Implementation of a Distributed Microgrid Controller on the Resilient Information Architecture Platform for Smart Systems (RIAPS)

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

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

***Abstract*— Formation of microgrids have been proposed as a solution to improve grid reliability, and enable smoother integration of renewables into the grid. Microgrids are sections of the grid that can operate in isolation from the main power system. Maintaining power balance within an islanded microgrid is a challenging task, due to the low system inertia, which requires complex control to maintain stable and optimized operation. Many studies have demonstrated feasible distributed microgrid controllers that can maintain microgrid stability in grid connected and islanded modes. However, there is little emphasis on how to implement these distributed algorithms on a computational platform that allows for fast and seamless deployment. This paper introduces a decentralized software platform called Resilient Information Architecture Platform for Smart Systems (RIAPS) that runs on processors embedded with the microgrid component. As an example, we describe the implementation of a distributed microgrid secondary control and resynchronization algorithms on RIAPS platform. The controller developed on RIAPS platform is validated on a real-time microgrid testbed.**

***Index Terms*—Microgrid, RIAPS platform, Distributed algorithm, Hierarchical control, Resynchronization**

I. INTRODUCTION

Microgrid controllers enable microgrid a smooth transition between grid-connected and islanded modes [1, 2]. Conventionally, hierarchical control ensures appropriate system operation and successful disconnection from main grid [3]. A synchronization algorithm limits the inrush current at point of common coupling (PCC), when the islanded microgrid reconnects to the main grid [4-6].

Currently, most commercial solutions for microgrid control use centralized approaches [7], where a microgrid centralized controller (MCC) uses a master-slave control approach. All the controllable resources have direct communication links to the centralized controller, which presents a single point of failure and possibly a slow system response if the communication channels are overtasked. To overcome the well-documented limitations brought by MCC, microgrid control approaches are transitioning from centralized to decentralized [8, 9]. Distributed control approach equips all the controllable resources with decision-making and plug-and-play capability, thus eliminating risk of single point failure.

Despite the vast literature on distributed microgrid control that handles specific issues in microgrid operation, little effort has been made to develop a platform to provide key capabilities to build and realize microgrid control applications. The platform should be able to handle microgrid operations with interacting software programs, deployed on devices across the network and solve problems collaboratively. Another requirement of this platform is that it should be capable of deploying and running applications that may use different communication and computational algorithms, and may be deployed on different components within the microgrid.

This paper gives an overview of the Resilient Information Architecture Platform for the Smart Grid (RIAPS) designed to address the needs of deploying distributed microgrid applications. We present an implementation of a couple of basic functions of a distributed microgrid controller, namely voltage and frequency regulation in islanded mode, and microgrid synchronization. We describe the algorithms, and present implementation details on the RIAPS platform.

II. RIAPS SYSTEM ARCHITECTURE OVERVIEW

RIAPS is a platform created for distributed computing. It facilitates distributed application implementation across a multitude of computing nodes. The advantages of the distributed computing in the context of the microgrid application include: (1) improved cyber and physical reliability by removing single point of failures; (2) faster

decision making by avoiding network penalties due to round-trip to the cloud; (3) improved scalability; and (4) better integration with hierarchical control systems. In RIAPS, each node has access to local measurements and actuators. The information exchange between nodes is handled as an inherent part as RIAPS platform.

The uniqueness of the RIAPS platform is that it provides a number advanced, distributed application services, without requiring the application developer to handle the low-level networking implementation details. Specifically the platform delivers a number of services that are orthogonal to the domain-specific application logic including: (1) time synchronization, (2) messaging middleware, (3) consensus and coordination mechanisms, (4) discovery and deployment mechanisms, (5) fault-detection and recovery mechanisms, and (6) distributed security mechanisms. Additional implementation details about RIAPS platform are introduced in [10].

Distributed control algorithms, called applications in RIAPS parlance, are deployed on the platform from a single control node, allowing for simple application deployment. Each application consists of components that run concurrently on one node and in parallel on many nodes. RIAPS platform distributes the application automatically to the nodes through a distributed middleware framework implemented using ZeroMQ (see Fig. 1). Beaglebone black (BBB) ARM board is selected as the hardware for the RIAPS nodes because of its low cost, high computational power and open-source nature.

