Multi-microgrids system’s resilience enhancement

Samira Chalah, Hadjira Belaidi
Signals and Systems Laboratory, Institute of Electrical and Electronic Engineering, University M’hamed Bougara of Boumerdes, Boumerdes, Algeria

ABSTRACT

Nowadays, electricity consumption is increasing rapidly which leads to conventional power systems exhaustion. Therefore, micro-grids (MGs) implantation can enhance the resilience of power systems by implication of new resources, such as renewable energy sources (solar panel and wind power systems), electric vehicles (EV), and energy storage systems (ESS). This paper proposes a new strategy for optimal power consumption inside one microgrid; then, the approach will be extended to optimize the power consumption to enhance the resilience in the case of multi-MGs systems. The system controller of each microgrid has been implemented using ESP32 microcontroller and Raspberry IP4. The proposed approach intends to enhance the resilience of the system to react to any contingency in the system such as loss of power linkage between MG and the network in case of any natural disaster, especially in the rural area. Two controllers are implemented; the first one ensures MG autonomy by the efficient use of its own sources. The second one handles the system resilience cases by demanding/delivering power from/into neighbor microgrids. Hence, this work enhances the system resilience with an optimal cost. Thus, the MG can offer ancillary services for the neighboring MGs.

1. INTRODUCTION

As digital technologies continue to advance, the concept of a smart grid has gained prominence. A smart grid builds upon the foundation of microgrids (MGs) by incorporating advanced communication, monitoring, and control systems. This intelligent system enables real-time monitoring, management, and integration of electricity flow to improve efficiency, reliability, and cost-effectiveness. Today's globe is more aware of renewable energy sources due to the scarcity of electricity, rising fuel costs, and negative environmental repercussions of conventional energy sources [1]. Especially, in remote or rural areas, where the power grid infrastructure is often limited, unreliable, and expensive to maintain, stand-alone MGs provide a reliable and sustainable energy solution [2], [3].

A MG can operate in two different modes: islanded, when it can run autonomously, and grid-connected in case it runs non-autonomously [4]. In the grid-connected mode, MGs are linked to the conventional grid; whereas, in the islanded mode, electricity from the main grid is disconnected, and the MG relies solely on its available distributed energy resources (DERs) to meet demand; if demand exceeds generation, it must be reduced in case of flexible loads [5]. An efficient and adaptable control system for managing energy flows between power sources, loads, and the utility grid is named the energy management
A renewable energy management system (REMS) is a gadget that monitors power in a microgrid. The goal of this system is to optimize power utilization generated by renewable energy, reducing power demand from the grid [21], [22]. However, due to the massive communication and computation requirements of linked power systems across a large region, it is challenging to implement a centralized REMS strategy. Though, the decentralized strategy necessitates a high level of coordination, which is made difficult by the availability of just local data when considering extended MGs [23]. Thus, a need for powerful MGs’ REMS. Accordingly, our aim throughout this work is to implement a REMS, based on Raspberry Pi4 board, able to optimize energy consumption in multiple systems and respond to the loads’ dynamic changes in real time.

Therefore, this work starts by proposing a control strategy to allow one microgrid to operate independently and/or collaboratively with the other microgrids based on the MAS concept [24]. The microgrid used in this paper joins many distributed power sources and home loads. Then, each microgrid is considered a separate agent which cooperates with the other agents in our MAS. The proposed system architectures and their control strategy are explained in the second section. Whereas the obtained results and the evaluation of the proposed strategy are presented in the third section. Therefore, the paper is concluded in the fourth section with the proposition of some perspectives.

2. METHOD

This section explains the proposed energy management strategy grounded in multi-agent systems for multi-microgrids’ collaboration. Through efficient coordination and resource sharing, microgrids can better handle power demand, ensure self-sufficiency, and enhance overall system resilience. This research showcases the potential for creating smarter, interconnected energy systems that are reliable and adaptable.

