Development of a Controller Hardware-in-the-Loop Platform for Microgrid Distributed Control Applications

Yuhua Du, Hao Tu, Srdjan Lukic, David Lubkeman
FREEDM Systems Center
North Carolina State University
Raleigh, NC, USA
{ydu7, htu, smlukic, dllubkem} @ncsu.edu

Abhishek Dubey, Gabor Karsai
Institute for Software Integrated Systems
Vanderbilt University
Nashiville, TN, USA
{abhishek.dubey, gabor.karsai} @vanderbilt.edu

***Abstract*—Microgrids (MGs) are ideally suited for distributed control solutions. However, implementation and validation of the developed distributed control algorithms are quite challenging. In this paper we propose a Controller Hardware-in-the-Loop (CHIL) platform for MG distributed control applications that satisfy the requirements of *IEEE Std. 2030.7* for MG control systems. We describe two main features of the proposed platform: 1) a software platform that enables the implementation of control algorithms that have been developed analytically and 2) a real-time MG testbed that replicates practical MG operation environment by using real-time communication network and grid solutions. Implementation and validation of a distributed MG synchronization operation control strategy are used to demonstrate the performance of the proposed CHIL platform.**

***Index Terms*—controller hardware-in-the-loop, distributed control, hardware-in-the-loop, microgrid control system**

I. INTRODUCTION

The world has witnessed a dramatic growth of microgrids (MG) in the last decade. The development of enabling technology and the increasing market demand are leading MG to a more important role in the future electric power systems. Unlike conventional distribution systems that have large inertia due to large online synchronous generators, MGs are mainly supported by inverter-based distributed generation (DG) and have very small inertia. As the MG systems are getting more complex, it is crucial to develop MG control systems that are generic, reliable, and ensure proper MG operation. In order to standardize the requirements for MG control systems, *IEEE Std 2030.7-2017 IEEE Standard for the Specification of MG Controllers* defines "the functional requirements of the MG controller in a manner that can be universally adopted" [1]. Many proposed MG control strategies follow the specifications from *IEEE Std 2030.7*, where the overarching goal is to move from centralized to decentralized and distributed control approaches [2]–[4].

Despite significant efforts to develop distributed MG control algorithms, few were tested in the deployment environment, due to the inherent cost and risk associated with testing algorithms in actual MGs [5]. Hardware testbeds are typically small-scale prototypes, with limited components and simple system topology [6]. Additionally, fault scenarios are difficult to test in the field without risk of equipment loss, for example, distribution system reconfiguration under abnormal operating conditions [7]. As a result, software validations (e.g. MATLAB/Simulink) have been widely adopted. Compared to a hardware implementation, such an approach shows significant advantages due to its easy and economic access to most researchers. Moreover, most simulation software provides models that can be easily adopted and modified for any use case, which greatly reduces the development time. However, such approach has the following disadvantages: 1) the simulated system is usually too ideal to provide convincing results for safe system deployment; 2) most models are simplified to a level where they do not accurately represent all the features of the component, making edge conditions difficult to test; 3) practical communication networks rely on multiple protocols and time-varying latency, while data exchanges in a purely simulation environment are usually ideal; and 4) time synchronization is guaranteed in a simulation due to the predetermined simulation time step.

To overcome the above mentioned drawbacks, real-time hardware-in-the-loop (HIL) simulations provide a higher level of fidelity [8]. Commercial simulators are able to model the MG system in real-time and integrate actual equipment and communication links into the simulation. Controllers of critical components, whose operating modes may be proprietary, can be integrated into the HIL simulation, while the non-critical components are modeled in the simulator. Still, the control algorithms that need to be validated, are typically modeled in the real-time simulator, reducing the fidelity of the resulting simulation approach. Since the main goal of the research effort is to validate the MG controller operation, simulating the control algorithm in the real-time simulator does not allow for evaluating possible design issues including: 1) distributed controller designed with complex structure could require too much computing power and becomes economically unfeasible; 2) algorithm scalability may be difficult to evaluate if the number of participating agents in a distributed control algorithm is large and variable; and 3) information exchange among distributed

978-1-5386-7665-3/18/$31.00 © 2018 IEEE

hardware controllers requires complementary communication network which presents its own challenges and limitations.