Components on different nodes can exchange information in various ways defined by the RIAPS platform such as request-reply and publish-subscribe, and others. The underlying communication utilizes the ZeroMQ middleware. Communication interfacing RIAPS nodes with other components and controllers is implemented as a library. A few examples, used in the application described in this work, include Modbus and IEEE C37.118.2 synchrophasor data transfer protocol.

For distributed microgrid control, several challenges have to be overcome. First, the devices may be distributed over a wide geological range. The software program deployment and communication link should cover the same range. Second, the control bandwidth should be high because the inertia of a microgrid is much smaller compared with the conventional power system. Third, synchronization between devices should be accurate enough as unsynchronized data

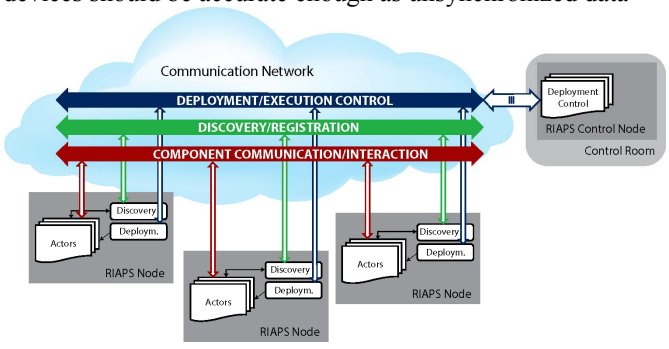


Figure 1. RIAPS Structure

may cause the system to be unstable. In this paper, we demonstrate how the RIAPS platform addresses these challenges.

III. MICROGRID CONTROL IMPLEMENTATION USING RIAPS

A. *Microgrid Structure*

Microgrid consists of several DGs and distributed loads. Fig. 2 shows a simplified microgrid structure used in this paper.

When islanded, DGs will be used to balance power generation and consumption. Inverter-based DGs in microgrid can be modeled as Voltage Source Converter (VSC). There are two types of control strategies applied to VSCs: 1) current control mode (CCM) and 2) voltage control mode (VCM). VSC-CCM units are usually used to interface renewable sources that are intermittent and uncontrollable like solar or wind. VSC-CCM units will be synchronized with microgrid by local phase-lock loop (PLL) and inject generated power. Because of the nature of such sources, there are limited controllability upon VSC-CCM units. Unlike VSC-CCM unit, VSC-VCM units are used to interface resources that are sustained and controllable like battery and generator. VCM-VSC units are used to provide power balance and voltage support when microgrid is islanded. Droop control is usually applied in all the VCM-VSC units to provide power balance.

