IEEE Open Access Journal of
Power and Energy

Received 11 December 2019; revised 9 May 2020; accepted 22 May 2020. Date of publication 26 May 2020;
date of current version 9 June 2020.

Digital Object Identifier 10.1109/OAJPE.2020.2997665

Software-Defined Microgrid Control: The Genesis of Decoupled Cyber-Physical Microgrids

LIZHI WANG[1] (Student Member, IEEE), YANYUAN QIN[2] (Student Member, IEEE), ZEFAN TANG[1] (Student Member, IEEE), AND PENG ZHANG[1] (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA
[2]Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269 USA
(CORRESPONDING AUTHOR: P. ZHANG (p.zhang@stonybrook.edu)

This work was supported in part by the National Science Foundation under Grant ECCS-1831811.

ABSTRACT Nowadays, microgrid controllers are often embedded in specialized hardware such as PLC and DSP. The hardware-dependency and fit-and-forget design make it difficult and costly for microgrid controllers to evolve and upgrade under frequent changes such as plug-and-play of microgrid components. Furthermore, different distributed energy resources in a microgrid require customized controllers, leading to long development cycles and high operational costs for deploying microgrid services. To tackle the challenges, a software-defined control SDC) architecture for microgrid is devised, which virtualizes traditionally hardware-dependent microgrid control functions as software services decoupled from the underlying hardware infrastructure, fully resolving hardware dependence issues and enabling unprecedentedly low costs. A generic SDC prototype is designed to generate microgrid controllers autonomously in edge computing facilities such as distributed virtual machines. Extensive experiments verify that SDC outperforms traditional hardware-based microgrid control in that it empowers a decoupled cyber-physical microgrid and thus makes microgrid operations unprecedentedly affordable, autonomic, and secure.

INDEX TERMS Microgrids, software-defined control, decoupled cyber-physical microgrids, virtual machine, plug-and-play.

I. INTRODUCTION

MICROGRID is a paradigm shifting solution which enhances electricity resiliency and supports the ever-increasing integration of distributed energy resources (DERs) and energy storage at grid edges [1], [2]. However, it is oftentimes prohibitively difficult to build and operate a microgrid mainly due to the hardware-dependence of microgrid protection, automation and control (PAC) where the microgrid intelligence resides. Especially, the existing microgrid control is always implemented on specific hardware such as DSP or PLC [3], [4], making it difficult and costly to evolve and update when the microgrid configuration changes and thus the controller parameters have to be retuned [5], [6]. In the state-of-the-practice, the high CAPEX and OPEX of hardware controllers multiplying the large number of DERs prohibit nearly any redundancy or backup in microgrid controllers [7], [8]. Thus, any failure in microgrid controller can lead to severe impacts on both microgrid and the main grid. Adding to those challenges is the fact that there is no universally applicable tool for designing microgrid control efficiently and optimally, making the designing and deployment of new hardware controllers daunting tasks.

Over the years, despite a plethora of literature in improving the DER control in microgrids (e.g. droop control, PQ control and V/f control [9]), a majority of research efforts [10], [11] only focused on the simulation-based performance analysis for control algorithms, assuming a hardware-dependent control architecture. Few of the existing publications address how to improve the response of a microgrid to hardware anomalies such as failures or being sabotaged. Recently, the software-defined networking (SDN) is attracting increasing attentions from the microgrid community as it offers a programmable,

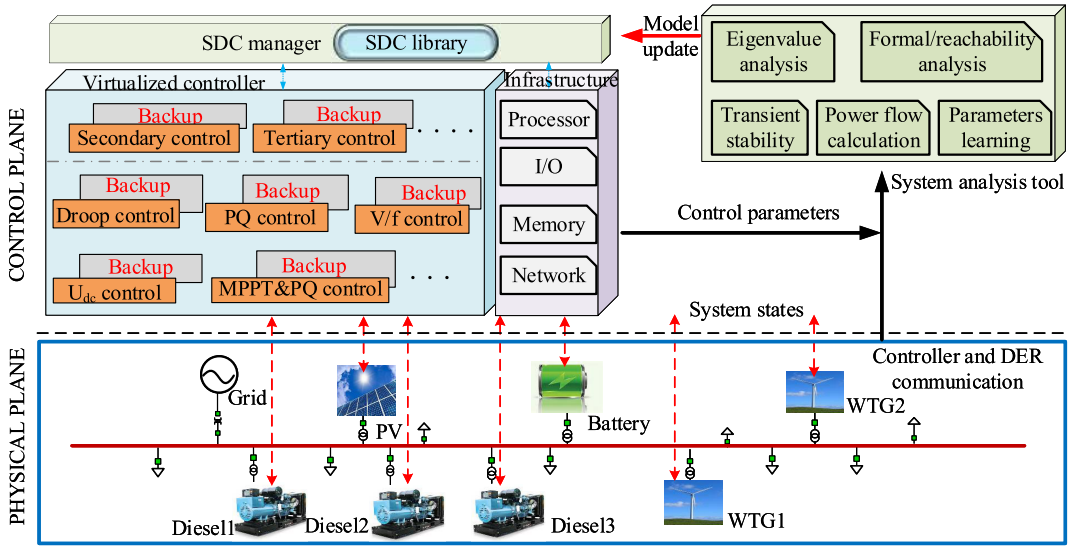


FIGURE 1. **SDC-enabled microgrid architecture.**

flexible and reliable solution to operate the microgrid, supporting diverse quality-of-service (QoS) requirements and making it much easier to develop new applications and enable fast innovation in microgrid [12]. However, although a few works have been done on developing SDN-enabled microgrid or even networked microgrids [12], [13], the existing literature is unfortunately largely silent on the topic of developing software-defined microgrid controls. Part of the reason for this stems from the fact that achieving the transition – replacing hardware-dependent controllers with software-based virtual ones – is not straightforward. It was unclear how to guarantee the software-based controllers perform as well as the hardware-dependent ones, and how to provide the redundancy for controllers with software-based solutions.

In this paper, we present a software-defined control (SDC) architecture for microgrid. It decouples the hardware infrastructure with microgrid control functions. Decoupling software from dedicated hardware enables easier modifications, managements and updates. Virtualization allows multiple independent users to efficiently use computational and network resources (e.g., processing power and communication bandwidth) by abstracting them into logical units. Specifically, the controllers are implemented in software, and run on white-box hardware platforms via an ultra-fast network [14]. As an illustration purpose, a primary DER microgrid controller, droop controller, is virtualized and implemented in the SDC architecture to demonstrate the system's performance. Extensive experiments in an RTDS environment verify that virtualization of microgrid controllers can get rid of the restriction of hardware implementation and at the same time achieve the goal of hot standby [15] and seamless switching of controllers at zero cost. The contributions of this paper are threefold:

- A novel SDC architecture is devised for microgrid, where the concept and validation of virtualized microgrid controller are addressed. A generalized control frame is designed in the SDC architecture, which automatically generates controllers for microgrid, providing high reliability and easily-deployable redundancy.
- A discrete model of virtual droop control, which incorporates double-loop controllers using the trapezoidal rule, is derived as an example for implementing interfaces between DERs and virtualized controllers.
- A testbed is built in an RTDS environment to evaluate the SDC's performance. Extensive experimental results validate the robustness, plug-and-play capability and reliability of the SDC-based system. Results also verify that the software-defined controllers perform better when microgrid encounters hardware failures.