In this paper, a Controller Hardware-in-the-loop (CHIL) platform for MG distributed control applications is proposed. The proposed platform provides a MG testbed to validate the performance of any MG controller that follows the requirements defined in *IEEE Std 2030.7*. In addition to the stated advantages of a conventional HIL simulation, the proposed CHIL platform enables the implementation and testing of distributed control strategies using real hardware controllers. We propose a software platform, called Resilient Information Architecture Platform for Smart Grid (RIAPS) to implement distributed control algorithms in hardware, using a scalable and modular approach [9], [10]. The proposed platform emulates the operating environment in which the MG distributed controllers operate. The rest of this paper is constructed as follows. In Section II, requirements for MG control system in *IEEE Std 2030.7* are briefly introduced. In Section III, the RIAPS platform is introduced to address the identified hardware implementation challenges. In Section IV, a comprehensive description of the CHIL platform setup is presented. In Section V, we briefly present the implementation of an application using the proposed platform. Finally, conclusions follow in Section VI.

II. IEEE STANDARD FOR THE SPECIFICATION OF MG CONTROLLERS

The main purpose of the *IEEE Std 2030.7* standard is to provide the minimum functional requirements for MG control to ensure a technically sound operation of the MG at the point of interconnection (POI). The standard is functionality-driven, and provides a modular approach, i.e. the defined minimum functions are applicable to MG operation regardless of system topology, configuration, or jurisdiction. The requirements and functions apply to a range of MGs and MG controllers.

There are in total six MG operation modes defined in the standard, namely: 1) steady state connected (**SS1**), 2) steady state islanded (**SS2**), 3) unplanned islanding (**T1**), 4) planned islanding (**T2**), 5) black start (**T3**) and 6) reconnect (**T4**). The interconnection requirements of the MG control system are satisfied using functions at the high level, core level and low level of its functional framework, as presented in Fig. 1. The functions designated as core functions in this standard are of highest importance for modular design of MG control systems and are under study in this work. Two core functions are defined in the standard:

- The dispatch function, which dispatches individual devices in given operating modes and with specified setpoints.
- The transition function, which supervises the transitions between connected and disconnected states, and ensures the dispatch is appropriate for the given state.

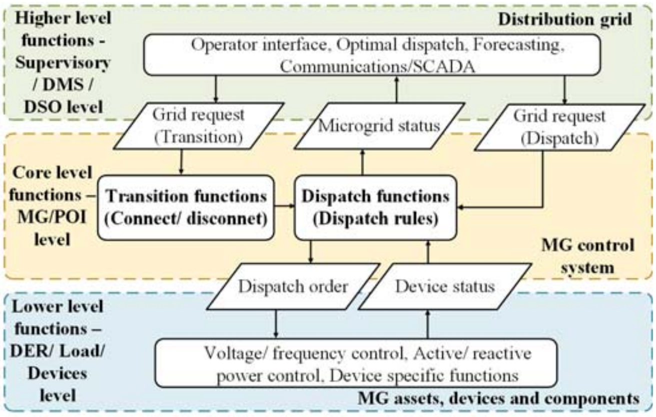

Fig. 1: MG control system functional framework (Source [1])

