We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

Multi-Agent Systems Based Advanced Energy Management of Smart Micro-grid

Leo Raju and Antony Amalraj Morais

Abstract

Microgrids play a major role in enabling the widespread adoption of renewable distributed energy resources. However, as the power generated from renewable resources is intermittent in nature, it impacts the dynamics and stability of the microgrid, and hence their integration needs new approaches to coordination and control. The existing systems lack run-time adaptive behavior. To face these constraints, the electric energy system must adapt by integrating Information and Communication Technologies (ICT). Multiagent system (MAS) is emerging as an integrated solution approach to distributed computing, communication, and data integration needs for smart grid application. Distributed and heterogeneous information can be efficiently processed locally, but utilized globally to coordinate distributed knowledge networks, resulting in reduction of information processing time and network bandwidth. Parallel operations, asynchronous communication, and autonomous actions of agents enable MAS to adapt to dynamic changes of the environment, thereby improving the reliability, responsiveness, fault tolerance, and stability of the microgrid. In this chapter, MAS is implemented with Java Agent DEvelopment (JADE) framework for advanced energy management of a microgrid. Also, MAS is linked with Arduino microcontroller for practical verification of agent operations. Three microgrids are interconnected to form a microgrid testbed, and smart grid features such as demand side management and plug and play are implemented, making it into a smart microgrid.

Keywords: multiagent system, microgrid, smart grid, Arduino, demand side management, plug and play

1. Introduction

The power sector is undergoing a profound change—depletion of fossil fuels and environmental considerations have made it to embrace renewable energy resources such as solar and wind. A microgrid is a building block of a smart grid and is poised to play a major role in enabling the widespread adoption of renewable distributed energy resources. However, as the power generated from renewable resources is intermittent in nature, it impacts the dynamics and stability of the microgrid, and hence their integration into the microgrid necessitates new approaches to coordination and control.

Power industry has vast experience in operating single, large, monolithic power systems, but little experience in the operation of the same grid with multiple renewable energy resources. However, there will be no single grid in the future, but only a large interconnection of virtual power plants, consisting of DERs, prosumers, and other energy resources. The high penetration of renewable energy resources requires new coordination and control approaches. The energy network has to be dynamically optimized for economic and environmental benefits. To meet these challenges, microgrid monitoring ought to incorporate communication and control in a distributive environment to achieve an optimal balance between generation, energy storage, and load demand.

Development in Information and Communication Technologies (ICT) has made a huge impact in the energy sector and the way energy is generated and distributed. We are moving toward more decentralized, more sustainable, and smarter power systems. The smart grid paradigm represents a transition toward intelligent two-way power delivery grids. Smart grid features such as self-healing, demand side management, plug and play, dynamic pricing, dynamic adaptation, and distributed optimization can be incorporated in the microgrid, turning it into a smart microgrid. A smart grid is a system of integrated smart microgrids.

2. Multiagent systems in microgrid

Agent-oriented programming is the latest paradigm of computer programming, used for complex and distributed systems. It has autonomic and proactive characteristics with higher level abstraction [1]. An agent is a computer program capable of flexible and autonomous action in a dynamic environment [2]. Multi agent system (MAS) has received a great deal of attention in the recent times due to its inherent ability to significantly improve the operational efficiency in a distributive environment [3]. A multiagent system (MAS) is a distributed system consisting of multiple software agents, forming a loosely coupled network and working together to solve problems that are beyond their individual capabilities [4]. MAS improves modularity, flexibility, adaptability, scalability, reconfigurability, and responsiveness with its inherent characteristics. MAS is run-time adaptive and distributes decisions to improve performance and stability.

Design and implementation of a MAS in a microgrid are discussed in [5]. MAS-based control in microgrid is explained in [6]. A real-time digital simulator used for real-time operation of MAS on microgrid is explained in [6]. MAS-based simulation of microgrid control is discussed in [7]. Ref. [8] gives a review of microgrids from MAS perspectives. Survey of MAS for microgrid control is given in [9]. MAS is promising heuristic techniques for solving problems whose domains are distributed, complex, and heterogeneous. Though MAS is widely recognized as the method of choice for realization of time-critical applications of smart grid, it has only been simulated for theoretical proof of concept [10]. There is a need for comprehensive real-time simulation considering all the options and practically verifying distributed energy management of microgrid using MAS, Arduino microcontroller through an experimental setup. MAS in JADE is linked with Arduino processor and the agent operations for environment dynamics are practically verified in [11, 12]. Verification and validation of MAS in microgrid have not been adequately addressed so far [13]. The recent development of control of microgrid using MAS is given in [14, 15]. Multiagent systems for grid outage management are discussed in [16]. Advanced energy management of microgrid using Arduino and multiagent system is discussed in detail in [17]. Here, only two microgrids are considered and transactive energy management is not adequately addressed.

To address the inadequacies, a smart microgrid testbed is implemented with interconnection of three microgrids, and the inherent features of MAS are leveraged for dynamic adaptation and distributed optimization to get the best of the service to the consumers, across the network, all the time, leading to economic and environmental optimization. Initially, two microgrids, one in the department and the other in the hostel building in the campus, each with solar unit, wind unit, load, battery, grid, and diesel unit, are considered. Then, the third microgrid in administration building is added to make a smart microgrid testbed. The characteristics of the smart grid are practically verified by linking MAS with Arduino microcontroller.

3. Environmental interaction in microgrid

The values from the environment such as the power required by the load, the power output of the solar PV, the state of charge of the battery, etc. are sensed and given as input to the JADE agent program through the Arduino microcontroller. Potentiometers are used to sense the environment values in Arduino. The communication with the Arduino microcontroller has to be in Java so that the sensed values can be directly given to the agents in the JADE environment.

3.1 Microcontrollers and communication methods

The microcontroller should have sufficient input pins for sensing the environment values. It should also be able to communicate with the agents using JADE framework. Various communication methods are discussed here.

3.1.1 Arduino as the microcontroller

There are numerous ways of sending the sensed values from the environment to the Java program using Arduino.

- Using Arduino plug-in in eclipse: Installation of the plug-in is simple and easy. However, the code is written in C++. This makes it difficult for the agents in the Java program to call the function accessing the microcontroller written in C++ language.

- Using J Arduino library: Using this library in the Java IDE, in our case Eclipse, we can access the microcontroller using a JAVA class itself. This library provides functions such as those in Arduino IDE `setup()`, `loop()`, etc. The main disadvantage of this method is the `loop()` function, without which the program is not allowed to run. When an agent calls this Java class containing the Arduino functions, to access the microcontroller, it goes into the `loop()` function and does not come out.

- Serial communication using RxTx library: This is the most suitable method for accessing Arduino from the Java code. A library called RxTx is used that facilitates serial communication with Arduino. Messages can be sent to and received from Arduino serially. However, since the communication is serial, only one agent can access the microcontroller at a time. In this case, the control agent is used for this communication.

The Arduino UNO board has limited number of analog input pins and hence Arduino MEGA is preferred to sense large number of environment variables in the microgrids.

3.2 Serial communication with Arduino

Potentiometers representing the environmental variables of the microgrid are connected to analog input pins of the Arduino board to input the environmental values. The potentiometers are range fixed with value of (0–1023) for all the environment variables. The environment variable values can be fixed such that the maximum value is fixed as 1024 and the variation is done accordingly. The agents receive the environmental values from the Arduino board through serial communication using RxTx library.