The rest of this paper is organized as follows: Section II presents the SDC-enabled microgrid architecture, the SDC's work flow and implementation, and the benefits of using SDC for microgrid. The derived discrete model of virtual droop control is presented in Section III. Section IV provides the experimental results, and Section V concludes the paper.

II. SDC-ENABLED MICROGRID

Virtualization provides a promising opportunity to develop novel applications flexibly in microgrid. To implement software-based control functions through virtualization on general-purpose servers, a critical challenge is to design a framework for SDC with no performance degradation. In this section, the devised SDC-enabled microgrid architecture is first presented, and this is followed by the description of SDC's work flow and implementation. The benefits of using SDC for microgird are also provided.

A. ARCHITECTURE OF THE SDC-ENABLED MICROGRID

The high-level design of the SDC-enabled microgrid architecture is illustrated in Fig. 1, where the physical and cyber layers are fully decoupled. The control plane and its corresponding functions are pushed to the edge of the microgrid,

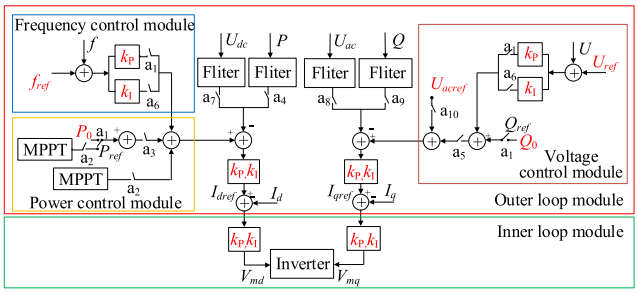


FIGURE 2. Framework of a generalized DER controller.

and the virtual controllers run in local servers. The components in the physical plane include various DERs such as solar panels, wind turbines, diesels and storages, and others like smart meters, loads, and transformers. Each DER communicates with its controller through an IP-address-assigned interface, which can adopt different communication protocols to transfer DER's data such as control signals.

The measurements from the physical plane (e.g., power generation, load, etc.) are transferred to the control plane via a communication network, and are used for various tasks such as eigenvalue analysis, power flow calculation, transient analysis, formal/reachability analysis, and parameters learning [16]. The results obtained from the analyses are used for designing control parameters and updating controllers' models. On the top of the architecture is the SDC manager. It manages different virtual controllers in the control plane, such as implementing control functions, managing the handover of controllers, and providing design parameters for controllers. A critical component for the SDC manager is the SDC library, which consists of parameters from a generalized DER controller.

The framework of a generalized DER controller is given in Fig. 2 [17]. It consists of frequency, power, and voltage control modules, and an inner loop module. A polymorphic switch a_1 is used to select the active power reference, and its state is initially off. Each controller's parameters, i.e., the references that are marked in red in Fig. 2, are tuned when the controller is created. The tuning process is based on the analysis of the system with the goal of guaranteeing the system's stability. The combinations of different modules and switches' states generate different control functions. This characteristic ensures the software-defined controllers inherit from the generalized controller easily.

The relationship between the control objective and the switches' states can be described as follows:

$$\boldsymbol{c} = \boldsymbol{b} \cdot \boldsymbol{a}, \tag{1}$$

where $\boldsymbol{c} = \{c_1, c_2, \ldots, c_N\}$ denotes the set of N microgrid controllers, $\boldsymbol{b} = \{b_1, b_2, \ldots, b_N\}$ is the set of binary numbers that control the states of different switches in the N microgrid controllers, and $\boldsymbol{a} = \{a_1, a_2, \ldots, a_{10}\}$ is the set of the 10 switches in the generalized control architecture in Fig. 2. For instance, if bit i of b_n is one, switch a_i will be turned on to implement the function of controller c_n. In particular, for the droop controller, the binary number is $b_1 = (0100110101)_2$,

TABLE 1. Control configuration of different DERs.

Operations	Inverter type	Controllers (Objectives)	Binary number
Grid-connected	Grid following	P/Q (Control $P\&Q$)	$(1100110101)_2$
Islanded	Grid forming	Droop (Control $f\&U$)	$(0100110101)_2$
		V/f (Control $f\&U$)	$(0100100101)_2$
		U_dc&U_ac Control $U_{dc}\&U_{ac}$	$(1111110010)_2$
		P&U_ac Control $P\&U_{dc}$	$(1100111001)_2$
	Grid following	P/Q Control $P\&Q$	$(1100110101)_2$
		MPPT&P/Q Control $P\&Q$	$(1000110101)_2$

which means the switches a_1, a_3, a_5, a_6 and a_9 are turned on. DERs typically operate as grid-following sources using the P/Q control when microgrid works in grid-connected mode. When microgrid works in islanded mode, some DERs operate as grid-forming sources to actively control their frequency output, making it possible for them to naturally support the system frequency while sharing a portion of the load change [18]. DERs operate as grid-following sources to track the voltage angle of the microgrid to control their output. Table 1 illustrates the control configurations of DERs based on the features of DERs' functions, which provides the specific guidance on how to design DER controllers for SDC users based on the generalized controller. The configuration table is stored in SDC library, which provides SDC manager reference to decide which control (e.g., PQ control, droop control, V/f control, etc.) should be selected from the library. Meanwhile, a backup controller is also established to provide the control redundancy for microgrid, and will be discussed later.

To implement new virtualized control methodologies, flexible computational resources such as memories and processors have to be virtualized and allocated to the new controller. This is achieved using technologies such as Hypervisor or Container by the SDC manager. Once the new methodology is created, the description on the virtualized controller will be stored in the SDC library, which contains information including its constitutions and the computational resource requirements. Meanwhile, the communication network resource information such as the bandwidth and delay is obtained by the SDC manager via the interface with the communication network. New communication channels will be created for new methodologies once the requirements of computational and communication network resources are satisfied. It is to be noted that the handshake and communication of different units are managed by the SDC manager.

