{ms}| - |\mathbf{I}_a^M|$, which are also fault current contributions from microgrids.

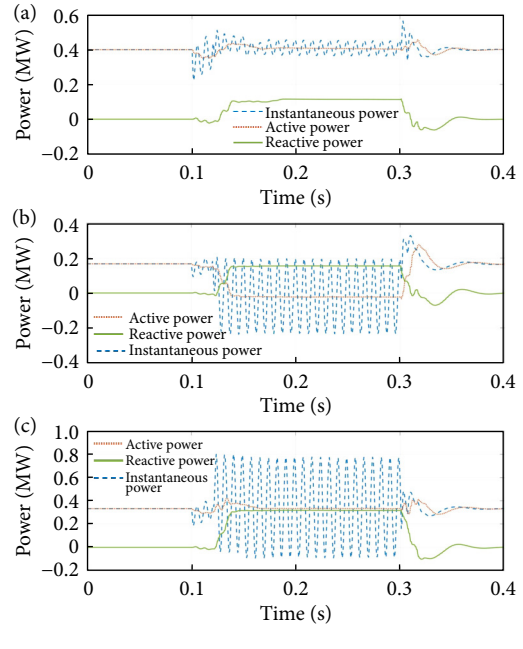

Fig. 19 Results for scalability analysis with 36 microgrids. The power output of (**a**) one type 1 microgrid, (**b**) one type 2 microgrid, and (**c**) one type 3 microgrid, including instantaneous power, active power, and reactive power.

rationale of replacing computation-intensive control with learning-based control can be applied to other dynamic control for better real-time performance. Future work includes transfer learning in fault management so that training results can be used between different types of faults and systems. Also, training for hundreds of microgrids should be carefully designed and optimized with respect to training data acquisition and training speed.

Appendix

Nomenclature

Subscripts

fp	Faulty phase
i	Micrgord *i*; neural network *i*
j	Phase *j*, $j \in \{a, b, c\}$

Superscripts

bf	Value before faults
bty	Value of battery
k	Iteration
M	Value of the main grid
m	Value of a single microgrid
ms	Value of multiple microgrids
R	Safety rating

Others

Im	Imaginary pars of complex numbers
Re	Real part of complex numbers

Variables

α	Weighting factor
β	Weighting factor
λ	Lagrangian multiplier
b	Bias in neural network
F	Objective function
I	Current
N	The number of microgrids; the number of neural networks
P	Active power
Q	Reactive power
V	Voltage
W	Weight in neural network

Acknowledgements

This work was supported in part by the National Science Foundation under Grants No. OIA-2134840 and ECCS-1810108, and in part by Department of Energy under Grant No. DE-EE0009341. This work relates to Department of Navy award N00014-20-1-2858 issued by the Office of Naval Research. The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein.

Article history

Received: 23 August 2022; Revised: 10 October 2022; Accepted: 12 October 2022

Additional information

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Wood Mackenzie (2020). Record number of microgrids installed in US last year. Available at https://www.woodmac.com/pressreleases/record-number-of-microgrids-installed-in-us-last-year/.

[2] Booth, S., Reilly, J., Butt, R., Wasco, M., Monohan, R. (2020). Microgrids for energy resilience: A guide to conceptual design and lessons from defense projects. Technical Report, NREL/TP-7A40-72586, National Renewable Energy Laboratory (NREL), Golden, CO, USA. Available at https://www.osti.gov/biblio/1598145/.

[3] Wang, L. Z., Qin, Y. Y., Tang, Z. F., Zhang, P. (2020). Software-defined microgrid control: The genesis of decoupled cyber-physical microgrids. *IEEE Open Access Journal of Power and Energy*, 7: 173–182.

[4] Microgrid Knowledge. What is a microgrid? Available at https://microgridknowledge.com/microgrid-defined/.

[5] Feng, F., Zhang, P., Zhou, Y. F., Wang, L. Z. (2022). Distributed networked microgrids power flow. *IEEE Transactions on Power Systems*, https://doi.org/10.1109/TPWRS.2022.3175933.

[6] Zhang, P. (2021). Networked Microgrids. Cambridge: Cambridge University Press.

[7] Hooshyar, A., El-Saadany, E. F., Sanaye-Pasand, M. (2016). Fault type classification in microgrids including photovoltaic DGs. *IEEE Transactions on Smart Grid*, 7: 2218–2229.

[8] Zhou, Y. F., Zhang, P., Yue, M. (2021). Reachable dynamics of networked microgrids with large disturbances. *IEEE Transactions on Power Systems*, 36: 2416–2427.

[9] IEEE standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces. IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003). Available at https://ieeexplore.ieee.org/servlet/opac?punumber=8332110.

[10] Hansen, A. D., Michalke, G. (2007). Fault ride-through capability of DFIG wind turbines. *Renewable Energy*, 32: 1594–1610.