2.1. One microgrid system’s power management

In the case of one microgrid agent, the controller manages load demand and ensures the system autonomy following the diagram shown in Figure 1. The solar panel being the power source is used to supply the load and charge the battery when it is enabled, the battery is used to supply the load when the solar panel is disabled; thus, the controller has two modes: charging and discharging. It controls the power going into the battery; it also stops power from running back into the photovoltaic (PV) arrays and prevents the battery from completely discharging or from overcharging as well by efficiently controlling its state of charge (SoC). The capacity to charge and discharge the battery at various voltage levels is one benefit of utilizing a bidirectional DC-CD buck-boost converter with a battery. The buck-boost converter functions as a buck converter when the PV array has surplus energy available; thus, the power flow route is from the DC bus to the battery bank in order to charge the battery with requested current. Whereas the buck-boost converter uses the current control feature to function as a boost converter during overcast conditions or at night; hence, the power flows from the battery bank to the DC bus and discharges the battery with requested current.

The proposed controller has been modelled in MATLAB Simulink. The controller compares different values of output power for all the primary platforms and then decides which one will be the best. The flowchart given in Figure 2 demonstrates the different states that can be executed by the controller. To apply this protocol in Simulink a series of measurements and comparison operations are performed. These operations result in Boolean control signals that are used to control ideal switches; each switch connects a power source to the grid. Table 1 defines the different controller inputs and outputs.
Multi-microgrids system’s resilience enhancement (Samira Chalah)

Figure 1. One microgrid controller scheme

Figure 2. Flowchart describing the different states that can be executed by the microgrid’s controller

Table 1. Inputs and outputs of the controller

<table>
<thead>
<tr>
<th>Inputs/Outputs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_pv</td>
<td>It represents the instantaneous power delivered from the PV array.</td>
</tr>
<tr>
<td>SoC</td>
<td>It represents the instantaneous SoC of the battery, recorded using the system implemented in our laboratory [25].</td>
</tr>
<tr>
<td>SurplusRegister</td>
<td>It represents the availability of Surplus power (in case the MG has produced extra power SurplusRegister state will be 1).</td>
</tr>
<tr>
<td>ResilienceRegister</td>
<td>It represents the state where the MG is in need for help from its neighbor MGs (MG in resilience situation in case ResilienceRegister=1).</td>
</tr>
<tr>
<td>MG response</td>
<td>Presents the response message of the MG to its neighbors.</td>
</tr>
<tr>
<td>PV_ESU</td>
<td>Control signal between PV and energy storage unit (Battery).</td>
</tr>
<tr>
<td>PV_load</td>
<td>Control signal between PV and MG’s load.</td>
</tr>
<tr>
<td>PV_extra_load</td>
<td>Control signal between PV and extra load from neighbor MGs.</td>
</tr>
<tr>
<td>ESU_load</td>
<td>Control signal between battery and load.</td>
</tr>
</tbody>
</table>

Figure 3 describes the realized prototype to control the system. For each microgrid, a controller system equivalent to the one shown in Figure 3(a) has been implemented and connected to the corresponding MG’s load, PV, battery, and to the other MGs. The primary purpose of the ESP32 is to read the initial open-circuit voltage (OCV) of the battery. It then proceeds to interpolate the data to obtain the initial SoC. After that, the ESP32 enters an infinite loop where it continuously calculates the SoC using the coulomb counting method. The calculated SoC is then published to the Raspberry Pi for further processing, as depicted in the flowchart given in Figure 2. ESP32 microcontroller is responsible for recording the status of the load and the
different power supplies inside the MG; moreover, it triggers the corresponding relay to switch between the appropriate sources. Furthermore, the relays are triggered by Raspberry Pi 4 commands enabling seamless transitions between the various MGs according to the communicated states of Surplus and Resilience registers as will be described in the coming section as shown in Figure 3(b).

![Figure 3. Implemented microgrid controller system of (a) MG’s controller system and (b) overall communication diagram](image)