B. SDC's WORK FLOW

SDC provides the flexibility to develop new applications such as optimization, distributed control and system parameters learning. Based on the demand of microgrid operation,

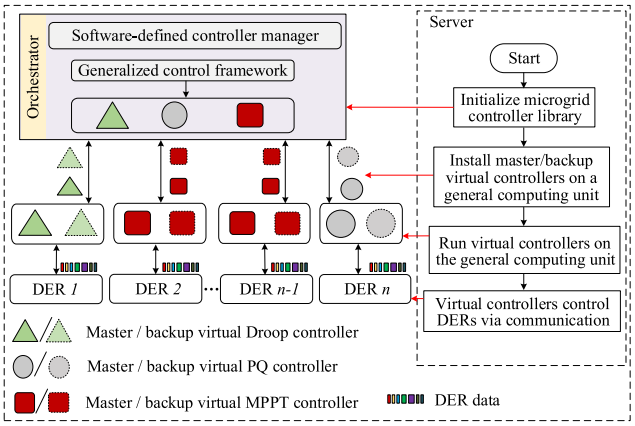


FIGURE 3. **SDC's work flow.**

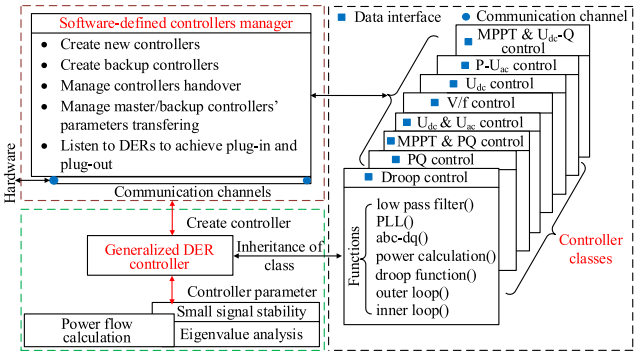


FIGURE 4. **An overview of the implementation for SDC in microgrid.**

different software-defined controllers can be installed for different DERs. For example, an islanded microgrid needs diesels or batteries to stabilize the system frequency, and therefore the software-defined droop controllers can be created to achieve the goal. The procedures of implementing the software-defined controllers are described as follows:

- Step 1: Select controllers from the SDC library, and set the controllers' corresponding parameters based on each DER's requirements.
- Step 2: The SDC manager installs the software-defined controllers on general computing platforms, i.e., servers. As the controllers are installed in software, one or multiple backup controllers can be installed to provide the redundancy. All the controllers are installed on servers, which can be close to or far away from DERs.
- Step 3: Run the virtual controllers on servers. Each master controller receives its corresponding DER's states and sends out control signals to the DER. Meanwhile, each backup controller runs in the "hot standby" mode, which improves the system's modularity and robustness.
- Step 4: A software-defined controller controls its DER via a data transmission network. For instance, the three phases of the voltage or current from each DER can be transferred to the corresponding controller. Various encryption protocols can be used for the data transmission to ensure its reliability and security.

The devised SDC's work flow is illustrated in Fig. 3. The SDC manager is in charge of the events including initialization, scaling, termination, and updating. Once the SDC manager detects that a new controller for a certain DER needs to be created, a new connection between the controller and the DER will be initialized. The SDC manager abstracts the physical resources of each controller, and creates the corresponding functions within the virtualized infrastructure. It ensures that the life cycle of each controller is independent of hardware platforms such as the DERs with standardized interfaces between the controllers and DERs. A software-defined controller can run either on a physical server, or on a virtual machine (VM).

Network protocols such as TCP/IP can be used to establish low-latency and loss-tolerating connections between DERs and controllers. To guarantee the performance of DER controls, the allowed time delay of communication between DERs and controllers should be within tens of microseconds [19], which can be achieved using the existing widely-used network switches.

C. IMPLEMENTATION OF SDC IN MICROGRID

The implementation of SDC in microgrid is described in this subsection. As shown in Fig. 4, the implementation process consists of three key components. The first one is the implementation of SDC manager. The SDC manager has the ability of creating new master and backup controllers, managing controller handovers, transferring parameters for master and backup controllers, and monitoring DERs for plug-in and plug-out. The second component is the establishment of a generalized DER controller library. This library is in charge of creating different DER controllers. A variety of controllers share some common functions, which can be inherited from the generalized controller, e.g., functions like Phase Lock Loop (PLL), coordinate transformation, low pass filtering, and double-loop control. The last component is the virtualization of software-defined controllers, which can be designed as classes based on the characteristics of object-oriented programming, and are inherited from the generalized DER controller.

The communications between DERs and virtual controllers are managed by the SDC manager. Specifically, when the SDC manager receives a request for creating a new controller for a DER, the function of *create_new_controller* is executed. The DER's IP address and port are sensed by the SDC manager, and the IP address and port of the GCU, on which the new created controller is implemented, are transferred to the DER. Meanwhile, the SDC manager sends commands to the generalized DER controller to request a new controller class. The controller then enters the *listening* mode after being initialized and connected to the DER. At this state, the controller is receiving any data packet whose destination IP and port match the controller, respectively [20]. Once a packet arrives, the controller begins to send corresponding control signals to the DER, e.g., when the SDC manager

receives a request from a battery storage, a software-defined droop controller is created from the generalized DER controller, and corresponding control signals are generated. The server then enters the *sending* mode and starts to send out control signals whose destination IP and port are the IP and port of the DER, respectively. The handover function is executed when the SDC manager detects one controller stops working. The IP and port of that controller will be transferred to a backup controller, and the corresponding control parameters will also be set for the backup controller.

D. BENEFITS AND COST OF USING SDC FOR MICROGRID

Virtualized controllers running on the commodity hardware use virtualized computational resources to facilitate the implementation of the SDC framework. A virtualized controller has the ability to produce optimal results with minimum resources. Integrating the intervention into existing structures greatly requires the replacement of existing controllers with virtualized resources and the establishment of connections between virtualized controllers and DERs. Compared to the existing hardware-dependent control architectures, SDC offers the following benefits for microgrid:

- The separation of software from hardware infrastructure allows the software to evolve independently, which provides microgrid with great flexibility such as the flexible deployment of control functions for different requirements in microgrid.
- The dynamic service provided by the SDC manager offers more intensive monitoring for controllers.
- The generalized DER control framework offers a high efficiency and great convenience to develop specific new controllers for DERs.
- The convenient creation and deployment of software-defined backup controllers provide high reliability for microgrid to operate with redundancy.

III. VIRTUAL DROOP CONTROLLER

The communication process between a DER and a virtual controller can be summarized as follows: 1) the controller establishes communication with the DER using information provided by the SDC manager; 2) at the DER side, measurements such as three phases of the output current and voltage of DER are sampled and transferred to the controller; and 3) control signals such as those used to generate SPWM are sent back to the DER.

A. VIRTUAL DROOP CONTROLLER MODEL

Discrete control algorithms deployed in the processor are the main components to operate DERs' controllers. In [21]–[23], different distributed discrete secondary control approaches are proposed to minimize frequency and voltage deviations and ensure accurate active and reactive power sharing for either radial- or mesh-structured microgrids. However, the primary discrete controllers are not addressed in these papers. A discrete-time mathematical model and an analytical framework for a droop-controlled inverter-based microgrid are presented in reference [24], where the double-loop controllers are not included. Moreover, the PI regulator is shown using the z-transformation, which however cannot be applied directly for software evolutions. In this paper, a droop controller considering the power calculation, frequency control, voltage-current loop control is presented in detail using the trapezoidal rule, which can be directly applied with the software. Discrete models for other controllers, such as PQ and V/f controls, can be generated in a similar way.