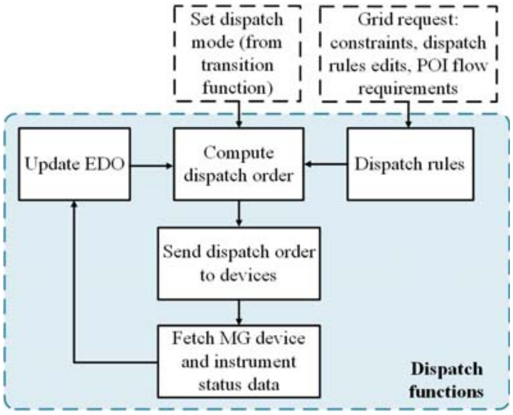

Fig. 2: Dispatch function high-level logic (Source [1])

The dispatch function determines the dispatching of MG assets in each mode and provides correct power setpoints, as shown in Fig. 2. The dispatch function provides the following functionalities in each operating mode:

- Balancing generation and load under normal islanded operating conditions;
- Re-dispatching controllable resources in response to internal events related to the load and generation profiles;
- Responding to external orders, for example interconnection agreement requirements, and external events by re-dispatching resources.

The emergency dispatch order (EDO) (see Fig. 2) is a continuously updated order that enables immediate non-critical load shedding to match available generation upon an unplanned islanding.

The transition function provides the logic to switch the dispatch function between the relevant dispatch modes. Four transition modes (**T1** to **T4**) and two steady state modes (**SS1** and **SS2**) exist. The transition logic for MG operation modes switch is shown in Fig. 3. As defined in the standard, the MG control system shall be able to carry out the operations for three transitions: **T1**, **T2** and **T4** whose operation steps are specified in the standard, while steps for **T3** are unique to each MG and are not specified.

III. RIAPS PLATFORM ARCHITECTURE OVERVIEW

The RIAPS platform [11] allows for efficient implementation of distributed control algorithms using a reusable development framework, which can be deployed on real hardware operating on the grid's edge. The RIAPS platform was first introduced in [12] as an open-source software platform; it provides a run-time and design-time software environment for building applications that execute on computing devices of the Smart Grid[1]. Applications include, but are not limited to, monitoring and control, data collection and processing, energy management, and safety applications. The key concept is to provide a "middleware" and various support service functions that enable each "actor" (i.e. an application process) to communicate with others so that the developers can focus on distributed application logic instead of messaging and networking. Compared to existing solutions, the RIAPS platform distinguishes itself with the following features:

- Dispersed fog-computing architecture with multi-tenant hosts;
- High-precision time synchronization and time-sensitive messaging;
- Coordination services and synchronized control actions across the network;
- Built-in resilience to faults anywhere in the system.

The RIAPS platform has a three-layer architecture, as shown in Fig. 4: **Component** is the reusable building block for applications and is used to provide specific physical functionality, like computation or measurement; **Components** are composed to form **actors** that realize an abstract function, like a control algorithm or state estimation. In RIAPS the distributed algorithms are implemented as **applications** and are composed of **actors**. Each **actor** encapsulates run-time layers of RIAPS that provide:

- Component framework that defines a concurrent model of computation for building distributed applications;
- Resource management framework for controlling the use of computational resources;
- Fault management framework for detecting and mitigating faults in all layers of the system;
- Security framework to protect the confidentiality, integrity, and availability of a system under cyber attacks;
- Fault tolerant time synchronization service;
- Coordination framework for coordinated computations and actions across the network.

Note that the **Application**'s business logic can be separated from the low-level details the framework.

A developed **application** is distributed to each computing node through RIAPS' deployment mechanism, as shown in Fig. 4. The distributed algorithms could be coded in Python or C++ by the developer using a single development machine and downloaded to all the distributed nodes running RIAPS. The RIAPS platform provides programming APIs to help development of device wrappers and provides a ZeroMQ-based messaging layer for information exchange between various RIAPS nodes. The RIAPS discovery service controls the information flow. Communication patterns available to applications include group-based publish-subscribe as well as point-to-point client-server mechanisms.