3.3 Actuators

The results of strategic operations of agents are reflected in the Arduino microcontroller, which can be given to the physical devices for actuation. This information is passed to the Arduino board through serial communication and given to its digital pins to which LEDs are connected. These LEDs indicate the action taken by the agents, and the status of the LEDs can be used for triggering the action of the actuators of solar power, load, battery, grid, etc. Each LED is fixed for a specific microgrid device and turning on the LED reflects the activation of corresponding device through actuators.

4. Advanced energy management of microgrid using MAS and Arduino

Two-grid connected microgrids, one in the department and the other in the hostel, each with solar unit, wind unit, load, battery, grid, and diesel unit, are considered. The model of the microgrid is shown in **Figure 1**.

Each component is considered as an agent and the agent communicates and coordinates for optimal energy management. The agents are decentralized and autonomous for local optimization and coordinate with other agents, distributed across the network for different levels of global optimization. Java agent programming is used to program the agents in JADE environment and executed in Eclipse Integrated Development Environment (IDE). The platform is multithreaded
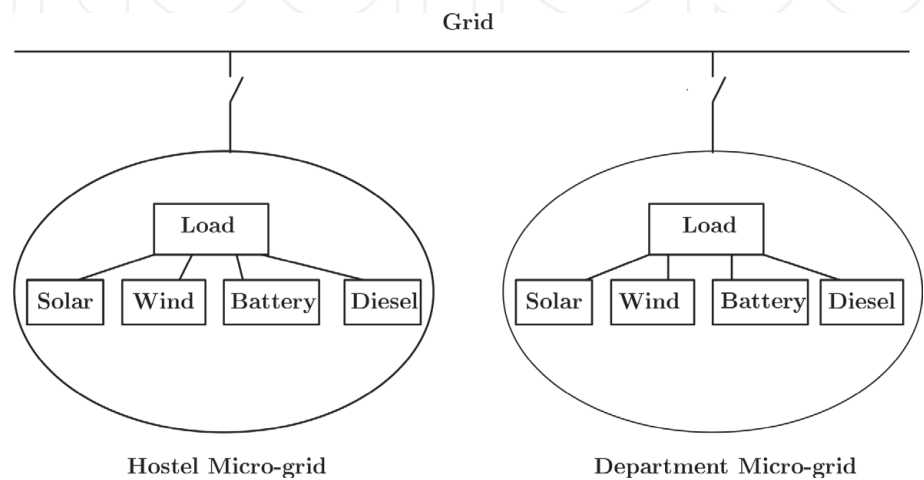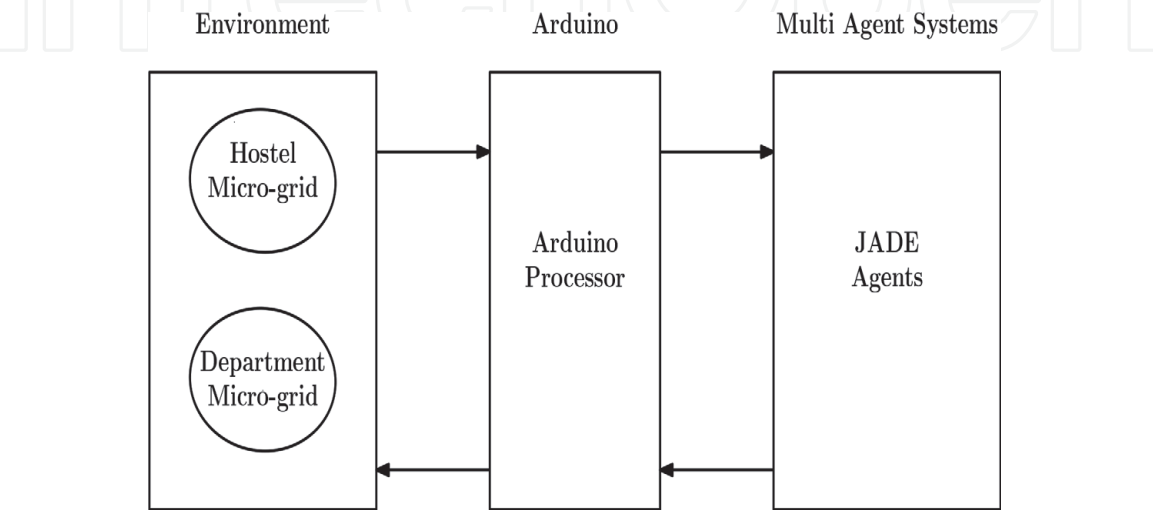

Figure 1.
Solar wind microgrid model.

allowing simultaneous decision-making capabilities for faster reaction and flexibility management. The agents communicate and coordinate with other agents for strategic decision and the resulting command signals are passed to the actuators for physical action. Arduino Mega board, which has more I/O pins to accommodate all the agents, is used for sensing the environment values. MAS is linked with Arduino Mega board for practical verification of operations of agents.

Initially, the department load receives power from the department solar. If it is not sufficient, it receives from the wind unit. If power is still required, it is received from the hostel solar and hostel wind units in the order. If the load requirement is more than the solar and wind power, it checks with the batteries and receives the available power from the batteries. If power is still required, it goes for noncritical load (NCL) shedding in department and hostel loads before going to the external power source of grid or diesel. Dynamic pricing is implemented and hence the choice of diesel or grid is made based on the unit price at that point of time. Similar procedure is followed for hostel load. Thus, the agents coordinate among themselves to take the best possible actions for optimal energy management of microgrid. The environment variables are sensed by using potentiometer where the maximum ranges are fixed for the value (0–1023). For example, if hostel solar power is 100 kW, then 1024 represents 100 kW. Similarly, all the environment variable values are fixed. By varying the potentiometer, these values can be varied within the fixed range. By varying these values, various scenarios are generated for validation and verification. A model for Arduino-based energy management of microgrid is shown in **Figure 2**. Here, the environment values are sensed by Arduino and given to the agents. The agents collaborate and coordinate for effective energy management, and after the operation, the resulting command signals are given to Arduino, which are observed through LED operations. Digital write of high or low is done on the agent when the ACL communication is active. An 8 bit number is considered with each bit representing a microgrid device action. Bit value 1 represents ON state and 0 represents OFF state. These 8 bit output is given through Arduino Mega microcontroller to the corresponding LEDs to show the physical action of the corresponding microgrid devices. In this case, all the LEDs glow, except the last two white LEDs that represent grid and diesel since the loads are not receiving any power from them. The console output of hostel and department units are shown in **Figures 3** and **4**. The sniffer diagram in **Figure 5** shows that the hostel load receives power from hostel solar, hostel wind, and hostel battery and that the department load receives power from department solar, department