For a droop control, the frequency and voltage magnitudes at time k can be obtained as follows:

$$f(k) = f^* - m(P(k) - P^*) = f^* - m\Delta P(k), \tag{2}$$

and

$$E(k) = E^* - n(Q(k) - Q^*) = E^* - n\Delta Q(k), \tag{3}$$

where $f(k)$ and $E(k)$ are the frequency and voltage magnitudes at time k, respectively. $P(k)$ and $Q(k)$ are the discrete samples of the active and reactive powers, respectively. f^* and E^* are the frequency and voltage references, respectively; and P^* and Q^* are the active and reactive power references, respectively. $\Delta P(k)$ and $\Delta Q(k)$ are the corresponding active and reactive power error inputs for the droop controller, and m and n are frequency and voltage droop coefficients, respectively.

For a converter-based DER, the droop control is typically implemented using a double-loop control diagram, where the outer control loop is designed to provide current references for inner loop, and the inner loop generates modulation waves for SPWM generation [25]. Let $v_d(k)$ and $v_q(k)$ be the voltage components in dq coordinate transformed from the three-phase voltage in abc coordinate at time k. Then, $E_r(k)$ can be defined as follows:

$$\begin{aligned} E_r(k) = {} & v_d(k)(1 + \sum\nolimits_{n=1}^{4}(-1)^n \frac{1}{2n}\Phi^{2n}(k)) \\ & + v_q(k)(\Phi(k) + \sum\nolimits_{n=1}^{4}(-1)^n \frac{1}{2n+1}\Phi^{2n+1}(k)), \end{aligned} \tag{4}$$

where $\Phi(k)$ is the output of the PLL function at time k, and can be calculated recursively as follows:

$$\Phi(k+1) = \Phi(k) + (w(k) + w(k+1))T/2, \tag{5}$$

where T is the sampling time, and is set at 0.2 ms in this study. $w(k)$ can be calculated recursively as follows:

$$\begin{aligned} w(k+1) = {} & k_p E_r(k+1) \\ & + k_i T((E_r(0) + E_r(k+1))/2 + \sum_{n=1}^{k} E_r(n)), \end{aligned} \tag{6}$$

where k_p and k_i are the PI parameters.

The phase of the voltage at time k, $\theta(k)$, in discrete domain can be calculated recursively as follows:

$$\theta(k+1) = \theta(k) + 2\pi(f(k) + f(k+1))T/2, \tag{7}$$

where $f(k)$ can be obtained from (2).

With $E_r(k)$ and $\theta(k)$ obtained from (4)-(7), the error of the input voltage for the outer loop can be calculated as follows:

$$\begin{cases} \Delta v_{dref} = E_r(k)(1 + \sum_{n=1}^{4}(-1)^n \frac{1}{2n}\theta^{2n}(k)) - v_d(k)) \\ \Delta v_{qref} = E_r(k)(\theta(k) + \sum_{n=1}^{4}(-1)^n \frac{1}{2n+1}\theta^{2n+1}(k)) \\ \qquad -v_q(k). \end{cases} \tag{8}$$

With v_{dref} and v_{qref} obtained from (11), the current references for the inner loop can be calculated as follows:

$$\begin{cases} i_{dref}(k+1) = k_{pv_d}(\Delta v_{dref}(k+1)) + k_{iv_d}T(\Delta v_d(0) \\ \qquad +\Delta v_d(k+1)/2 + \sum_{n=1}^{k}\Delta v_d(n)) \\ i_{qref}(k+1) = k_{pv_q}(\Delta v_{qref}(k+1)) + k_{iv_q}T(\Delta v_q(0) \\ \qquad +\Delta v_q(k+1)/2 + \sum_{n=1}^{k}\Delta v_q(n)), \end{cases} \tag{9}$$

where $k_{pv_d}, k_{iv_d}, k_{pv_q}$ and k_{iv_q} are PI parameters, and $\Delta i_{dref}(k) = i_{dref}(k) - i_d(k)$. Then, the voltage modulation references v_{dm} and v_{qm} are obtained as follows:

$$\begin{cases} v_{dm}(k+1) = v_d(k+1) + k_{pi_d}(\Delta i_{dref}(k+1)) \\ \qquad +k_{ii_d}T(\Delta i_d(0) + \Delta i_d(k+1)/2 \\ \qquad +\sum_{n=1}^{k}\Delta i_d(n)) + i_q(k+1)wL \\ v_{qm}(k+1) = v_q(k+1) + k_{pi_q}(\Delta i_{qref}(k+1)) \\ \qquad +k_{ii_q}T(\Delta i_q(0) + \Delta i_q(k+1)/2 \\ \qquad +\sum_{n=1}^{k}\Delta i_q(n)) - i_d(k+1)wL, \end{cases} \tag{10}$$

where $k_{pi_d}, k_{ii_d}, k_{pi_q}$ and k_{ii_q} are PI parameters. $i_d(k)$ and $i_q(k)$ are the current components in dq coordinate transformed from the three-phase current in abc coordinate at time k. w is the frequency reference, i.e., 120π, and L is the inductance of the filter in the droop controller.

A variety of functions in a droop control such as inner loop and outer loop, can be used for other software-defined controls. Similar methodologies can be applied for the derivation of discrete models for other software-defined controls such as PQ and V/f controls.

B. TIME DELAY FACTORS IN VIRTUAL DROOP CONTROLLER

In the SDC architecture, the flow of the information consists of three steps: 1) measurement, 2) communication among controllers, and 3) execution. In the measurement process, transmitting the incremental frequency in the f/P droop control from a DER to a virtualized controller can introduce a time-delay τ_m in the feedback path. The execution process uses an interface to send control signals from a virtualized controller to a DER. The control delay τ_e, which can affect the stability of the distributed system, should be considered. The impact of the control algorithms' iterations in the SDC can be neglected due to the large computing capacity of the SDC. The delay in the closed-loop τ, which is the sum of τ_m and τ_e, forms the total delay of the SDC-enabled microgrid. The small signal model [26] of the time-delayed microgrid is

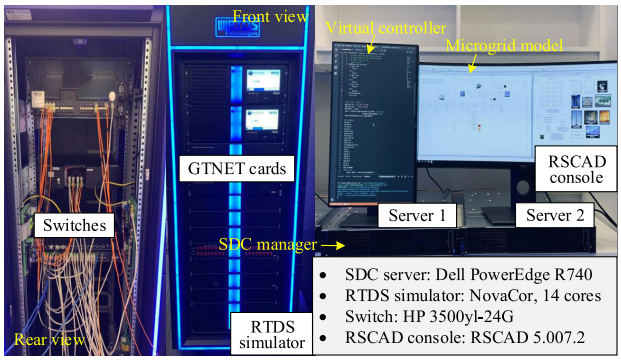


FIGURE 5. **The cyber and physical components in the simulation testbed.**

as follows:

$$\begin{cases} \Delta\dot{x}_{MG}(t) = A_{MG0}\Delta x_{MG}(t) + B_{MG0}\Delta y_{MG}(t) \\ \qquad \sum_{i=1}^{S}(A_{MGi}\Delta x_{MG}(t-\tau_i) \\ \qquad +B_{MGi}\Delta y_{MG}(t-\tau_i)) \\ \qquad 0 = C_{MG0}\Delta x_{MG}(t) + D_{MG0}\Delta y_{MG}(t) \end{cases} \tag{11}$$

where $\tau_i = \tau_{mi} + \tau_{ei} (i = 1, \ldots, m)$ refers to the time delay constants, and $0 < \tau_1 < \ldots < \tau_1 \triangleq \tau_{max}$. In the SDC-enabled architecture, the measurement and execution delays are trivial compared with the switching and communication delays caused by master and backup controllers. The delay's impact will be further discussed in Section IV.