The communication starts with reading data from the sensor attached to an ESP32 microcontroller using the inter-integrated circuit (I2C) communication protocol. The sensor precisely measures the battery’s current and voltage, and the resulting data is interpolated to calculate the battery’s SoC. After acquiring the data, the ESP32 works as a message queuing telemetry transport (MQTT) client, publishing data wirelessly to the Raspberry Pi. The MQTT protocol [26], known for its lightweight and efficient nature, allows reliable connection between the ESP32 and the Raspberry Pi, and between the neighbor MGs [27]. The Raspberry Pi can also be used as a MQTT subscriber, subscribing to certain MQTT topics. When data is published to a subscribed topic, the MQTT broker accepts it and sends it to the Raspberry Pi subscriber. In addition to MQTT communication, two sensors, each with a unique address, are attached to the Raspberry Pi. These sensors read voltage and current data independently. The Raspberry Pi gathers all data and prepares it for transmission. An InfluxDB database is created to store and analyze the acquired data [28], [29]. The Raspberry Pi connects to WiFi and provides data to the InfluxDB database and to the other MGs. InfluxDB, which was created for the effective storing and structuring of time-stamped data, provides the seamless integration and retrieval of the acquired measurements. Grafana has been incorporated with the system to...
provide a strong visualization and monitoring tool after the data has been securely saved in InfluxDB. Customized dashboards have been constructed using the Grafana web interface to display data in useful and visually appealing visualizations. The needed data was retrieved from InfluxDB in real-time by setting panels and queries within Grafana. This dynamic data retrieval allows users to obtain up-to-date insights and a comprehensive picture of the data.

2.2. Multi_microgrid system

In MAS, each device functions as an agent performing a specific task and communicating with other devices (agents). The MAS controller serves as a management or coordinating agent, facilitating communication and energy exchange among the microgrids. The MG requests help from the other MGs when the battery is depleted of power, and the solar panel is not generating electricity; thus, the load has to draw power from other MGs’ resources. In the case of MAS, the controller manages the power going into an agent with resilience status; it responds to the other agents in resilience once it has a surplus of power. Our proposed MAS autonomy is ensured by the implementation of the architecture illustrated in Figure 4.

![Figure 4. Multi-microgrids’ controller scheme](image)

Initially, the two buffers indicating the presence of surplus/resilience of power are reset to zero (SurplusRegister=0 and ResilienceRegister=0). Once, a microgrid’s resources are not satisfying its load’s need, the ResilienceRegister is set to one. Therefore, the MG broadcasts a message expressing its need to the other MGs indicating (‘in need of PResilience=PLoad–PProduced’); with PResilience is the amount of the needed power. Algorithm 1 explains the process executed by the MG to indicate its state.

**Algorithm 1. Definition of the MG situation (resilience/surplus)**

SurplusRegister=0  
ResilienceRegister=0  
If PProduced>PLoad Then  
  SurplusRegister=1  
  ResilienceRegister=0  
ElseIf PProduced<PLoad Then  
  SurplusRegister=0  
  ResilienceRegister=1  
End if  

Broadcast msg: ‘in need of help for PResilience=PLoad–PProduced’
The other MGs, with SurplusRegister=1, hear the message of the MG in resilience and respond by a message indicating (‘I can help you with PSurplus=Pproduced–PLoad). Hence, the MG in resilience accepts the help of the first MG responding to it if the amount of the delivered power satisfies its need (PSurplus–Presilience) and stops the broadcast; otherwise, it keeps broadcasting a message indicating the amount of the missed power (Presilience=Psurplus–Psurplus). This approach is summarized in the given Algorithm 2 with N indicating the number of MGs.

Algorithm 2. MGs’ energy exchange
For i=1 to N
    If SurplusRegister i=1 Then
        PSurplus i=Pproduced i–PLoad i
        Respond to MGj (Microgrid in resilience) → ‘I can help you with PSurplus i’
        If MGj response = ‘Accept’ Then
            Transmit extra power to MGj
        Else
            Keep listening for other MGs broadcast
        End if
    Else if ResilienceRegister i=1 Then
        If PSurplus j>=Presilience i Then
            If MGi response = ‘Accept’ Then
                Get missed power from MGj
                Stop broadcast
            Else
                MGi response = ‘Accept’
                PResilience i=PResilience i–Psurplus j
                Keep broadcasting to get more power from other MGs
            End if
        End if
    End if
End for

3. RESULTS AND DISCUSSION
3.1. Single MG management
To evaluate the efficiency of the proposed approach, we started by testing a single MG behavior. Simulation results obtained from Matlab Simulink are given in Figure 5. Batteries are the first backup systems that store excess energy generated by PV. The first portion of Figure 5(a) depicts the battery’s state of charge (SoC). To safeguard the batteries from damage, they were charged from 0% to a maximum of 80%.