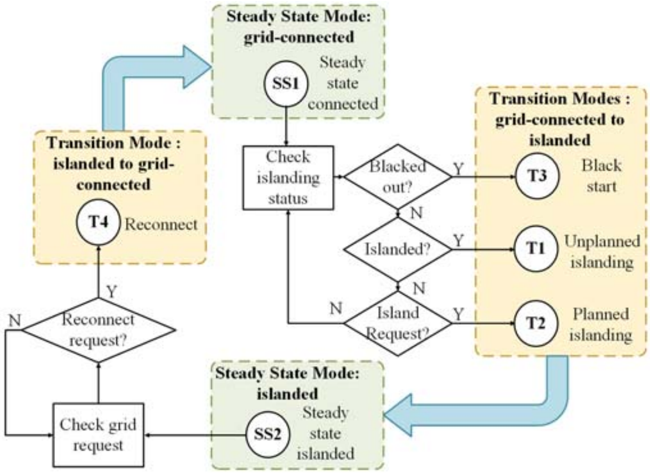

Fig. 3: Sequences associated with the transition function (Source [1])

IV. CONTROLLER HARDWARE-IN-THE-LOOP PLATFORM SETUP

To replicate the environment in which the MG distributed controllers operate, a real-time HIL MG testbed is proposed in Fig. 5. The testbed consists of three major parts, namely real-time simulators, hardware-in-the-loop and controller-in-the-loop:

- The real-time simulator models the MG response in real-time. Switching components (eg. power electronic converters) are modeled in FPGA-based simulator (Opal 5607) that runs with the simulation time step of a few hundred nanoseconds, while the non-switching components (eg. power transformer) are modeled in CPU-based simulator (Opal 5031), and execute with a time step of a few microseconds. The simulated MG system operates in both grid-connected (**SS1**) and islanded modes (**SS2**) and transition between the two modes.
- The hardware-in-the-loop part provides integration of hardware devices with the simulated MG system. They can be either industry-standard devices that provide realistic responses to disturbances, or customized devices that

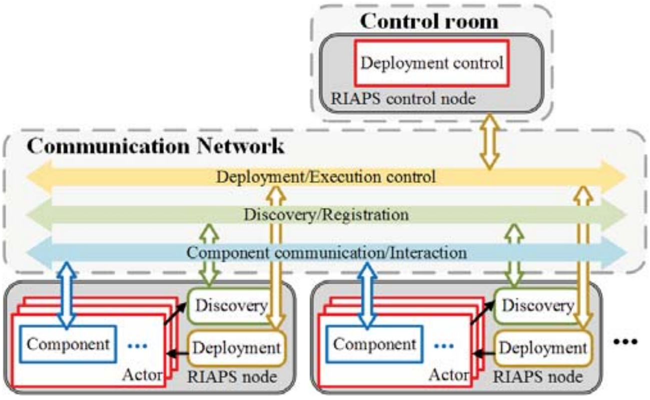

Fig. 4: RIAPS architecture overview

---

[1] Find more about RIAPS in *https://riaps.isis.vanderbilt.edu/* and get access to the open source files in *https://riaps.github.io/*