IV. TEST AND VALIDATION OF SDC

A. EXPERIMENTAL SETUP

To test the SDC's performance, a real-time simulation testbed is built in an RTDS environment. As illustrated in Fig. 5, this testbed consists of the RTDS hardware, its auxiliary facilities such as the GTNET cards, and the servers. A console PC is used to develop and compile the microgrid model in the RSCAD, a power system simulation software designed to interact with the RTDS hardware. The GTNET cards can be used to transmit data between RTDS and external equipment through a LAN/WAN using protocols such as the GTNET-SKT (Socket) Protocol. The RTDS and servers are connected through switches and a campus network. Meanwhile, each server has a specific port, which is linked to the campus network and can be used for the console PC to access the server. In this study, the simulator in RDTS has 16 cores for physical-layer simulation running in real time. It provides Gigabit Ethernet ports for all IP-based communications including data transmissions between RTDS and external servers.

A typical microgrid shown in Fig. 6 is used to test and validate the SDC's performance. This test system contains two diesel units, one photovoltaic (PV) unit, one wind turbine, and one battery. The microgrid can operate in islanded or grid-connected mode, depending on whether the circuit breaker S1 is open or closed. The detailed parameters of the microgrid can be found in Appendix.

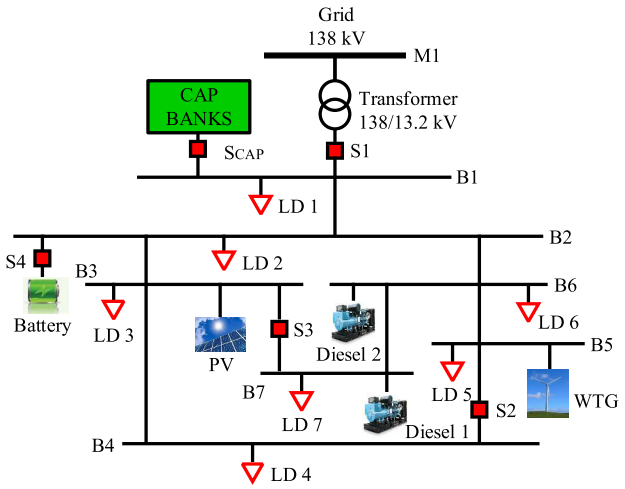


FIGURE 6. **A typical microgrid used to test and validate the SDC's performance in this paper [29].**

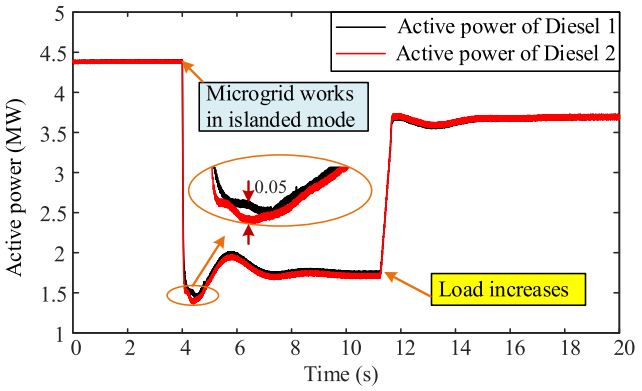


FIGURE 7. **Comparison of the two diesels' active powers.**

In the simulation, the sampling time step is set at 2 μs for converter-based components and 50 μs for other components in microgird. The sampling rate of transmission data is 600 packets per second. A software-defined virtual droop control is designed for the battery in Fig. 6, whose parameters are given in table 2 in Appendix V. The battery is connected to Bus 2 through a 0.48/13.2 kV step-up transformer. As the current constraint is commonly set at 1.1-1.3 times of the rated current [27], [28], a constraint of 1.25 p.u. is set for the converter's current in the test cases.

B. COMPARISON WITH TRADITIONAL DROOP CONTROLLER

The performances of the SDC and a traditional droop controller are compared in this subsection. In this test case, the two diesels share the same droop factor and capacity. Diesel 1 is controlled by a software-defined droop controller running on a remote server, and Diesel 2 is controlled by a traditional droop controller, which runs in the RTDS.

The operation of the microgrid works as follows: Initially, the microgrid is connected to the main grid, and the battery and capacitor banks are disconnected. At time $t = 4$ s, the switch S1 is open, making the microgrid operate in islanded mode. Meanwhile, the two diesels immediately start to regulate their output powers for balancing the power generation with the load consumption. At time $t = 11.5$ s, the active power of Load 6 (see Fig. 6) suddenly increases from 1 MW to 5 MW.

The comparison results of the two diesels' active powers are given in Fig. 7. It can be observed that:

- From time $t = 4$ s to around $t = 7$ s, the two diesels have almost the same dynamic performances. The maximum difference between the active powers of the two diesels is only around 0.05 MW.
- At time $t = 11.5$ s, the two diesels increase their active power to an almost same degree during the regulation of the active power output. It verifies that the software-defined droop controller performs as well as a traditional hardware-dependent one; there is almost no performance degradation.

C. IMPACT OF AN SDC-ENABLED DER ON MICROGRID

In this test case, we demonstrate how the microgrid reacts when an SDC-enabled DER is plugged into the system. Initially, the microgrid operates in islanded mode with the battery disconnected. At time $t = 1.2$ s, the battery is plugged into the system. Meanwhile, the SDC manager receives a handshake request, and immediately generates and installs a virtual droop controller on the remote server. A new connection is built between the battery and the virtual droop controller. The three phases of voltage and current of the battery's output are sampled and transferred to the virtual droop controller on the server through the campus network. At time $t = 6.8$ s, the active power of Load 6 suddenly increases from 1 MW to 5 MW.