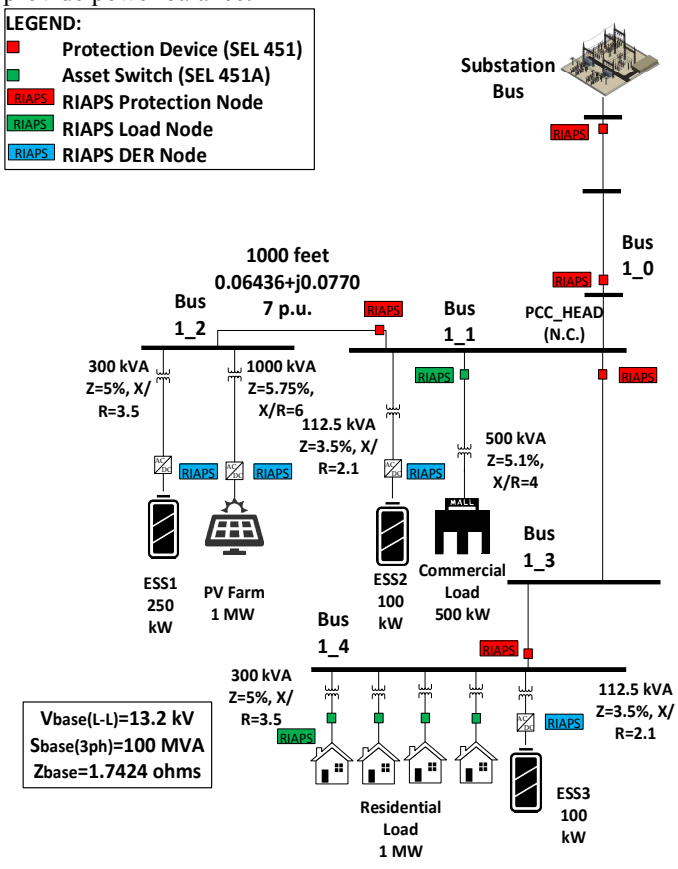


Figure 2. Simplifed Microgrid Testbed Topology

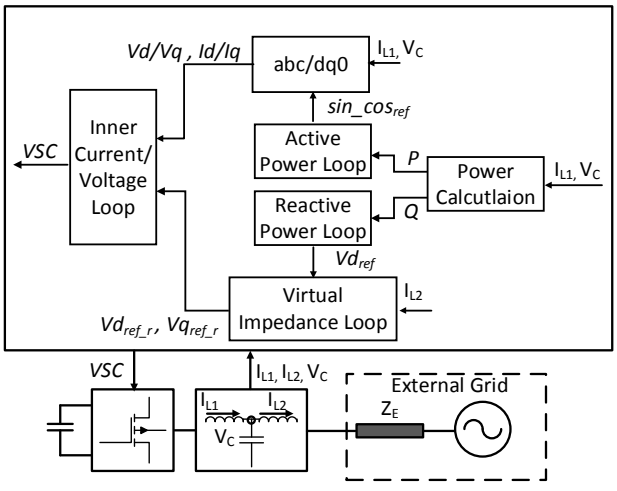


Figure 3. Droop Controlled VCM-VSC Structure

A general control structure for VCM-VSC is shown in Fig. 3.

B. Microgrid Secondary Control and Implementation

Due to the nature of primary droop control, any power imbalance within the microgrid before islanding will lead to frequency and voltage deviation when microgrid is disconnected from main grid. Secondary control is introduced to remove such deviations [3]. Conventionally, secondary control is done using Automatic Generation Control (AGC), which is a slow and centralized control approach. A lot of works have been done on achieving secondary control in distributed ways for islanded microgrid [11-13]. Unfortunately, few have been able to implement their developed algorithms under hardware level.

In this paper, the distributed-averaging proportional-integral (DAPI) controller [13][14] is adopted. It is developed using RIAPS platform and implemented on all the VSC-VCM units. The DAPI frequency controller is shown below:

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

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

where Ω_i is the frequency secondary control variable, m_i presents active power- frequency droop gain and ω^* presents rated frequency. Equation (1a) is a standard droop controller with additional secondary control regulation Ω_i. In (1b), k_i is a positive controller gain, and $A = \{a_{ij}\}$ presents the system communication topology: $a_{ij} = 1$ indicates a valid communication link between node *i* and node *j* and $a_{ij} = 0$ indicates no information exchange between node *i* and node *j*.

The DAPI voltage controller is shown below:

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

$$\kappa_i \frac{de_i}{dt} = -\beta_i (E_i - E^*) - \sum_{j=1}^{n} b_{ij}\left(\frac{Q_i}{Q_i^*} - \frac{Q_j}{Q_j^*}\right) \tag{2b}$$

where e_i is the voltage secondary control variable, n_i presents reactive power- voltage droop gain and E^* presents rated voltage. Equation (2a) is a standard droop controller with the additional secondary control regulation e_i. κ_i and β_i are positive controller gains and $B = \{b_{ij}\}$ presents system communication topology. Q_i^* is the reactive power sharing factor of VSC-VCM unit on node i and usually set to be the rated reactive power by default.