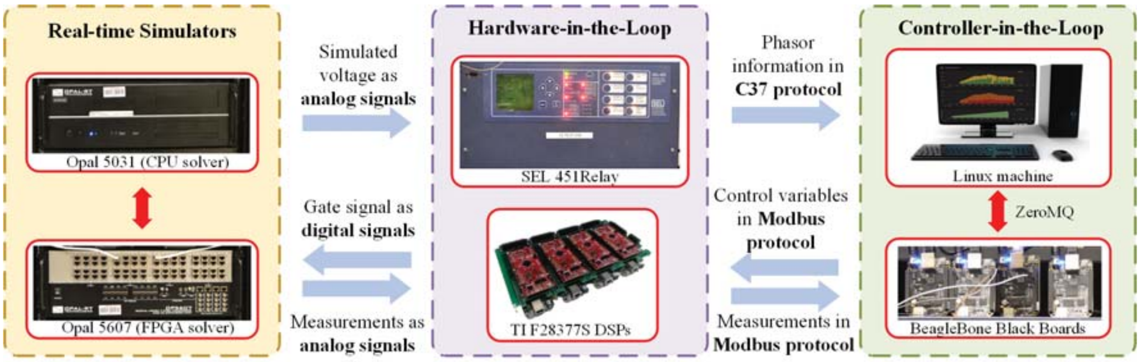

Fig. 5: Real-time HIL platform setup

await validation. Currently in our lab, MCUs from Texas Instruments (F28377S) are interfaced to the DG inverter switching models and function as the primary controllers. An SEL 451 Relay is integrated with the MG distribution network to emulate the behavior of a practical relay.

- The controller-in-the-loop enables the comprehensive validation of distributed algorithms. BeagleBone Black boards (BBBs) are selected as the hardware to implement local intelligence in the form of a control algorithm. Each BBB represents a distributed controller that can be attached to any controllable hardware device or the simulated MG device directly. In our lab, each MCU and relay is assigned a BBB. The execution and coordination of BBBs are realized by the RIAPS platform.

The proposed CHIL platform supports multiple communication protocols. The simulated voltage are read by the relay from the real-time simulator as analog signals. The relay reacts to the analog measurements and communicate with its assigned BBB using *IEEE C37.118.2* communication protocol. Each DG inverter modeled in the FPGA is directly controlled by the gate signals generated by its assigned MCU using PWM digital signals. Meanwhile, the MCUs are able to take the local measurements from the real-time simulator (e.g. inverter output current and voltage) as analog signals. Communication between the MCU and its assigned BBB uses Modbus communication protocol. The distributed control algorithm runs in the BBB and the communication among BBBs is enabled through the RIAPS platform using the messaging layer. The messaging architecture can be configured in the development environment, typically a Linux machine.

Although the delays introduced to the data exchange in each communication channel depend on the real-time network traffic, expected ranges of time delay are summarized in Table I: 1) The analog/digital signals are transferred using cables that are less than one meter long and thus the communication delay is ignored; 2) *IEEE C37.118.2* protocol delay is estimated referring to *IEEE Standard for Synchrophasor Data Transfer for Power Systems* [13]; 3) Communication delay introduced in the RIAPS platform is measured experimentally. The amount of time it takes for the data package transfers from the relay BBB to the phasor regulation DG BBB has been measured for 10K data packages; 4) Communication delay introduced by Modbus protocol is measured experimentally. The amount of time between the phasor regulation DG BBB sending one data package to the MCU and get the receiving confirmation back from the MCU has been measured for 10K data packages. This round-trip time is then halved to get the average Modbus protocol delay in Table I.

V. MG Distributed Control Applications

To demonstrate the salient features of the developed CHIL MG testbed, we introduce the implementation process of the distributed control strategy for islanded MG synchronization [14]. The proposed algorithm serves for the **SS4** (reconnect) mode as specified in *IEEE 2030.7 Std*. The algorithm is implemented in hardware using the RIAPS platform and validated using the CHIL MG testbed.

Due to the nature of MG synchronization problem, the voltage phasor mismatch on both sides of PCC needs to be measured and eliminated by all the DGs in a coordinated manner. However, such information is not locally accessible to all the DGs. In our proposed MG synchronization controller, a pinning-based consensus algorithm coordinates all the DGs: only the selected DG, named *phasor regulation DG* (PR-DG), directly receives the measured PCC voltage phasor mismatch while the non-PR-DGs operate using only its' neighbours' information. In an inductive MG, system frequency/phase regulation is decoupled with system voltage regulation. We assume that the main grid frequency is constant and at the rated value. The proposed frequency/phase regulation approach is defined as follows:

$$\omega_i = \omega^* - m_i P_i + \Omega_i \tag{1a}$$

$$k_i \frac{d\Omega_i}{dt} = -(\omega_i - \omega^*) - \sum_{j=1}^{N} a_{ij}(\Omega_i - \Omega_j) - r_i \Delta\delta_C \tag{1b}$$

where ω^* represents the main grid frequency; ω_i is the frequency measured by the $i-th$ DG ($i = 1, \cdots, N$); Ω_i