Figure 2.
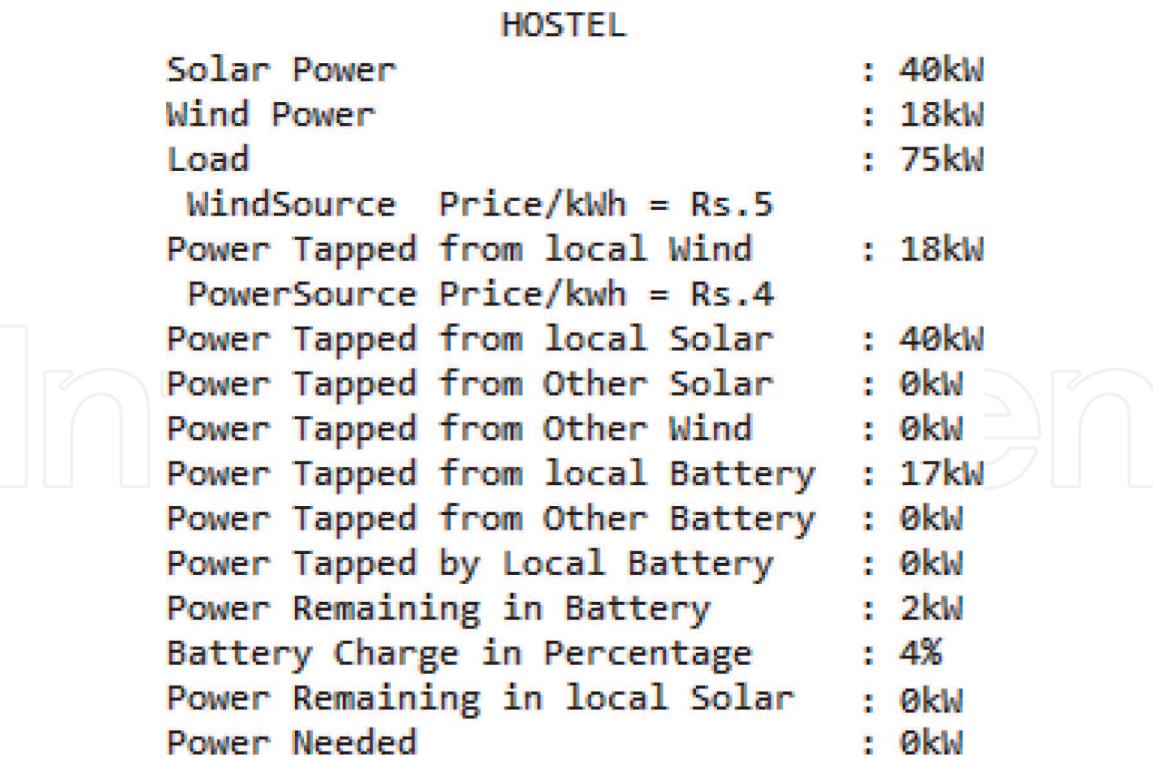*MAS Arduino-based microgrid control model.*

```
                        HOSTEL
        Solar Power                           : 40kW
        Wind Power                            : 18kW
        Load                                  : 75kW
         WindSource  Price/kWh = Rs.5
        Power Tapped from local Wind          : 18kW
         PowerSource Price/kwh = Rs.4
        Power Tapped from local Solar         : 40kW
        Power Tapped from Other Solar         : 0kW
        Power Tapped from Other Wind          : 0kW
        Power Tapped from local Battery       : 17kW
        Power Tapped from Other Battery       : 0kW
        Power Tapped by Local Battery         : 0kW
        Power Remaining in Battery            : 2kW
        Battery Charge in Percentage          : 4%
        Power Remaining in local Solar        : 0kW
        Power Needed                          : 0kW
```

Figure 3.
Console output of energy management of microgrid in hostel.

```
                      DEPARTMENT
        Solar Power                           : 40kW
        Wind Power                            : 21kW
        Load                                  : 70kW
         WindSource1  Price/kWh = Rs.8
        Power Tapped from local Wind          : 21KW
         PowerSource1 Price/kwh = Rs.7
        Power Tapped from local Solar         : 40kW
        Power Tapped from Other Solar         : 0kW
        Power Tapped from Other Wind          : 0kW
        Power Tapped from local Battery       : 9kW
        Power Tapped from Other Battery       : 0kW
        Power Tapped by Local Battery         : 0kW
        Power Remaining in Battery            : 10kW
        Battery Charge in Percentage          : 20%
        Power Remaining in local Solar        : 0kW
        Power Needed                          : 0kW
```
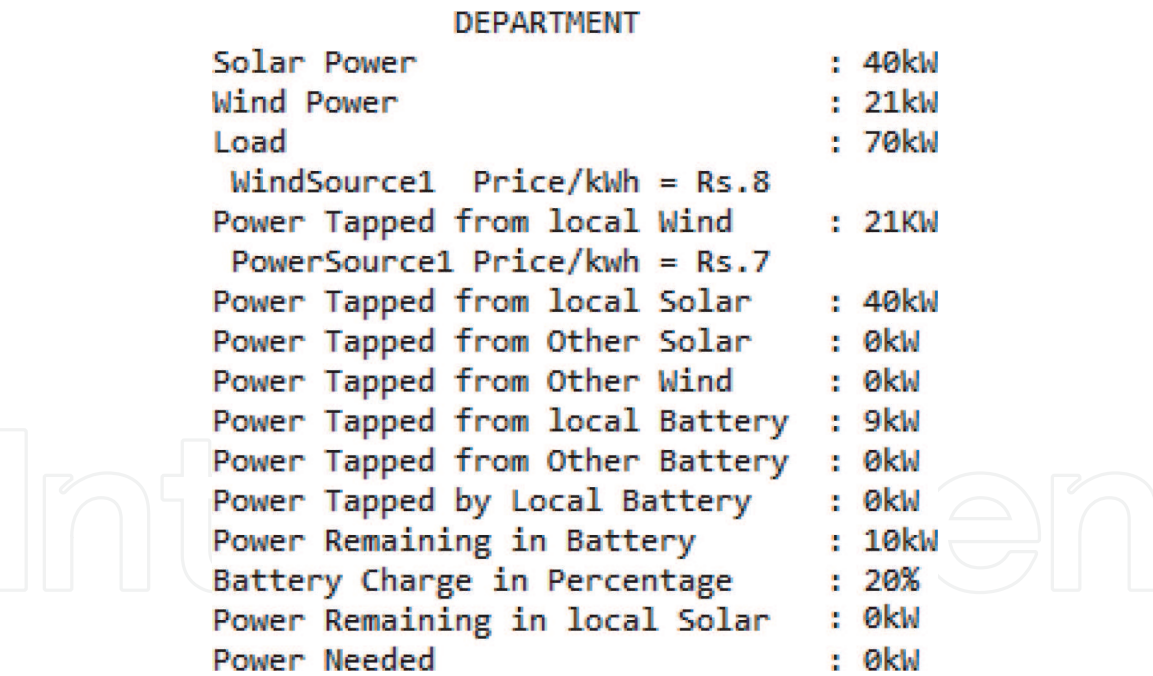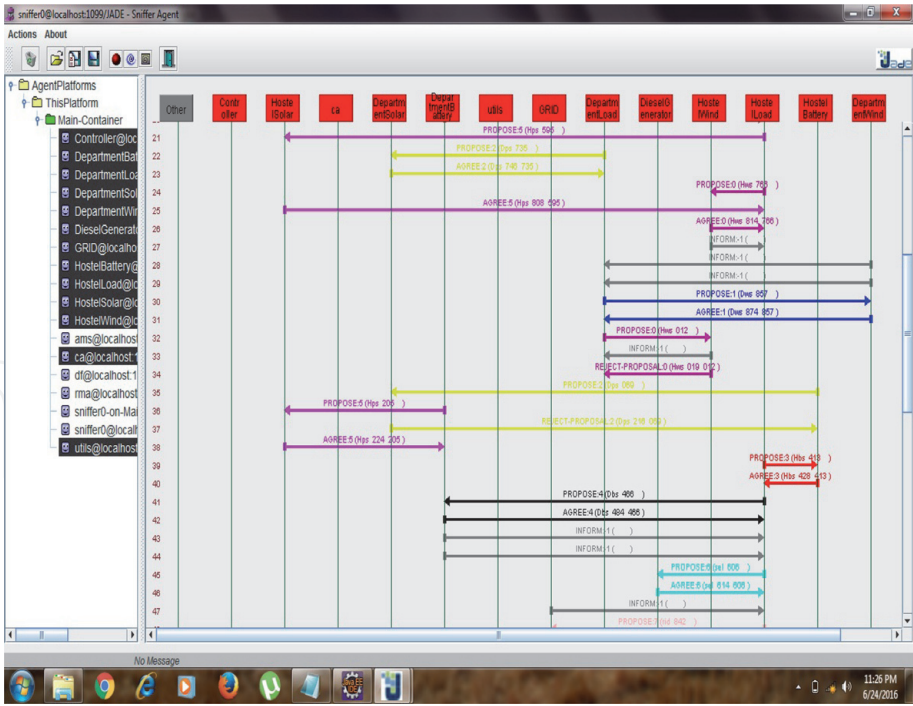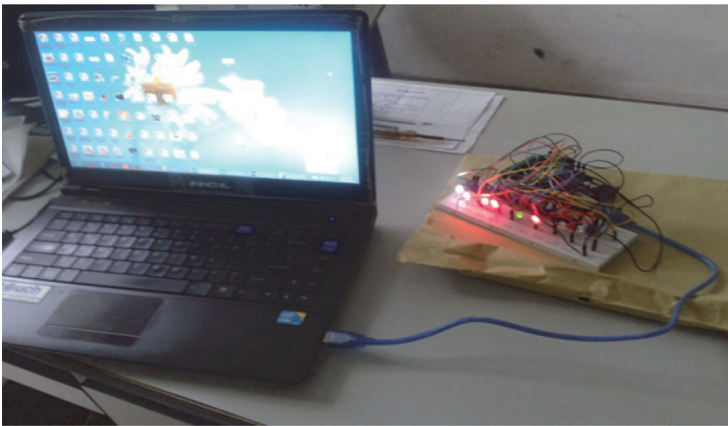
Figure 4.
Console output of energy management of microgrid in department.