Because of the nature of digital controller, DAPI controllers presented in (1b) and (2b) are first discretized:

$$\Omega_i(k+1) = \Omega_i(k) + \frac{1}{\kappa_i}[-(\omega_i(k) - \omega^*) - \sum_{j=1}^{n} a_{ij}(\Omega_i(k) - \Omega_j(k))] \times t \tag{3a}$$

$$e_i(k+1) = e_i(k) + \frac{1}{\kappa_i}[-\beta_i(E_i(t) - E^*) - \sum_{j=1}^{n} b_{ij}\left(\frac{Q_i(k)}{Q_i^*} - \frac{Q_j(k)}{Q_j^*}\right)] \times t \tag{3b}$$

where t is the designed update iterative step.

Based on (3), local measurement ω and E are needed in the secondary control algorithm. The local measurements are sent from each DG to its associated RIAPS node. Once this measurement is available, a RIAPS component ***Sensor*** will time stamp it and forward to a second RIAPS component ***Averager***.

In ***Averager*** this measurement will be packed as ***Node Data*** message together with timestamp and the control command Ω_{k+1} from the last time step. ***Node Data*** will be published across the RIAPS network via a publish port. In ***Averager*** a subscribe port subscribes ***Node Data***. Therefore, component ***Averager*** can received the ***Node Data*** messages from other nodes while publish its own. This is conveniently realized by RIAPS platform since the logic on all nodes is the same. Since the TCP/IP backbone is used in the system demonstration, it is safe to assume that communication link exits between any two nodes.

After ***Averager*** receives the ***Node Data*** from all nodes, it uses the equation (3) to calculate the new control commands Ω_k. This algorithm is called by a clock port every 20 ms. As a standard port in RIAPS, the time step of a clock port can be easily configured to best suit the algorithm.

The new control commands Ω_k then are sent to the inverter to modify the droop curve. They are packed into the ***Node Data*** as they are needed for the next time step. Fig. 4 shows the DAPI controller implementation in RIAPS.

C. Microgrid Resynchronization and Implementation

Secondary control can synchronize microgrid frequency and voltage with main grid, but to have islanded microgrid

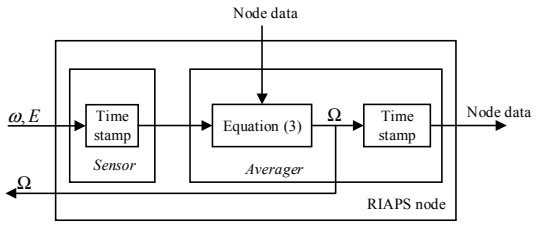


Figure 4. DAPI controller implementation in RIAPS

reconnected to main grid, synchronization of voltage phase on both sides of PCC are also needed. Microgrid resynchronization is usually done by a single VSC-VCM unit to minimize the natural conflict between phase regulation and frequency regulation [4]. The resynchronization VSC-VCM unit can be static or selected dynamically. Different approaches have been proposed for phase regulation [14-16]. Assume that before resynchronization, the frequency and voltage of islanded microgrid have already been regulated by secondary control, the resynchronization controller is developed as:

$$\omega = \omega^* - m_i P_i + R_i \tag{4a}$$

$$R_i = (k_P + \frac{k_I}{s})(\theta_G - \theta_I) \tag{4b}$$

where R_i is the resynchronization correction signal, k_P and k_I are the designed PI controller gains, θ_G and θ_I are the voltage phase measured at main grid side and islanded microgrid side of PCC, respectively. Phase difference $\theta_G - \theta_I$ will be measured by PCC relay and sent to the designated resynchronization VSC-VCM unit. Once the voltage phasor difference falls in the predefined criteria, PCC relay will automatically reclose. Resynchronization controller presented in (4) is implemented locally in each available VSC-VCM unit. Work flow of the proposed resynchronization procedure is shown in Fig. 5.

The resynchronization unit assignment and communication between PCC relay and resynchronization unit are implemented on RIAPS platform.