The current and voltage responses of the battery before and after the battery is plugged in the microgrid are illustrated in Fig. 8. It can be seen that:

- The virtual droop controller immediately starts to control the battery when the battery is plugged in the microgrid at time $t = 1.2$ s. The current of the battery is stabilized in a very short time.
- The voltage of the battery has a little fluctuation but becomes stable within only around 0.05 s after the battery is connected to the microgrid.
- When Load 6 increases at time $t = 6.8$ s, the voltage magnitude of the battery maintains constant, while the current magnitude of the battery increases. It indicates that the SDC-enabled battery has started to participate in the power sharing within the microgrid.

D. IMPACT OF SWITCHING BETWEEN MASTER AND BACKUP SOFTWARE-DEFINED CONTROLLERS RUNNING ON THE SAME SERVER

A backup controller can be implemented in microgrid to provide redundancy. In this subsection, the impact of the switching between a master controller and a backup controller running on the same server is evaluated. Specifically, at time

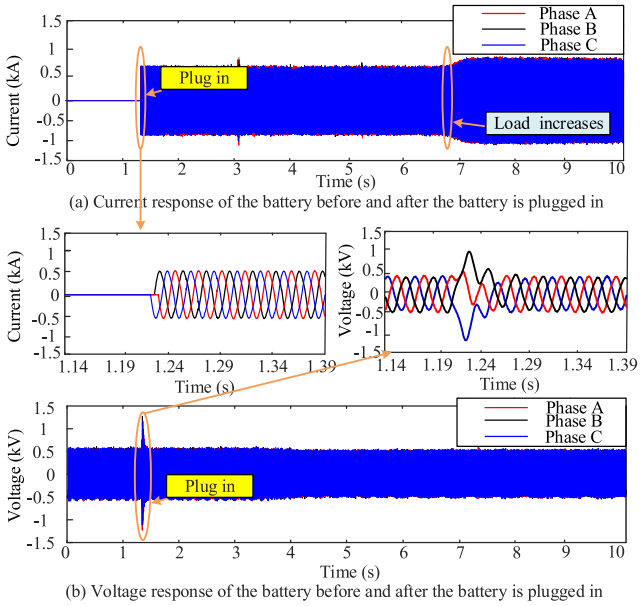


FIGURE 8. Current and voltage responses of the battery before and after the battery is plugged in the microgrid. (a) Current response of the battery. (b) Voltage response of the battery.

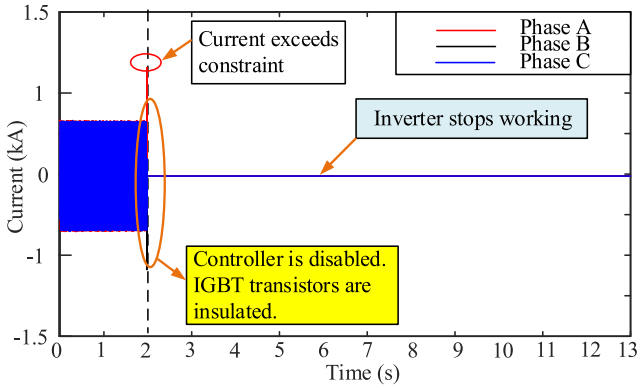


FIGURE 9. Current response of the battery with the traditional hardware-dependent controller before and after the controller fails.

$t = 2$ s, the master controller fails, meaning it cannot process data and achieve the droop function. The battery's current response using a traditional hardware-dependent droop controller with no backup controller is illustrated in Fig. 9. It can be seen that when the traditional droop controller fails, no control signals are sent back to the battery because of the lack of redundancy. As there is no feedback control signal, no modulation wave can be used to generate the PWM signals. The insulated-gate bipolar transistors (IGBTs) in the converter are isolated due to the overcurrent at time $t = 2$ s, and the converter stops working.

The current and voltage responses of the battery with the master and backup virtual droop controllers are given in Fig. 10. At time $t = 2$ s, the master controller fails and the backup controller immediately starts to work. Meanwhile, the new connection between the battery and the backup controller is established. The states of the master controller are

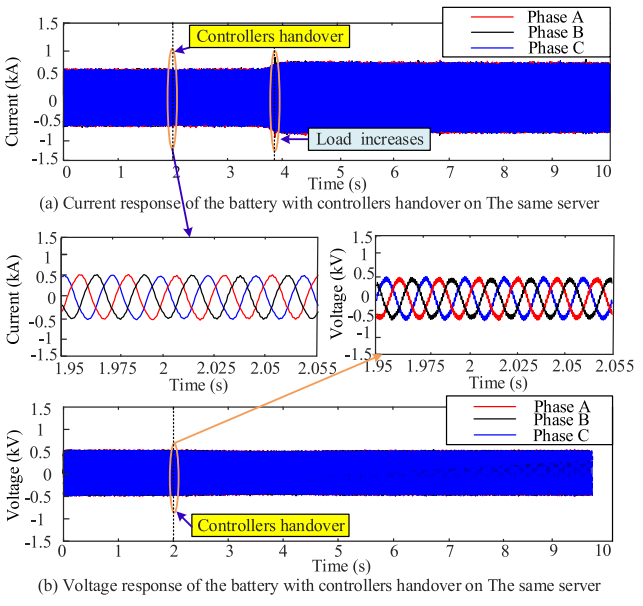


FIGURE 10. Current and voltage responses of the battery with master and backup virtual controllers running on the same server before and after the master controller fails. (a) Current response of the battery. (b) Voltage response of the battery.

transferred to the backup controller, making the battery be controlled by the backup controller. At time $t = 3.8$ s, the active power of Load 6 is adjusted from 1 MW to 5 MW. From Fig. 10, it can be seen that there are no overcurrent or fluctuation issues during the controllers handover when the controllers are running on the same server.

E. IMPACT OF SWITCHING BETWEEN MASTER AND BACKUP SOFTWARE-DEFINED CONTROLLERS RUNNING ON DIFFERENT SERVERS

The impact of the switching between master and backup software-defined controllers running on different servers is evaluated in this subsection. Specifically, the master software-defined droop controller runs on server 1, and the backup software-defined droop controller runs on server 2. At time $t = 2$ s, the master controller on server 1 fails. The states of the master controller are transferred to the backup controller on server 2. The current and voltage responses of the battery before and after the master controller fails are illustrated in Fig. 11.

It can be observed that the current magnitude becomes slightly higher, i.e., increasing from 0.5 kA to around 0.8 kA, during the controllers handover. This is mainly caused by the communication latency between two servers. However, the increased current is still within the limit of constraints, promising the normal operation of the system during the handover between controllers on different servers.