wind, and department battery. Both the units are self-sufficient in power requirements and so there is no intercommunication between them. The execution of agents with the status of LEDs is shown in **Figures 6** and 7.

The department and hostel units are self sufficient and hence no power is received from the grid or diesel units as indicated in console outputs. The last two white LEDs that represent grid and diesel are OFF and the other LEDs are ON as shown in the pictures. Thus, the operations of the agents are verified using MAS-Arduino testbed. The inherent features of MAS increase the switching speed and

Figure 5.
JADE sniffer diagram of energy management of microgrid.

Figure 6.
Verification through Arduino output-1.
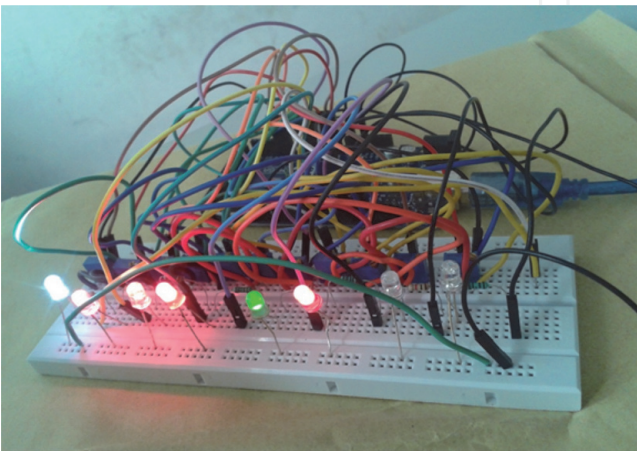
Figure 7.
Verification through Arduino output-2.

7

reduce the network load, which leads to better operational efficiency of microgrids even under intermittent solar power and randomness in load.

5. Verification of plug and play

Plug-and-play systems are used to provide flexible service based on a spontaneous networking facility. The nature of JADE architecture supports the plug-and-play capability and is scalable without much modification to the existing control scheme. So, without much reconfiguration on the system control architecture, agents can be added and removed on the fly. Microgrid dynamically reconfigures to adopt the new resources. Two renewable resource agents, one to the department and the other to the hostel, are added seamlessly by introducing two sensing devices in the Arduino board. The newly added agents are sensed by the system and the system reconfigures itself to accommodate new agents. The new renewable power resource values are accounted and the load agent receives power from external power resources only after considering the newly added agents. In the plug-out, the two agents are removed by removing the corresponding sensing devices connected to the Arduino board. The MAS-based energy management system in the microgrid reconfigures itself and manages the power deficit due to plug-out of agents by receiving power from grid or diesel unit based on their unit price at that point of time.

5.1 Verification of plug-in operation

A solar power resource is added to the department and hostel on the fly by introducing corresponding sensing devices in the Arduino board. The MAS senses the newly added agents and reconfigures itself to accommodate the new agents. In JADE, the plug-and-play operations are executed by walker and cyclic behaviors. When the new agents appear in environment, the walker behavior alerts the directory facilitator (DF). The new agent is given an agent ID by the agent management system (AMS) and it registers with AMS with its AID and also it registers its services with the DF. DF has the subscribe agent class, which will inform to all the agents registered to it about the new agent's arrival. Load agent is allowed to receive power from the new seamless solar agent through DF. The new solar power value is accounted and the load agent considers the seamlessly added solar power resource before going to grid.

The components of microgrid have their corresponding agents in MAS. Java agent programming is implemented in JADE environment and executed in Eclipse Integrated Development Environment (IDE). The input and output terminals of the Arduino Mega board is connected to potentiometers and LEDs, respectively. This Arduino Mega board is connected with computer serial port through Ethernet card. Arduino software is run on computer to upload the program into the board. The potentiometers are fixed with the environment variables of the microgrid. A bread board is fixed with eight LEDs representing two solar units, two wind units, two battery units, grid, and diesel units. The microgrid agents receive their environment variables from the sensing device through the Arduino microcontroller and take strategic decisions by communicating and coordinating with other agents. The resulting command signals are given to Arduino for showing the output through LEDs. The potentiometers' values are varied for various environment values and the ON/OFF operations of LED are verified with environmental dynamics. The console outputs and sniffer diagrams that represent the visual interaction of the agents are studied.

Solar power of 11 and 15 kW is seamlessly plugged in to hostel and department units of the microgrid by physically connecting the corresponding potentiometer pins to the Arduino input board. The microgrid reconfigures itself to account these newly added solar power resources. After receiving power from its internal resources, it receives power from seamless power resources. Then, the hostel load receives 15 kW from diesel unit, as it is cheaper, and the remaining power of 23 kW from grid. The capacity of the diesel unit is taken as 15 kW. The department load first receives power from its internal resources of solar and wind and then from seamless solar power resources. Since there is no diesel power, the department load receives the remaining power of 8 kW from grid irrespective of the price. The console and sniffer outputs of the hostel and department units are shown in **Figures 8–10**. In the sniffer diagram, the interactions of the agents are visually shown. In Arduino output, shown in **Figure 11**, when the two seamless resources are plugged in, all the LEDs go to ON state as all the components of microgrid are operational. Thus, the plug-in operation is implemented seamlessly by the behavior characteristics of JADE in MAS, leading to dynamically adding resources on the fly with autoreconfiguration.