At PCC, the grid side voltage phasor and the microgrid side phasor are measured by the PCC relay. The relay has a RIAPS node associated with it. IEEE C37.118.2 synchro phasor data transfer protocol is implemented between the relay and its RIAPS node. A ***C37 Receiver*** component in the RIAPS node receives the phasor measurement from the relay. Then, the measured phasor data will be converted to a standard RIAPS message ***Phasor Data*** and it is published across RIAPS network. Any component that subscribes ***Phasor Data*** can receive this message. In our implementation, this message is subscribed by two nodes. One is the resynchronization DG. Upon receiving ***Phasor Data***, the resynchronization DG will use it for resynchronization as in equation (4). The second node subscribing ***Phasor Data*** is a general RIAPS data logger. It receives and display the data in real time. It also stores the data in database for future use.

As only one DG will act as the resynchronization DG, a distributed assignation algorithm is implemented to select the resynchronization DG.

If one DG can serve as the resynchronization DG, it will send a resynchronization availability signal to its RIAPS node. Upon receiving this signal, a component ***Resynchronizer*** will pack it as ***Resynchronization Data*** message with timestamp. Then, a publish port of ***Resynchronizer*** publishes this message. At the same time ***Resynchronization Data*** is subscribed by a subscribe port of ***Resynchronizer***. This way ***Resynchronizer*** on one node can receive the resynchronization availability status of any nodes. A priority list is implemented to determine the resynchronization DG when more than one DG are available. The priority list is based criteria such as the power rating of the DGs and their distance to PCC. According to the priority list, the DG with the highest priority among the available DGs will be selected as the resynchronization DG. Fig. 6 shows the implementation of resynchronization DG assignation in RIAPS.

Once the resynchronization DG is selected, the result will publish across RIAPS network under message ***Resynchronization Selection***. The data logger subscribes ***Resynchronization Selection*** to record the resynchronization process.

IV. RESULTS

To further prove the outstanding features of RIAPS platform, microgrid secondary control and resynchronization procedure presented in previous section are implemented on hardware testbed. A real-time microgrid testbed is developed in Opal-RT. A SEL-451 relay is implemented with Opal-RT to provide Hardware-in-the-Loop (HIL) simulation. RIAPS nodes are associated with important devices such as relay, DGs and loads. Fig. 7 shows the photograph of hardware implementation.