During the period \(0<t\leq6.5\), the PV system is exposed to sunlight, leading to an increase in irradiance. Consequently, the PV system generates power levels in the range of 400 \(W\leq P_{pv}<1,000\ W\), which is more than sufficient to meet the load’s demand. By selecting the PV_load control signal, the power generated by the PV system is directly supplied to the load. Additionally, since the battery is initially at SoC <80%, it is not fully charged. The excess power generated by the PV system is used to charge the battery by selecting the PV_ESU control signal as illustrated in Figure 5(b) third graph.

In this case, the bidirectional converter operates as a buck converter, reducing the DC bus voltage and allowing the flow of current into the battery for charging. The continuity of charging the battery yields it to reach the SoC max where the battery becomes fully charged (SoC=80%) when being supplied with power from the PV system (\(P_{pv}=500\ W\)) during \([6.5\leq t\leq12]\). At this point, the control system takes action to halt the battery charging by disabling the PV_ESU control signal as shown in Figure 5(b) third graph.

Consequently, the PV system exclusively provides power to the load, and any excess energy is transferred to supply the load of the neighbor MG in resilience situation as illustrated in Figure 5(b) second graph. This approach effectively prevents the battery from overcharging and potential damage and helps the other MGs in need. The last signal shown in Figure 5(b) ESU_load will be enabled once the load is supplied by the battery.

Once the battery is charged and the PV steel provides enough energy to power the load, a surplus state is generated (SurplusRegister=1), indicating that this MG can generate energy for neighboring MGs in a resilience state. An illustration of the MG’s load power is shown in Figure 5(c) which will be used for the evaluation of multi-MGs approach. It is clearly shown that the load is variable, and in some situations, it generates peaks that cannot be handled by the existing MG’s DERs, and it requires help from the neighbor MGs.
Figure 5. One MG control Matlab Simulink results: (a) one MG obtained simulation results, (b) one MG’s control signals, and (c) load power consumption scenario
3.2. Multi-MGs management

Three MGs are interconnected to illustrate the efficiency of the proposed approach. Moreover, three scenarios are tested to evaluate the possible MG’s situations as shown in Figures 6 and 7. Table 2 summarizes the SurplusRegister and ResilienceRegister of the three MGs corresponding to each scenario recorded from the Raspberry controllers. Scenario 1 has been captured in midday (between 12:00 to 16:00); when the PVs are satisfying the loads’ needs, and the batteries are already charged by the extra amount of PVs power. Therefore, ResilienceRegister are reset to zero; whereas surplus registers of the three MG’s are set to one indicating that there is an extra power that can be injected to MGs in need.

![Figure 6. The MGs’ consumed power amount](image)

![Figure 7. The MGs’ delivered power amount](image)

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Surplus register</th>
<th>Resilience register</th>
<th>Surplus register</th>
<th>Resilience register</th>
<th>Surplus register</th>
<th>Resilience register</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario 1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Scenario 2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Scenario 3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2. SurplusRegister and ResilienceRegister status
Scenario 2 was recorded at night when PVs were not able to produce any power. It illustrates the case where the load of MG1 is supplied via its own battery; therefore, both its registers are reset to zero (ResilienceRegister and SurplusRegister). On the other hand, MG2 is in a resilience situation indicated by setting ResilienceRegister=1. The load of MG3 is very low and can largely be supplied by its battery; at the same time, its SurplusRegister=1 indicates that it has extra power that can be injected into neighboring MGs. Hence, the load of MG2 will be supplied via MG3, which has enough power.

Scenario 3 was also recorded at night. In this scenario, both loads of MG2 and MG3 require load power of less than 300 W, which can be supplied by their batteries. At the same time, they can produce a surplus of power enough to supply the load of MG3 which is in a resilience situation.