5.2 Verification of plug-out operation

In this case, the two plugged in seamless solar power agents are removed from the microgrid by disconnecting the two sensing devices representing the two seamless solar powers from the Arduino board. The microgrid accounts the absence of the two seamless agents and reconfigures by itself. Then, it receives power from the other available resources to manage the power deficiency due to plug-out action. Here, the seamless power of hostel and department is removed. The DF deregisters the plugged out agents. The DF rejects power request proposal of load agents, which were earlier receiving power from the seamless agents. Now, the load agents do the
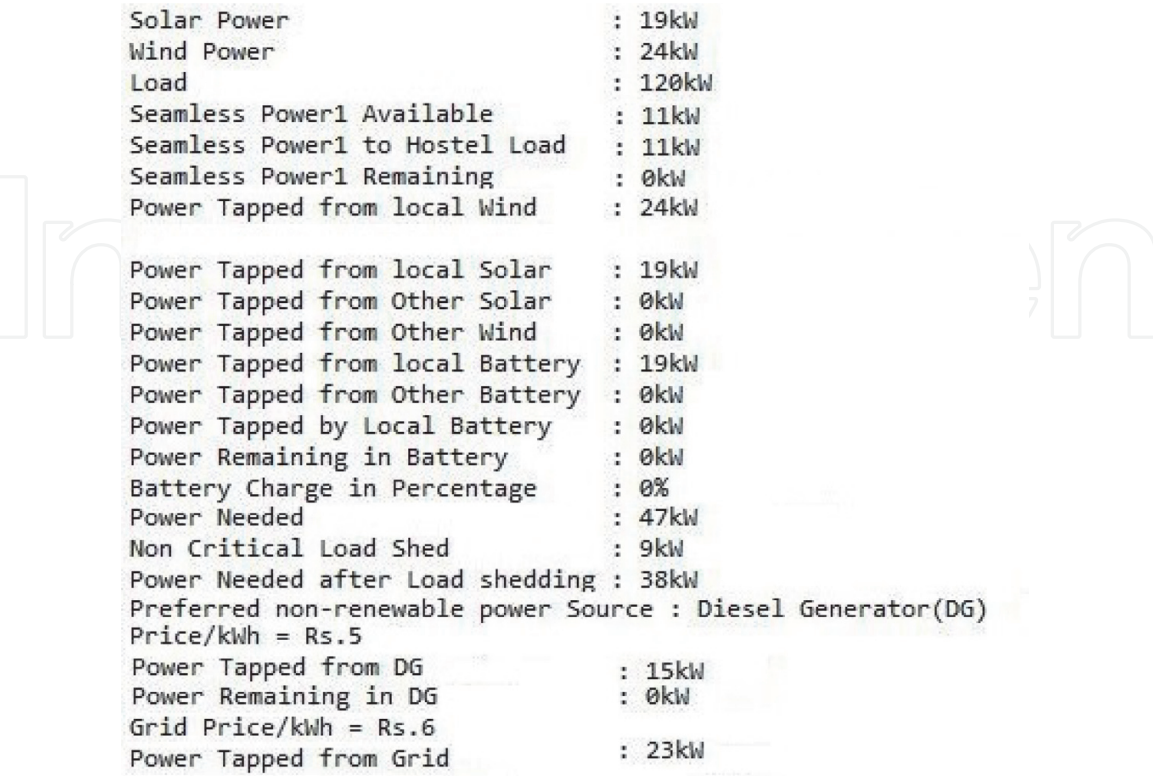
```
Solar Power                          : 19kW
Wind Power                           : 24kW
Load                                 : 120kW
Seamless Power1 Available            : 11kW
Seamless Power1 to Hostel Load       : 11kW
Seamless Power1 Remaining            : 0kW
Power Tapped from local Wind         : 24kW

Power Tapped from local Solar        : 19kW
Power Tapped from Other Solar        : 0kW
Power Tapped from Other Wind         : 0kW
Power Tapped from local Battery      : 19kW
Power Tapped from Other Battery      : 0kW
Power Tapped by Local Battery        : 0kW
Power Remaining in Battery           : 0kW
Battery Charge in Percentage         : 0%
Power Needed                         : 47kW
Non Critical Load Shed               : 9kW
Power Needed after Load shedding     : 38kW
Preferred non-renewable power Source : Diesel Generator(DG)
Price/kWh = Rs.5
Power Tapped from DG                 : 15kW
Power Remaining in DG                : 0kW
Grid Price/kWh = Rs.6
Power Tapped from Grid               : 23kW
```

Figure 8.
Console output of plug and play of microgrid in hostel with seamless power.

```
Solar Power                          : 34kW
Wind Power                           : 21kW
Load                                 : 99kW
Seamless Power2 Available            : 15kW
Seamless Power2 to Dept Load         : 15kW
Seamless Power2 Remaining            : 0kW
Power Tapped from local Wind         : 21kW
Power Tapped from local Solar        : 34kW
Power Tapped from Other Solar        : 0kW
Power Tapped from Other Wind         : 0kW
Power Tapped from local Battery      : 15kW
Power Tapped from Other Battery      : 0kW
Power Tapped by Local Battery        : 0kW
Power Remaining in Battery           : 0kW
Battery Charge in Percentage         : 0%
Power Needed                         : 14kW
Non Critical Load Shed               : 6kW
Power Needed after Load shedding     : 8kW
Power Tapped from DG                 : 0kW
Power Tapped from Grid               : 8kW
```
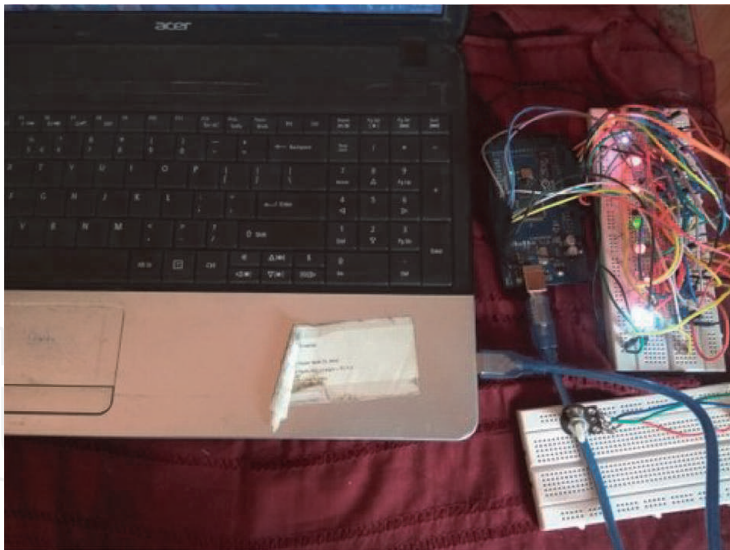
Figure 9.
Console output of plug and play of microgrid in department with seamless power.
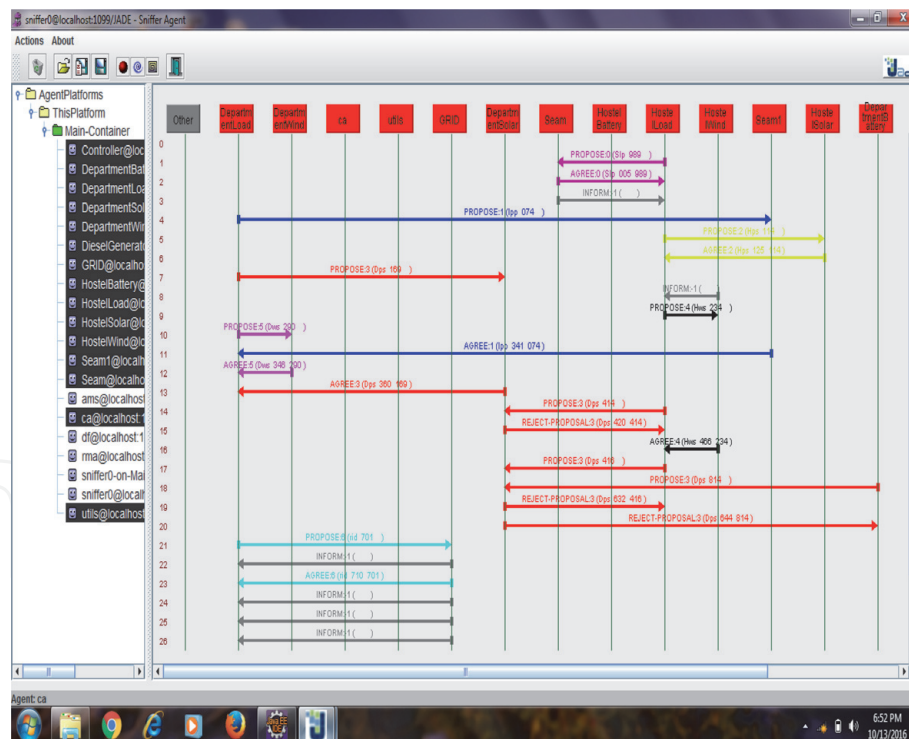
Figure 10.
JADE sniffer diagram of plug and play of microgrid with seamless power.