F. IMPACT OF DELAY

For an SDC-enabled system, a critical issue is the impact of the communication latency on a software-defined controller's

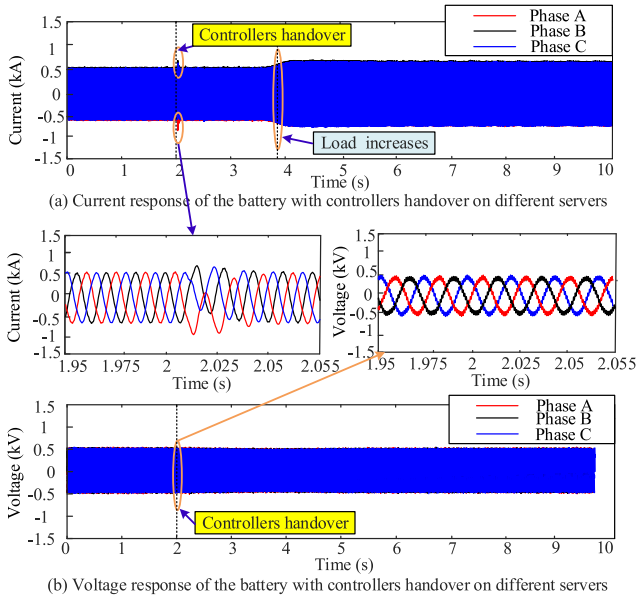


FIGURE 11. **Current and voltage responses of the battery with master and backup virtual controllers running on different servers before and after the master controller fails. (a) Current response of the battery. (b) Voltage response of the battery.**

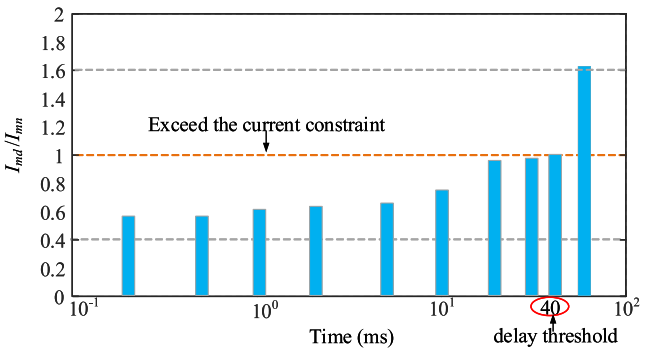


FIGURE 12. **System's performance with different time delays.**

performance. Delays caused by measurement, execution, and algorithm iteration can have different impacts on the system's performance. In this paper, it is found that the measurement and execution cause very small delays due to the high rate of date sampling and large communication capability of the test bed. Fig. 12 illustrates the system's performance with different time delays caused by the controllers' handover in different servers, where the current constraint of the battery is denoted as I_{mn}, and the maximum current magnitude of the battery during the controllers handover with a certain time delay is denoted as I_{md}.

It can be seen that the nominal delay caused by the controllers' handover in different servers is tolerated, which indicates that the system is still stable. When different time delays are introduced, the current of the battery is below the constraint of the converter's current when the delay is less than 40 ms. Note that the communication delay with existing Ethernet switches is only several milliseconds, which is much smaller than 40 ms.

TABLE 2. **Power loads (KVA) at different buses in Fig. 6.**

Bus	Phase A	Phase B	Phase C	pf
1	506	506	506	0.9
2	367	367	367	0.95
3	344	344	344	0.9
4	356	356	356	0.9
5	325	625	100	0.95
6	333	333	333	0.95
7	275	625	150	0.95

TABLE 3. **DERs' generation(MW/MVA) and control parameters.**

Bus	DER	Control method	m_p	K_{pv}	K_{iv}	k_{pi}	K_{ii}
1	Capacitors (0.5) banks	N/A	N/A	N/A	N/A	N/A	N/A
2	Battery (0.5)	Droop	0.6	0.1	5	0.05	0.3
3	PV (1.74)	PQ	N/A	1	5	0.025	0.5
5	WTG (2)	PQ	N/A	1	10	0.3	10
6	Diesel (3)	Droop	0.2	N/A	N/A	N/A	N/A
7	Diesel (3)	Droop	0.2	N/A	N/A	N/A	N/A

V. CONCLUSION

In this paper, a software-defined control architecture is devised for microgrid. Compared with the existing microgrid control systems that heavily rely on hardware infrastructure which is inconvenient and costly to evolve and upgrade, the presented SDC architecture provides robustness and plug-and-play capability for DER controllers. It is also capable of supporting a variety of applications such as secondary and tertiary controls, and other advanced control schemes in microgrid. The experimental results in an RTDS environment validate that SDC is a universal and plug-and-play platform for microgrid, where new DER controllers can be easily and flexibly defined and deployed, and high redundancy can also be provided to deal with DER controllers' failures.

APPENDIX

Table 2 gives the three-phase power loads at different buses. Loads on buses 1-4 are modeled as switched RL passive loads while loads on buses 5-7 are modeled as non-switchable dynamic loads representing the critical loads in the system. The DERs' generation and control schemes are summarized in Table 3. The diesels and the battery use droop control to achieve frequency and power sharing when the microgrid operates in islanded mode, while the PV and wind turbine use the PQ control. For more details on the microgrid, readers are referred to [29].

REFERENCES

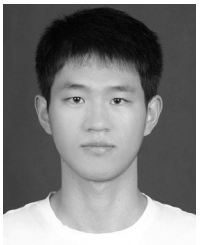
[1] R. R. Kolluri, I. Mareels, T. Alpcan, M. Brazil, J. de Hoog, and D. A. Thomas, "Power sharing in angle droop controlled microgrids," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4743–4751, Nov. 2017.

[2] Y. Mohamed and E. F. El-Saadany, "Adaptive decentralized droop controller to preserve power sharing stability of paralleled inverters in distributed generation microgrids," *IEEE Trans. Power Electron.*, vol. 23, no. 6, pp. 2806–2816, Nov. 2008.

[3] K. De Brabandere, K. Vanthournout, J. Driesen, G. Deconinck, and R. Belmans, "Control of microgrids," in *Proc. IEEE Power Eng. Soc. Gen. Meeting*, vol. 7, Jun. 2007, pp. 1–7.