3.3. The proposed control strategy benefits and comparison

Compared to the literature, demand-side management (DSM) lowers the carbon environmental impact of traditional generators while also providing a large economic benefit to all parties participating in and interacting with one another in the existing energy network [30]. Nevertheless, the idea of energy management has evolved from utility-driven regulation to one where end users are involved in setting prices and clearing the market. Consequently, the invented DSM’s methods appear out of date at this point. Another challenge that faces the newly developed technics is due to the operation of distributed generation (DGs) which is unpredictable and depends on the environment. This unpredictability makes it difficult to integrate DGs into the energy market and necessitates sufficient stability and flexibility in the system. Furthermore, the positioning and kinds of DG have a big influence on how the electrical market is planned and run. This problem is particularly noticeable in bigger distribution networks that have numerous connected microgrids [31]. As well as, among the difficulties that must be overcome to effectively employ microgrids and realize their potential are the fast switching between the different modes.

These challenges were addressed in our work. The successful deployment of the MAS multi-MG management system has been facilitated by the application of several procedures and methodologies. Over the course of the battery's operating cycles, a precise SoC estimation was attained. The data was transmitted seamlessly between the ESP and Raspberry Pi thanks to the wireless MQTT protocol, although a possible drawback could be the wireless network’s available bandwidth. The Raspberry Pi's switching time has shown to be incredibly quick and effective in controlling the system's power supply. In addition, the Raspberry Pi plays a key role in the collection and real-time transmission of important data to the user database. This facilitates the microgrid interaction with the other MGs, importing additional power as needed to maintain a reliable load supply. In summary, the proposed multi-MG system’s adoption has produced positive outcomes, proving its ability to meet loads’ power requirements in a variety of situations.

4. CONCLUSION

The management of MGs operating in island mode is very critical, especially at peak load demands. Hence, this paper proposed an approach to increase the system's resilience to respond to any incident in the system resources. The proposed controller ensured MG autonomy by efficient use of its own supplies. Moreover, the controller handles system resilience cases by requesting and providing power from/to neighboring microgrids by creating status registers indicating the existence of surplus or deficiency of energy in the system. The status of the two registers is broadcast between the MGs to enable them to communicate and respond to help each other.

The system's ability to maintain consistent and dependable energy to meet load consumption was shown by its implementation. This offered useful information for additional research and development. Additionally, the Raspberry Pi's switching capabilities have been shown to be incredibly quick and effective in controlling power flow throughout the system, guaranteeing a high level of supply continuity. Real-time data collection and transmission from/to the MG ensures that data is available for additional analysis and usage via the Grafana user interface. The system's successful operation has been considerably aided by the smooth integration of its various components.

After the implementation of the proposed architecture and its simulation, the system efficiency is enhanced, and the system resilience is handled effectively by the controllers. However, the system can be improved by considering multicriteria optimization such as considering the nearest MG first and the one with the lowest cost, to get an optimized cost function. These issues will be addressed in our future works.

ACKNOWLEDGMENTS

We would like to thank The Directorate-General for Scientific Research and Technological Development (DGRSDT) for its financial help.

Multi-microgrids system’s resilience enhancement (Samira Chalah)
REFERENCES


Multi-microgrids system’s resilience enhancement (Samira Chalah)

BIOGRAPHIES OF AUTHORS

**Samira Chalah** is a lecturer in the Power and Control Department, at the Institute of Electrical and Electronic Engineering, University M’hamed Bougara of Boumerdes, Algeria since 2014. She received the Magister degree in electrical engineering from the University Mouloud Mammeri of Tizi-Ouzou, Algeria, in 2012. Currently, she is a Ph.D. student in Electrical Engineering at the Institute of Electrical and Electronic Engineering, University M’hamed Bougara of Boumerdes, Algeria. She can be contacted at email: s.chalah@univ-boumerdes.dz.

**Hadjira Belaidi** received her Doctorate degree from Institute of Electrical and Electronic Engineering, University M’hamed Bougara of Boumerdes in 2015. She has been an assistant professor at the Institute of Electrical and Electronic Engineering since 2014. Currently, she is the leader of “Embedded Systems and Robotics” team in the Signals and Systems Laboratory, as well as a lecturer at Department of Electronic at Institute of Electrical and Electronic Engineering, University M’hamed Bougara of Boumerdes, Algeria. Her research interests include all-terrain mobile robots, environment modeling and manipulator robots, drones’ control, and smart grid. She can be contacted at email: ha.belaidi@univ-boumerdes.dz.