possible NCL shedding and get the remaining required power from grid or diesel to balance the microgrid after the plugged out event. The microgrid receives power from the grid irrespective of the price as full diesel power is exhausted. In this case, when the seamless solar power is plugged out from the department microgrid, the department load manages the event by receiving 12 kW power from the grid after noncritical load shedding of 6 kW. When the seamless power is plugged out in the hostel, the hostel load agent compensates by receiving 21 kW power from the grid

Figure 11.
Arduino output of microgrid with two seamless agents plugged in.

after NCL shedding of 9 kW. Thus, the plug out of the seamless power is managed by NCL shedding and receiving power from the grid. Since there is no diesel power available, all the external power requirement is received from the grid without considering the price. The console outputs of hostel and department units are shown in **Figures 12** and **13**.

The Arduino output is shown in **Figure 14**. In the sniffer diagram, shown in **Figure 15**, the power proposals to seamless power agents are rejected as they are disconnected. Here, the microgrid receives power from the grid to compensate the loss of power due to plug out event. Since all the microgrid devices are operational except the diesel unit, LED representing diesel unit is in OFF state and all the other

```
                        HOSTEL
Solar Power                           : 40kW
Wind Power                            : 30kW
Load                                  : 120kW
  WindSource   Price/kWh = Rs.5
Power Tapped from local Wind          : 30kW
   PowerSource Price/kwh = Rs.4
Power Tapped from local Solar         : 40kW
Power Tapped from Other Solar         : 0kW
Power Tapped from Other Wind          : 0kW
Power Tapped from local Battery       : 20kW
Power Tapped from Other Battery       : 0kW
Power Tapped by Local Battery         : 0kW
Power Remaining in Battery            : 0kW
Battery Charge in Percentage          : 0%
Power Remaining in local Solar        : 0kW
Power Needed                          : 30kW
Non Critical Load Shed                : 9kW
Power Needed after Load shedding      : 21kW
Power Remaining in DG                 : 0kW
Power Tapped from DG                  : 0kW
Grid Price/kWh = Rs.12
Power Tapped from Grid                : 21kW
```

Figure 12.
Console output of plug and play of microgrid in hostel without seamless power.

```
                    DEPARTMENT
Solar Power                           : 35kW
Wind Power                            : 22kW
Load                                  : 100kW
 WindSource1  Price/kWh = Rs.8
Power Tapped from local Wind          : 0kW
 PowerSource1 Price/kwh = Rs.7
Power Tapped from local Solar         : 35kW
Power Tapped from Other Solar         : 22kW
Power Tapped from Other Wind          : 0kW
Power Tapped from local Battery       : 25kW
Power Tapped from Other Battery       : 0kW
Power Tapped by Local Battery         : 0kW
Power Remaining in Battery            : 0kW
Battery Charge in Percentage          : 0%
Power Remaining in local Solar        : 0kW
Power Needed                          : 18kW
Non Critical Load Shed                : 6kW
Power Needed after Load shedding      : 12kW
Power Tapped from DG                  : 0kW
Grid Price/kWh = Rs.12
Power Tapped from Grid                : 12kW
Power Remaining in DG: 0kW
```
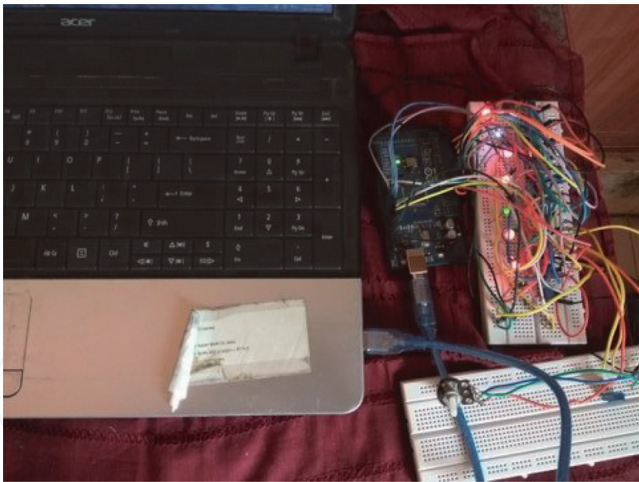
Figure 13.
Console output of plug and play of microgrid in department without seamless power.
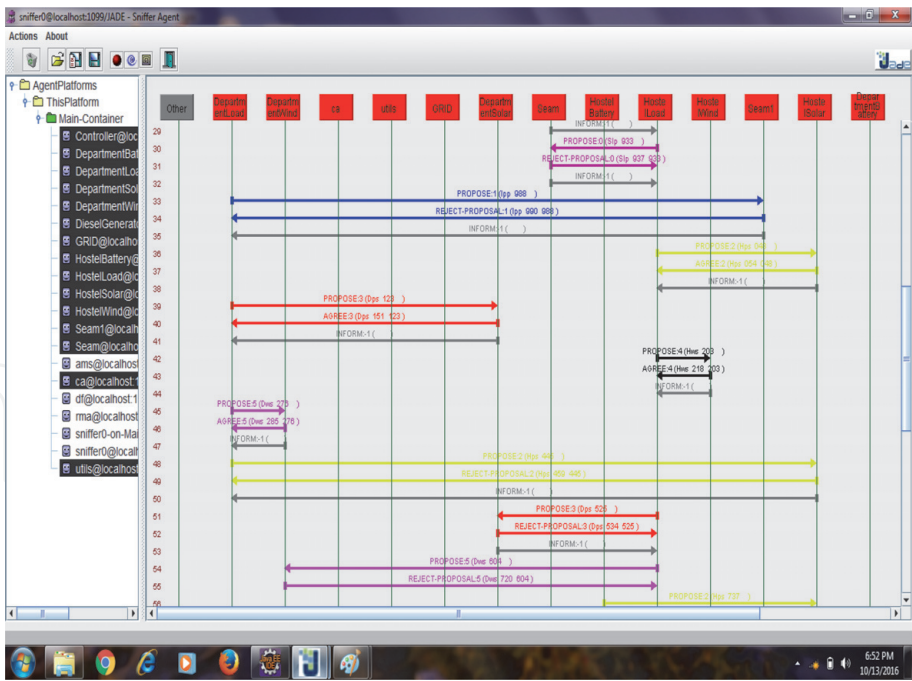

Figure 14.
Arduino output of microgrid with two seamless agents plugged out.