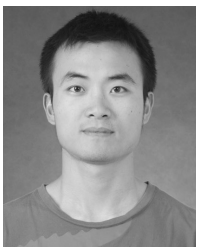
[4] P. Danzi, M. Angjelichinoski, C. Stefanovic, T. Dragicevic, and P. Popovski, "Software-defined microgrid control for resilience against denial-of-service attacks," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5258–5268, Sep. 2019.
[5] T. Dragi evi ć, X. Lu, J. C. Vasquez, and J. M. Guerrero, "DC microgrids—Part I: A review of control strategies and stabilization techniques," *IEEE Trans. Power Electron.*, vol. 31, no. 7, pp. 4876–4891, Jul. 2016.
[6] M. N. D. P. D. Bhuyar and A. Jadhav, "Review on IoT based smart solar photovoltaic plant remote monitoring and control unit," *Int. J.*, vol. 3, no. 3, pp. 1–5, 2018.
[7] P. Basak, S. Chowdhury, S. H. N. Dey, and S. P. Chowdhury, "A literature review on integration of distributed energy resources in the perspective of control, protection and stability of microgrid," *Renew. Sustain. Energy Rev.*, vol. 16, no. 8, pp. 5545–5556, Oct. 2012.
[8] Y. Li, D. M. Vilathgamuwa, and P. C. Loh, "Design, analysis, and real-time testing of a controller for multibus microgrid system," *IEEE Trans. Power Electron.*, vol. 19, no. 5, pp. 1195–1204, Sep. 2004.
[9] P. Piagi, "Microgrid control," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Wisconsin–Madison, Madison, WI, USA, 2005.
[10] W. Liu, W. Gu, Y. Xu, Y. Wang, and K. Zhang, "General distributed secondary control for multi-microgrids with both PQ-controlled and droop-controlled distributed generators," *IET Gener., Transmiss. Distrib.*, vol. 11, no. 3, pp. 707–718, Feb. 2017.
[11] S. Adhikari and F. Li, "Coordinated v-f and p-q control of solar photovoltaic generators with MPPT and battery storage in microgrids," *IEEE Trans. Smart Grid*, vol. 5, no. 3, pp. 1270–1281, May 2014.
[12] L. Ren, Y. Qin, B. Wang, P. Zhang, P. B. Luh, and R. Jin, "Enabling resilient microgrid through programmable network," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2826–2836, Nov. 2017.
[13] L. Ren *et al.*, "Enabling resilient distributed power sharing in networked microgrids through software defined networking," *Appl. Energy*, vol. 210, pp. 1251–1265, Jan. 2018.
[14] R. Guerzoni, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action, introductory white paper," in *Proc. SDN OpenFlow World Congr.*, vol. 1, 2012, pp. 5–7.
[15] S. Batra and G. Taneja, "Reliability modeling and optimization of the number of hot standby units in a system working with two operative units," *Compusoft*, vol. 8, no. 2, pp. 3059–3068, 2019.
[16] J. Hwang, K. K. Ramakrishnan, and T. Wood, "NetVM: High performance and flexible networking using virtualization on commodity platforms," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 1, pp. 34–47, Mar. 2015.
[17] C. Wang, *Analysis Simulations Theory Microgrids*. Beijing, China: Science Press, 2013.
[18] D. Pattabiraman, R. H. Lasseter., and T. M. Jahns, "Comparison of grid following and grid forming control for a high inverter penetration power system," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Aug. 2018, pp. 1–5.
[19] C. Macana, A. Abdou, H. Pota, J. Guerrero, and J. Vasquez, "Cyber physical energy systems modules for power sharing controllers in inverter based microgrids," *Inventions*, vol. 3, no. 3, p. 66, Sep. 2018.
[20] V. Venkataramanan, A. Srivastava, and A. Hahn, "Real-time co-simulation testbed for microgrid cyber-physical analysis," in *Proc. Workshop Modeling Simulation Cyber-Phys. Energy Syst. (MSCPES)*, Apr. 2016, pp. 1–6.
[21] X. Lu, X. Yu, J. Lai, Y. Wang, and J. M. Guerrero, "A novel distributed secondary coordination control approach for islanded microgrids," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2726–2740, Jul. 2018.
[22] W. Gu, G. Lou, W. Tan, and X. Yuan, "A nonlinear state estimator-based decentralized secondary voltage control scheme for autonomous microgrids," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4794–4804, Nov. 2017.
[23] J. Schiffer, T. Seel, J. Raisch, and T. Sezi, "Voltage stability and reactive power sharing in inverter-based microgrids with consensus-based distributed voltage control," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 1, pp. 96–109, Jan. 2016.
[24] M. B. Delghavi and A. Yazdani, "An adaptive feedforward compensation for stability enhancement in droop-controlled inverter-based microgrids," *IEEE Trans. Power Del.*, vol. 26, no. 3, pp. 1764–1773, Jul. 2011.
[25] A. Yazdani and R. Iravani, *Voltage-Sourced Converters in Power Systems*, vol. 34. Hoboken, NJ, USA: Wiley, 2010.
[26] H. Ye, Y. Liu, and P. Zhang, "Efficient eigen-analysis for large delayed cyber-physical power system using explicit infinitesimal generator discretization," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2361–2370, May 2016.
[27] M. Abdolkarimzadeh, M. Nazari-Heris, M. Abapour, and M. Sabahi, "A bridge-type fault current limiter for energy management of AC/DC microgrids," *IEEE Trans. Power Electron.*, vol. 32, no. 12, pp. 9043–9050, Dec. 2017.
[28] A. A. Bani-Ahmed, "Design and implementation of a true decentralized autonomous control architecture for microgrids," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Wisconsin-Milwaukee, Milwaukee, WI, USA, 2017.
[29] O. Nzimako, "Real-time simulation of a microgrid system with distributed energy resources," M.S. thesis, Dept. Elect. Comput. Eng., Univ. Manitoba, Winnipeg, MB, Canada, 2015.

LIZHI WANG (Student Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 2015 and 2018, respectively. He is currently pursuing the Ph.D. degree in electrical engineering with Stony Brook University, Stony Brook, NY, USA. His current research interests include stability analysis and distributed control of microgrid, cyber physical security for electric power networks, and quantum computation and its application in power grid.



YANYUAN QIN (Student Member, IEEE) received the B.S. degree in automation from the Nanjing University of Aeronautics and Astronautics, China, in 2011, and the M.S. degree in control science and engineering from Shanghai Jiao Tong University, China, in 2014. He is currently pursuing the Ph.D. degree with the Computer Science and Engineering Department, University of Connecticut.

His research interests are in software-defined networking and wireless networks.



ZEFAN TANG (Student Member, IEEE) received the B.S. degree in mechanical engineering from Zhejiang University, Zhejiang, China, in 2014, and the M.S. degree in electrical and computer engineering from the University of Michigan–Shanghai Jiao Tong University Joint Institute, Shanghai Jiao Tong University, Shanghai, China, in 2017. He is currently pursuing the Ph.D. degree in electrical engineering with Stony Brook University, Stony Brook, NY, USA.

His current research interests include distributed renewable energy systems, cyber physical security for electric power networks, quantum security, microgrid, power system resilience, and cyberattack-resilient load forecasting.

PENG ZHANG (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 2009.

He was a Francis L. Castleman Associate Professor and a Centennial Associate Professor with the University of Connecticut, Storrs, CT, USA, from 2017 to 2019. He was a System Planning Engineer at BC Hydro and Power Authority, Vancouver, from 2006 to 2010. He is currently a SUNY Empire Innovation Professor with Stony Brook University, Stony Brook, NY, USA. His research interests include networked microgrids, programmable microgrids, cyber resilience, formal methods and reachability analysis, and quantum engineering.

Dr. Zhang is an individual member of CIGRÉ. He is an Editor of the IEEE TRANSACTIONS ON POWER SYSTEMS, the IEEE TRANSACTIONS ON SUSTAINABLE ENERGY, and the IEEE POWER AND ENERGY SOCIETY LETTERS. He is also an Associate Editor of the IEEE JOURNAL OF OCEANIC ENGINEERING and the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.