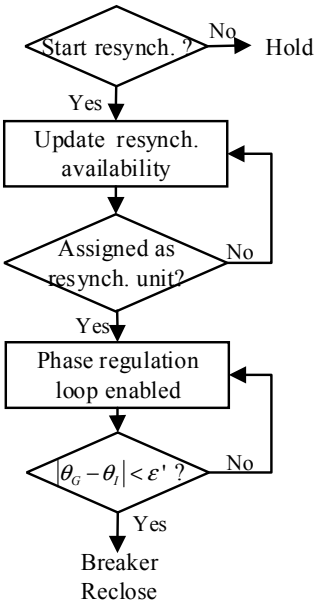


Figure 5. Resynchronization Workflow

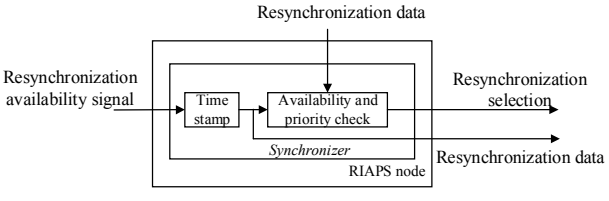


Figure 6. Resynchronization DG assignation implementation in RIAPS

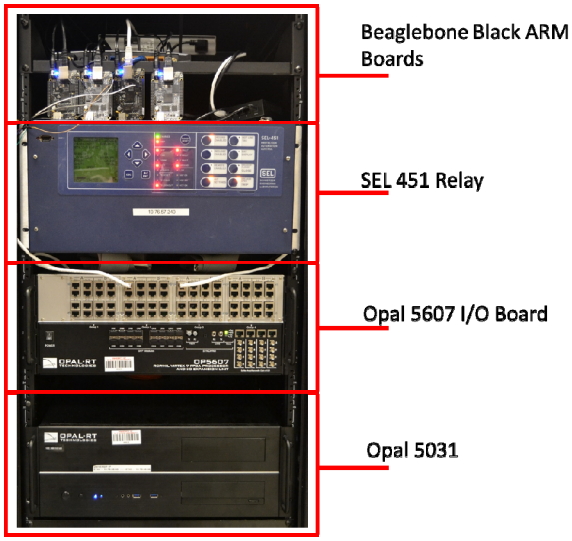


Figure 7. Photograph of Hardware Implementation

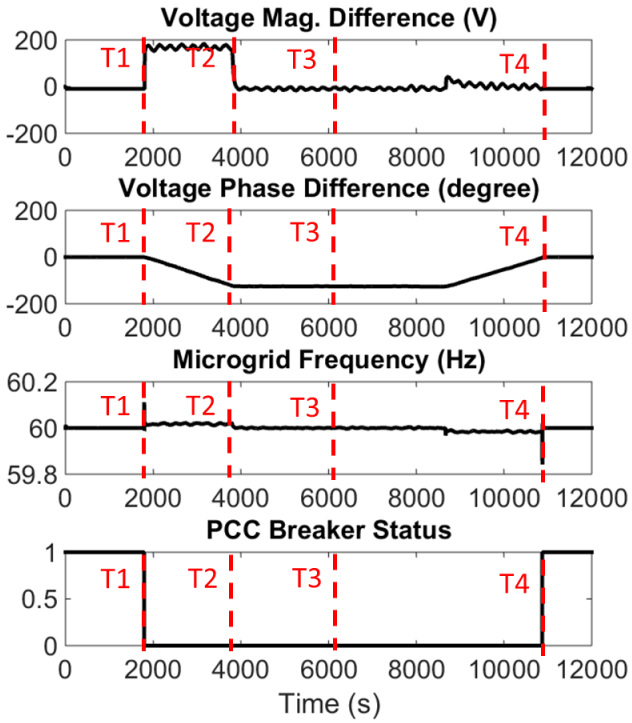
A. *Microgrid Testbed Specification*

The microgrid testbed structure is presented in Fig. 2. Three different types of RIAPS node are presented: RIAPS protection node, RIAPS load node and RIAPS Distributed Energy Resource (DER) node. In this case study, only protection node in PCC relay and DER node in ESS will be involved. Each ESS is associated with one BBB. The relay also has a dedicated BBB. An additional BBB is introduced for data logging. The BBBs, relay and Opal-RT simulator are connected to the same network.

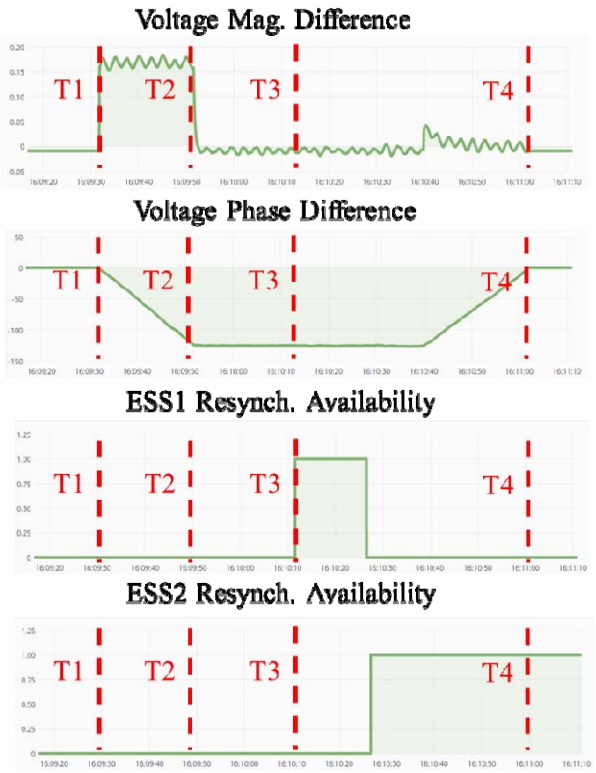
The Energy Storage System (ESS) are modeled as VSC-VCM unit and PV farm is modeled as VSC-CCM. Dynamic load model proposed by Lincoln Lab is adopted [17]. Relay HIL simulation is achieved in the following way: PCC relay is modeled as breaker in Opal-RT and controlled by SEL relay. The simulated three phase voltages will be sent from Opal-RT to SEL relay in real-time. SEL relay reads the voltage measurement and operates accordingly. The status of SEL relay (close/trip) can be captured and sent to Opal-RT. Communication between Opal-RT and SEL relay is achieved by analog signal.