LEDs are in ON state. Thus, the plug-out operation is implemented in the microgrid with behavior characteristics of JADE for autoreconfiguration.

6. Smart microgrid testbed using MAS and Arduino

In future, there may not be one single grid and there may be interconnection of multiple microgrids and prosumers to form virtual grid. The local flexibilities due to dynamic environment in the microgrid are addressed by decentralized multiagent approach for higher level distributed optimization to reduce overall cost in the

Figure 15.
JADE sniffer diagram of plug and play of microgrid without seamless power.

network. The consumer can choose the best resources available in the network for economical and environmental optimization. Also, the consumers can sell their power at best possible price. Effective demand side management can be implemented for optimal use of energy in a distributed environment. Here, a smart microgrid testbed is formed by MAS operations linked with Arduino for verifications of agent operations in supplement to visuals of sniffer diagram. The MAS-based dynamic management leads to economic use of resources in the network. MAS optimizes the use of flexibility provided by the local microgrids and prosumers. Thus, capacity management is implemented to diffuse the stress in the network. The optimization does not involve each end user only as an individual, but empowers them to participate in a larger community for global optimization. The scope of the optimization is to reduce the costs associated with the energy usage of end user. This can be effective in dealing with congestions at local level by optimal flexibility management and enhance performance, reliability, and resilience.

6.1 Simulation and verification of smart microgrid testbed

Three microgrids located in the hostel, department, and administration building are considered. Each microgrid has solar and wind and is grid connected. Each microgrid receives power from the local resources. The further power required is received from other sources of other microgrids by negotiating for better price and eventually getting the power from sources that offer lowest price. In this case, administration and department microgrids have their requirement satisfied by their own resources. In hostel microgrid, the local power is insufficient and so it receives the remaining required power from both administration and department power sources. The total power deficit is calculated and shared equally among the microgrids for maintaining balance in the network. In this case, the total power deficit is 27 kW. The microgrids share it equally and so each microgrid has to shed 9 kW. 3 kW is shed in each microgrid through NCL and the remaining 6 kW is received from the grid. The console output is shown in **Figure 16**. In the department microgrid, the load requires 36 kW. Out of this, the load shedding requirement
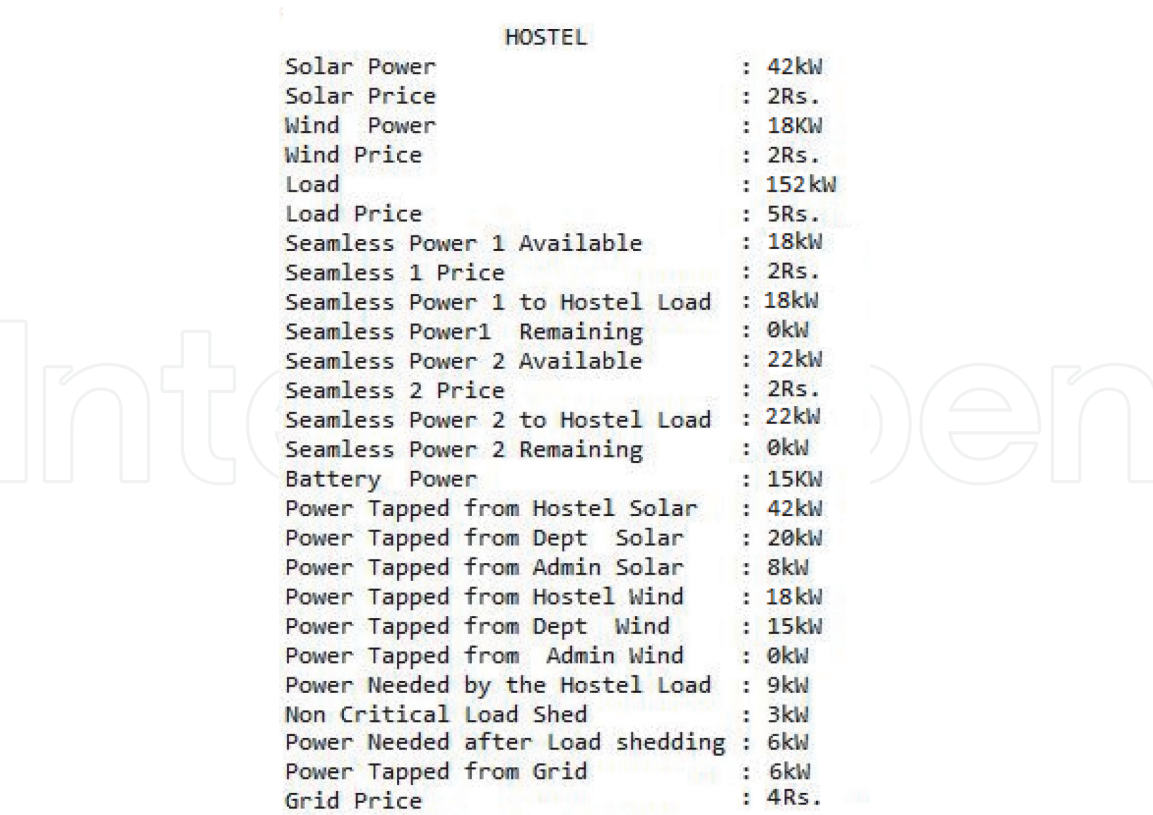
```
                            HOSTEL
        Solar Power                          : 42kW
        Solar Price                          : 2Rs.
        Wind  Power                          : 18KW
        Wind Price                           : 2Rs.
        Load                                 : 152kW
        Load Price                           : 5Rs.
        Seamless Power 1 Available           : 18kW
        Seamless 1 Price                     : 2Rs.
        Seamless Power 1 to Hostel Load      : 18kW
        Seamless Power1  Remaining           : 0kW
        Seamless Power 2 Available           : 22kW
        Seamless 2 Price                     : 2Rs.
        Seamless Power 2 to Hostel Load      : 22kW
        Seamless Power 2 Remaining           : 0kW
        Battery  Power                       : 15KW
        Power Tapped from Hostel Solar       : 42kW
        Power Tapped from Dept  Solar        : 20kW
        Power Tapped from Admin Solar        : 8kW
        Power Tapped from Hostel Wind        : 18kW
        Power Tapped from Dept  Wind         : 15kW
        Power Tapped from  Admin Wind        : 0kW
        Power Needed by the Hostel Load      : 9kW
        Non Critical Load Shed               : 3kW
        Power Needed after Load shedding     : 6kW
        Power Tapped from Grid               : 6kW
        Grid Price                           : 4Rs.
```

Figure 16.
MAS Arduino console output of hostel with three microgrids.