TABLE I: MG Testbed Communication Delay

Communication Form	*Time Delay*
Analog/Digital signals	$\Delta T_1 \approx 0$
C37.118.2 protocol	$\Delta T_2 = 10 \sim 100$ ms
The RIAPS platform	$\Delta T_3 = 1 \sim 50$ ms
Modbus protocol	$\Delta T_4 = 7 \sim 11$ ms

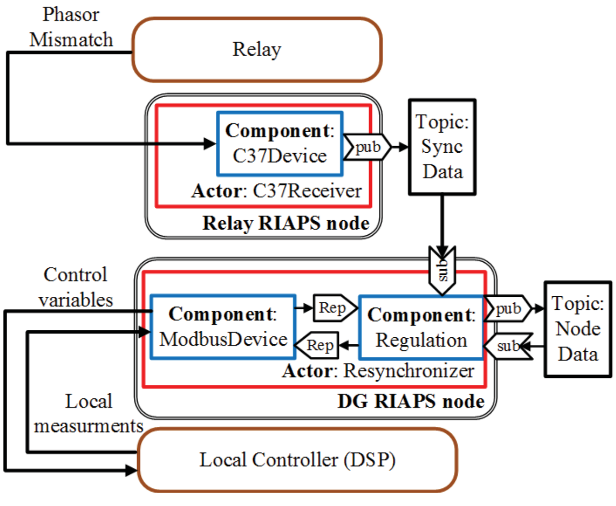

Fig. 6: Architecture of RIAPS **actor** realizing MG re-synchronization

is the frequency/phase regulation variable; k_i is the designed regulation gains; $a_{ij} = a$ if the $i-th$ and $j-th$ DG communicates, otherwise $a_{ij} = 0$; r_i is the designed phase regulation gain, $r_i > 0$ if the $i-th$ DG is the PR-DG, otherwise $r_i = 0$; $\Delta\delta_C$ represents the phase mismatch between the main grid and the islanded MG at PCC. The phase mismatch, $\Delta\delta_C$, as the external state of interest, is measured by the main relay and shared with the PR-DG.

The proposed voltage regulation approach is defined as follows:

$$E_i = E^* - n_i Q_i + e_i \tag{2a}$$

$$\kappa_i \frac{de_i}{dt} = -\sum_{j=1}^{N} a_{ij}\left(\frac{Q_i}{Q_i^*} - \frac{Q_j}{Q_j^*}\right) - \beta_i \Delta E_C \tag{2b}$$

where E^* represents the rated voltage; e_i is the voltage regulation variable; κ_i is the designed regulation gain; Q_i^* represents the rated reactive power of the $i-th$ DG; ΔE_C represents the voltage magnitude mismatch between the main grid and the islanded MG; and β_i is the designed magnitude regulation gain. If the $i-th$ DG is selected to be the PR-DG $\beta_i > 0$, otherwise $\beta_i = 0$.

The architecture for implementing the designed distributed controller in RIAPS is schematically presented in Fig. 6. The **application** has two actors. The first one is called **C37receiver** and it is deployed to the relay RIAPS node. **C37receiver** has one **component** called **C37device** which can communicate with the PCC relay using *IEEE C37.118.2* protocol. The measured phasor mismatch is packed under the topic **SyncData** and published across the RIAPS network. **SyncData** is only subscribed by the PR-DG RIAPS node. On each DG, one **actor** called **Synchronizer** is deployed to realize the proposed synchronization regulation. The **actor** has two **components**. One is called **ModbusDevice** and used to provide Modbus communication with DG's local controller (ie. the MCU). The other is called **Regulation** and used to realize the developed distributed algorithm, as shown in (1) and (2). The updated synchronization control variables (Ω_i and e_i) are calculated and then sent back to the DG's local controller through another Modbus message. At last, each DG RIAPS node will publish its updated variables while receive the ones from its neighbouring DGs.