[11] Giachetti, R. E., Peterson, C. J., van Bossuyt, D. L., Parker, G. W. (2020). Systems engineering issues in microgrids for military installations. *INCOSE International Symposium*, 30: 731–746.

[12] Ortiz, L., González, J. W., Gutierrez, L. B., Llanes-Santiago, O. (2020). A review on control and fault-tolerant control systems of AC /DC microgrids. *Heliyon*, 6: e04799.

[13] Buraimoh, E., Davidson, I. E., Martinez-Rodrigo, F. (2019). Fault ride-through enhancement of grid supporting inverter-based microgrid using delayed signal cancellation algorithm secondary control. *Energies*, 12: 3994.

[14] Rosas-Caro, J. C., García-Vite, P. M., Rodríguez, A., Mendoza, A., Alejo-Reyes, A., Cuevas, E., Beltran-Carbajal, F. (2021). Differential evolution based algorithm for optimal current ripple cancelation in an unequal interleaved power converter. *Mathematics*, 9: 2755.

[15] Wan, W. F., Bragin, M. A., Yan, B., Qin, Y. Y., Philhower, J., Zhang, P., Luh, P. B. (2020). Distributed and asynchronous active fault management for networked microgrids. *IEEE Transactions on Power Systems*, 35: 3857–3868.

[16] Wan, W. F., Zhang, P., Bragin, M. A., Luh, P. B. (2022). Cooperative fault management for resilient integration of renewable energy. *Electric Power Systems Research*, 211: 108147.

[17] Li, T., Sahu, A. K., Talwalkar, A., Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37: 50–60.

[18] Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D., Beaufays, F. (2018). Applied federated learning: Improving google keyboard query suggestions. arXiv preprint, 1812.02903.

[19] Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., C. Kiddon, Koneˇcn´y, J., Mazzocchi, S., McMahan H. B., et al. (2019). Towards federated learning at scale: System design. In: Proceedings of the 2nd Machine Learning and Systems Conference, Palo Alto, CA, USA.

[20] Yang, Q., Liu, Y., Chen, T. J., Tong, Y. X. (2019). Federated machine learning. *ACM Transactions on Intelligent Systems and Technology*, 10: 1–19.

[21] Bragin, M. A., Yan, B., Luh, P. B. (2020). Distributed and asynchronous coordination of a mixed-integer linear system via surrogate lagrangian relaxation. *IEEE Transactions on Automation Science and Engineering*, 18: 1191–1205.

[22] Bragin, M. A., Luh, P. B., Yan, J. H., Yu, N. P., Stern, G. A. (2015). Convergence of the surrogate Lagrangian relaxation method. *Journal of Optimization Theory and Applications*, 164: 173–201.

[23] Feng, F., Zhang, P., A Bragin, M., Zhou, Y. F. (2022). Novel resolution of unit commitment problems through quantum surrogate Lagrangian relaxation. *IEEE Transactions on Power Systems*, https://doi.org/10.1109/TPWRS.2022.3181221.

[24] Nikmehr, N., Bragin, M. A., Zhang, P., Luh, P. B. (2022). Computationally distributed and asynchronous operational optimization of droop-controlled networked microgrids. *IEEE Open Access Journal of Power and Energy*, 9: 265–277.
[25] Ray, S. (2019). A quick review of machine learning algorithms. In: Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India.
[26] Haddadpour, F. Mahdavi, M. (2019). On the convergence of local descent methods in federated learning. arXiv preprint, 1910.14425.
[27] Nguyen, H. T., Sehwag, V., Hosseinalipour, S., Brinton, C. G., Chiang, M., Vincent Poor, H. (2021). Fast-convergent federated learning. *IEEE Journal on Selected Areas in Communications*, 39: 201–218.
[28] Chen, M. Z., Poor, H. V., Saad, W., Cui, S. G. (2021). Convergence time optimization for federated learning over wireless networks. *IEEE Transactions on Wireless Communications*, 20: 2457–2471.
[29] Anderson, W. W. (2020). Resilience assessment of islanded renewable energy microgrids. Technical Report, Naval Postgraduate School.
[30] Hildebrand, J. P. (2020). Estimating the life cycle cost of microgrid resilience. Technical Report, Naval Postgraduate School Monterey Ca.
[31] Peterson, C. J., van Bossuyt, D. L., Giachetti, R. E., Oriti, G. (2021). Analyzing mission impact of military installations microgrid for resilience. *Systems*, 9: 69.
[32] Giachetti, R. E., Bossuyt, D. L. V., Anderson, W. W., Oriti, G. (2022). Resilience and cost trade space for microgrids on Islands. *IEEE Systems Journal*, 16: 3939–3949.
[33] Anglani, N., Oriti, G., Fish, R., Bossuyt, D. L. V. (2021). Design and optimization strategy to size resilient stand-alone hybrid microgrids in various climatic conditions. In: Proceedings of the 2021 IEEE Energy Conversion Congress and Exposition, Vancouver, BC, Canada.