B. *Simulation Results*

This case study presents how to have the islanded microgrid reconnected to main grid through RIAPS platform. Data sets sampled in Opal-RT are: voltage magnitude and phase difference at both sides of PCC, microgrid frequency and PCC breaker status (1 means close and 0 means open). Data sets sampled in RIAPS data logger are: voltage magnitude and phase difference at both sides of PCC, resynchronization availability of both ESS1 and ESS2. The results are shown in Fig. 8:



(a) Data Sets Collected in Opal-RT



(b) Data Sets Collected in RIAPS Data Logger

Figure 8. Simulation Results

Microgrid is originally grid-connected. At T1, PCC breaker is manually opened and microgrid is islanded. After the transients, the deviation introduced by droop control on

both microgrid frequency and voltage can be observed. Absolute PCC voltage phase difference keeps increasing because of the frequency deviation.

At T2, secondary control is enabled for all the ESSs. Voltage and frequency steady state errors are then quickly eliminated. PCC voltage phase difference remains relatively constant at some non-zero value. Islanded microgrid at this state cannot be reconnected to main grid.

At T3, resynchronization procedure starts. At first ESS1 is available to serve as the resynchronization DG while ESS 2 not. Therefore, the resynchronization assignation algorithm selects ESS1 as the resynchronization DG ('1' means enabled and '0' means disabled). Then, ESS2 also sends resynchronization availability signal to its RIAPS node. The resynchronization assignation algorithm then selects ESS2 as the resynchronization DG because ESS2 has higher priority, which in this specific case, is because ESS2 is closer to PCC. As ESS2 is selected, the status of ESS1 changes back to zero while the status of ESS2 remains one. After a predefine assignation window during which DG's availability signal can change, the resynchronization assignation is finalized. ESS2 is selected as the resynchronization DG. It starts the resynchronization algorithm based on the measurements from the relay node. Absolute PCC voltage phase difference then starts to reduce.

At T4, the voltage phasor difference falls in the SEL relay pre-defined auto-reclose criteria. PCC breaker recloses and the microgrid is reconnected to the grid.

V. Conclusions

RIAPS platform provides easy access for users to fulfil their developed distributed algorithms under a hardware level. This paper presents an implementation of microgrid secondary control and resynchronization through RIAPS platform on a real-time microgrid testbed. The control algorithms are fulfilled by RIAPS platform and implemented in each distributed local controller. Real time HIL simulation results show that the local controllers are able to regulated the frequency and voltage of an islanded microgrid in a distributed manner. The resynchronization procedure implemented in RIAPS platform ensures a smooth reconnection to main grid.

VI. Future Plan

There are still great potentials of RIAPS platform that need to be uncovered. The authors are developing more microgrid control and manage solutions using RIAPS platform. Some but not all possible topics include: load shedding, volt-var control, fast transition scheme for intentional/unintentional islanding.

Acknowledgement

The information, data, or work presented herein 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] Lidula, N. W. A., & Rajapakse, A. D. (2011). Microgrids research: A review of experimental microgrids and test systems. Renewable and Sustainable Energy Reviews, 15(1), 186-202.

[2] Ustun, T. S., Ozansoy, C., & Zayegh, A. (2011). Recent developments in microgrids and example cases around the world—A review. Renewable and Sustainable Energy Reviews, 15(8), 4030-4041.