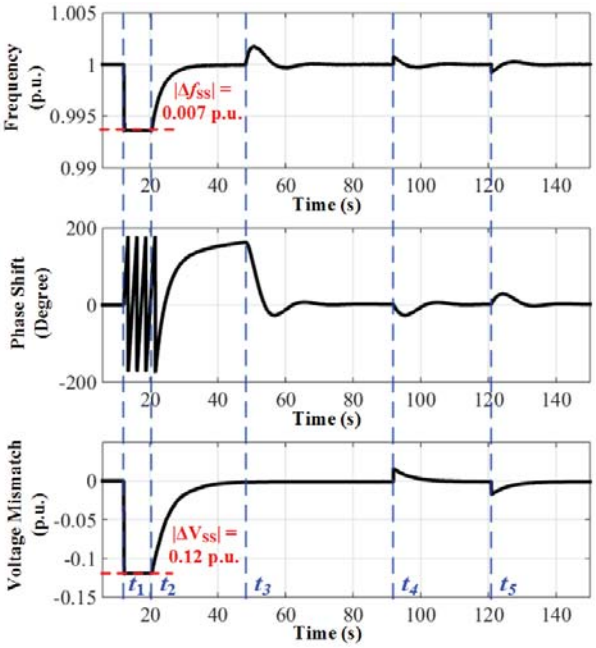

Fig. 7: Recorded voltage phasor mismatch at PCC

The developed distributed controller has been fully implemented in BBBs and validated using the proposed CHIL platform with practical operation and communication delay. Recorded system operation states are presented in Fig. 7. The MG is initially grid-connected and then islanded at $t = t_1$. The islanded MG is stabilized by droop control and results in steady state deviations of system frequency and voltage. The proposed frequency and voltage regulation is initiated at $t = t_2$ and the phase regulation is enabled at $t = t_3$. It can be observed that as the system converges, the islanded MG is fully synchronized with the main grid as the voltage phasor mismatch is eliminated. Load changes are introduced at $t = t_4$ and $t = t_5$ to demonstrate the controller's response to load/generation variations. The load change causes instant system frequency and voltage variations and results in new synchronization mismatch. We observe in Fig.7 that the synchronization controller quickly eliminates the phase and voltage mismatch and the system is quickly resynchronized with the main grid. Additionally, proportional power sharing among DGs are achieved, as shown in Fig. 8.

VI. CONCLUSIONS

In this paper, we propose a CHIL platform for MG distributed control applications that follows the requirements from *IEEE Std 2030.7*. For an analytically developed distributed control algorithm, the proposed platform is able to have it fully implemented and distributed on hardware controllers using the RIAPS platform. The controllers are integrated with developed HIL MG testbed which emulates

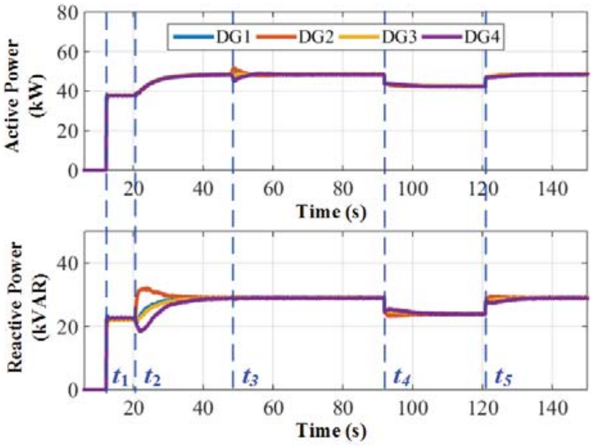

Fig. 8: Recorded power sharing among DGs

the practical MG operation environment. Real devices are integrated with the real-time simulator to provide practical system response and the CHIL platform runs on a practical communication network which contains multiple communication channels/protocols. Operation mechanism of the proposed platform are introduced and a distributed MG synchronization application is implemented to demonstrate its salient features.

ACKNOWLEDGMENT

This work was funded in part by the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000666. The views and opinions of authors expressed herein do not necessarily state or reflect those of the US Government or any agency thereof.

REFERENCES

[1] "IEEE standard for the specification of microgrid controllers," *IEEE Std 2030.7-2017*, pp. 1–43, April 2018.

[2] Z. Cheng, J. Duan, and M. Chow, "To centralize or to distribute: That is the question: A comparison of advanced microgrid management systems," *IEEE Industrial Electronics Magazine*, vol. 12, no. 1, pp. 6–24, March 2018.

[3] W. Gu, G. Lou, W. Tan, and X. Yuan, "A nonlinear state estimator-based decentralized secondary voltage control scheme for autonomous microgrids," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4794–4804, 2017.

[4] X. Lu, J. M. Guerrero, K. Sun, and J. C. Vasquez, "An improved droop control method for dc microgrids based on low bandwidth communication with dc bus voltage restoration and enhanced current sharing accuracy," *IEEE Transactions on Power Electronics*, vol. 29, no. 4, pp. 1800–1812, April 2014.

[5] P. Piagi and R. H. Lasseter, "Industrial application of microgrids," *Power System Engineering Research Center, University of Wisconsin-Madison*, 2001.

[6] G. Lou, W. Gu, J. Wang, J. Wang, and B. Gu, "A unified control scheme based on a disturbance observer for seamless transition operation of inverter-interfaced distributed generation," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5444–5454, Sept 2018.

[7] Y.-J. Kim, J. Wang, and X. Lu, "A framework for load service restoration using dynamic change in boundaries of advanced microgrids with synchronous-machine dgs," *IEEE Transactions on Smart Grid*, 2016.

[8] M. Burr, J. Camilleri, D. Lubkeman, Q. Long, and Y. Du, "Microgrid optimized resource dispatch for public-purpose resiliency and sustainability," Microgrid Inst., Little Falls, MN (United States), Tech. Rep., 2017.

[9] Y. Du, H. Tu, S. Lukic, D. Lubkeman, A. Dubey, and G. Karsai, "Implementation of a distributed microgrid controller on the resilient information architecture platform for smart systems (riaps)," in *Power Symposium (NAPS), 2017 North American*. IEEE, 2017, pp. 1–6.

[10] H. Tu, Y. Du, H. Yu, S. Lukic, P. Volgyesi, M. Metelko, A. Dubey, and G. Karsai, "An adaptive interleaving algorithm for multi-converter systems," in *2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*. IEEE, 2018, pp. 1–7.

[11] A. Dubey, G. Karsai, P. Volgyesi, M. Metelko, I. Madari, H. Tu, Y. Du, and S. Lukic, "Device access abstractions for resilient information architecture platform for smart grid," Vanderbilt Univ., Nashville, TN (United States), Tech. Rep., 2018.

[12] S. Eisele, I. Madari, A. Dubey, and G. Karsai, "Riaps: resilient information architecture platform for decentralized smart systems," in *20th IEEE International Symposium On Real-Time Computing, IEEE. Toronto, Canada: IEEE*, vol. 5, 2017, p. 2017.

[13] "IEEE standard for synchrophasor data transfer for power systems," *IEEE Std C37.118.2-2011 (Revision of IEEE Std C37.118-2005)*, pp. 1–53, 2011.

[14] Y. Du, H. Tu, and S. Lukic, "Distributed control strategy to achieve synchronized operation of an islanded mg," *IEEE Transactions on Smart Grid*, pp. 1–1, 2018.