[3] Bidram, A., & Davoudi, A. (2012). Hierarchical structure of microgrids control system. IEEE Transactions on Smart Grid, 3(4), 1963-1976.

[4] Assis, T. M. L., & Taranto, G. N. (2012). Automatic reconnection from intentional islanding based on remote sensing of voltage and frequency signals. IEEE Transactions on Smart Grid, 3(4), 1877-1884.

[5] Vandoorn, T. L., Meersman, B., De Kooning, J. D., & Vandevelde, L. (2013). Transition from islanded to grid-connected mode of microgrids with voltage-based droop control. IEEE transactions on power systems, 28(3), 2545-2553.

[6] Lai, J., Zhou, H., Hu, W., Lu, X., & Zhong, L. (2015). Synchronization of hybrid microgrids with communication latency. Mathematical Problems in Engineering, 2015.

[7] Kaur, A., Kaushal, J., & Basak, P. (2016). A review on microgrid central controller. Renewable and Sustainable Energy Reviews, 55, 338-345.

[8] Yazdanian, M., & Mehrizi-Sani, A. (2014). Distributed control techniques in microgrids. IEEE Transactions on Smart Grid, 5(6), 2901-2909.

[9] Simpson-Porco, J. W., Shafiee, Q., Dörfler, F., Vasquez, J. C., Guerrero, J. M., & Bullo, F. (2015). Secondary frequency and voltage control of islanded microgrids via distributed averaging. IEEE Transactions on Industrial Electronics, 62(11), 7025-7038.

[10] Eisele, S., Madari, I., Dubey, A., & Karsai, G. (2017). RIAPS: Resilient Information Architecture Platform for Decentralized Smart Systems. In 20th IEEE INTERNATIONAL SYMPOSIUM ON REAL-TIME COMPUTING, IEEE. Toronto, Canada: IEEE (Vol. 5, p. 2017).

[11] Bidram, A., Davoudi, A., Lewis, F. L., & Qu, Z. (2013). Secondary control of microgrids based on distributed cooperative control of multi-agent systems. IET Generation, Transmission & Distribution, 7(8), 822-831.

[12] Liu, W., Gu, W., Sheng, W., Meng, X., Xue, S., & Chen, M. (2016). Pinning-based distributed cooperative control for autonomous microgrids under uncertain communication topologies. IEEE Transactions on Power Systems, 31(2), 1320-1329. Simpson-Porco, J. W., Shafiee, Q., Dörfler, F., Vasquez, J. C., Guerrero, J. M., & Bullo, F. (2015). Secondary frequency and voltage control of islanded microgrids via distributed averaging. IEEE Transactions on Industrial Electronics, 62(11), 7025-7038.

[13] Simpson-Porco, J. W., Dörfler, F., & Bullo, F. (2013). Synchronization and power sharing for droop-controlled inverters in islanded microgrids. Automatica, 49(9), 2603-2611.

[14] Tang, F., Guerrero, J. M., Vasquez, J. C., Wu, D., & Meng, L. (2015). Distributed active synchronization strategy for microgrid seamless reconnection to the grid under unbalance and harmonic distortion. IEEE Transactions on Smart Grid, 6(6), 2757-2769.

[15] Sun, Y., Zhong, C., Hou, X., Yang, J., Han, H., & Guerrero, J. M. (2017). Distributed cooperative synchronization strategy for multi-bus microgrids. International Journal of Electrical Power & Energy Systems, 86, 18-28.

[16] Lee, C. T., Jiang, R. P., & Cheng, P. T. (2013). A grid synchronization method for droop-controlled distributed energy resource converters. IEEE Transactions on Industry Applications, 49(2), 954-962.

[17] Salcedo, R. O., Nowocin, J. K., Smith, C. L., Rekha, R. P., Corbett, E. G., Limpaecher, E. R., & LaPenta, J. M. (2016). Development of a Real-Time Hardware-in-the-Loop Power Systems Simulation Platform to Evaluate Commercial Microgrid Controllers. Technical Report. Lexington, MA: MIT Lincoln Laboratory.