from control agent is 9 kW and so it requires 27 kW power. It receives 27 kW from local solar power source as the price of solar and wind is same. Then, out of 9 kW load shedding requirement from control agent, it sheds 3 kW of NCL and the remaining 6 kW is received from grid. In the administration microgrid, the load requires 52 kW. Out of this, the load shedding requirement is 9 kW and so it requires 43 kW power. It receives 30 kW from local wind and the remaining 13 kW from local solar. Then, out of 9 kW load shedding requirement from control agent, 3 kW is shed through NCL and the remaining 6 kW is received from grid. In the hostel microgrid, the load requires 152 kW. Out of this, the load shedding requirement is 9 kW and so it requires 143 kW of power. It receives 42 kW from local solar and 18 kW from local wind. Then, it receives the remaining available power from
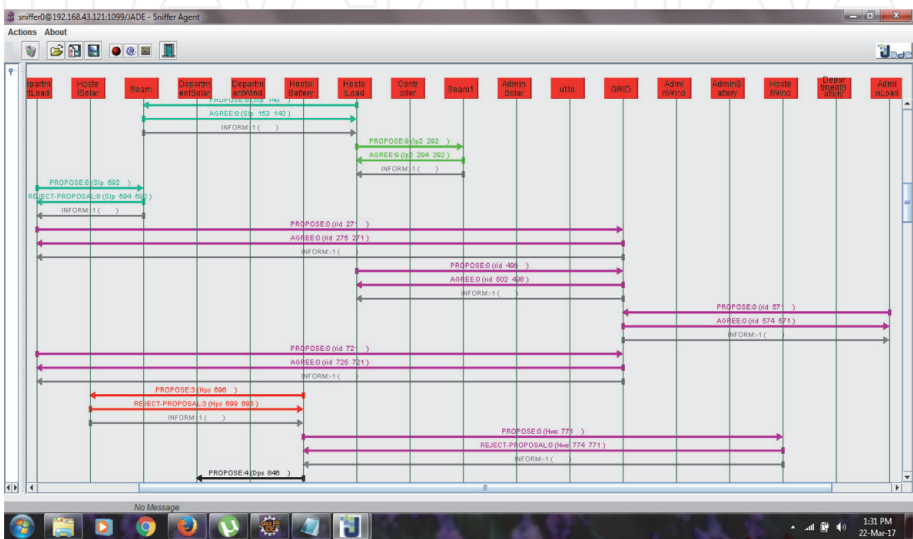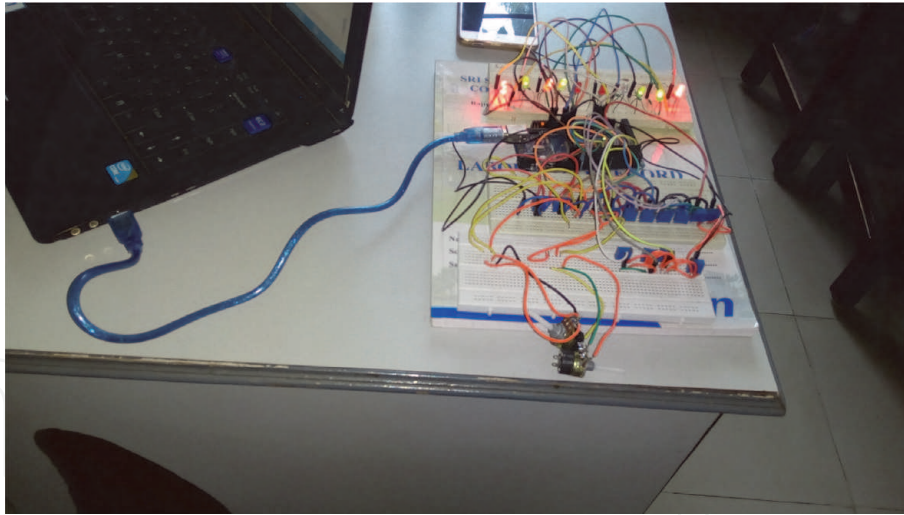

Figure 17.
JADE sniffer diagram 1 with three microgrids.

Figure 18.
Arduino output with three microgrids.

department and administration microgrids. It receives 20 kW from department solar, 8 kW from administration solar, and 15 kW from department wind. Also, it receives 18 kW from seamless source 1 and 22 kW from seamless source 2 through plug-and-play mechanism. Then, out of 9 kW load shedding requirement from control agent, it sheds 3 kW of NCL and the remaining 6 kW is received from grid. The sniffer diagrams are shown in **Figure 17**.

Sniffer diagrams show the complete communication between agents interconnecting all the three microgrids. The Arduino output is shown in **Figure 18**. In the Arduino output, except the three battery LEDs, the other LEDS are in ON state as all these devices are activated. Thus, dynamic energy management is implemented in smart microgrid testbed using MAS and Arduino. This work can be scaled up to large number of microgrids to form smart grid.

7. Conclusions

In this chapter, we have carried out a practical verification of multiagent operations in controlling the distributed energy management in a microgrid by linking MAS with Arduino microcontroller to form a smart microgrid testbed. This work forms a foundation for dynamic energy management in a distributed network of microgrids. The multiagent operation in plug and play, which is the significant feature of smart grid, is practically verified by adding and removing the renewable power resources in the smart microgrid testbed. Optimal flexibility management is implemented in smart microgrid testbed with a network of three microgrids, for best use of resources in a MAS-based distributed and dynamic environment. This method of using multiagent system to control the microgrid operation in a smart microgrid can be extended for effective energy management and demand side management across the networks so that the consumer can receive the best available resource, considering the dynamic nature of the environment. The prosumers can dynamically receive or sell the power at optimal price across the network. Also, the control unit can influence the customer by varying the price, leading to demand response management. This smart microgrid testbed can be scaled up to include large number of microgrids, and IoT (Internet of Things) can be incorporated in it to make smart grid a reality. After due diligence of experimental verification, real-time implementation of MAS-based autonomous distributed energy management of

cyber physical microgrid can be implemented for economic and environmental optimization. Parallel and autonomous operation, asynchronous communication, faster computation, and flexibility management make MAS-based cyber physical system (CPS) a novel solution to large complex adaptive systems. This approach has its significance in the present energy Internet scenario as the energy industry is exploring distributed network through block chain for identifying opportunities for optimizing operations.

Abbreviations

ICT   information and communication technology
MAS   multiagent systems
DER   distributed energy resources
JADE   Java agent development environment
LED   light-emitting diode
ACL   agent communication language
NCL   noncritical load
DF   directory facilitator
IDE   integrated development environment
IOT   Internet of Things
CPS   cyber physical system

Author details

Leo Raju[1,2*†] and Antony Amalraj Morais[2]

1 SSN College of Engineering, Chennai, India

2 Tiger Analytics, Bangalore, India

*Address all correspondence to: leor@ssn.edu.in

†The authors contributed equally.

IntechOpen

References

[1] Odell J. Objects and agents compared. Journal of Object Technology. 2015;**1**(1):41-53. DOI: 10.5381/jot.2002.1.1.c4

[2] Russell S, Norvig P. Artificial Intelligence. In: A Modern Approach. Pearson Education Inc; 2011. DOI: 10.1016/j.artint.2011.01.005

[3] Jennings NR, Bussmann S. Agent-based control systems. IEEE Control Systems. 2003;**23**(3):61-74. DOI: 10.1109/MCS.2003.1200249

[4] Wooldridge M. An Introduction to MultiAgent Systems. West Sussex, United Kingdom: John Wiley Press; 2010

[5] Pipattanasomporn M, Feroze H, Rahman S. Multi-agent systems in a distributed smart grid: Design and implementation. In: Proceedings of the IEEE Power Engineering Society Power Systems Conference (IEEE-PES)-2009; 15–18 March 2009; Seattle, WA, USA: IEEE. 2009. pp. 1-8. DOI: 10.1109/PSCE.2009.4840087

[6] Dimeas AL, Hatziargyriou ND. Agent based control of virtual power plants. In: Proceedings of the International Conference on Intelligent Systems Applications to Power Systems. 2007; 5–8 Nov. 2007; Toki Messe, Niigata, Japan: IEEE. 2007. pp. 1-6. DOI: 10.1109/ISAP.2007.4441671

[7] Logenthiran T, Srinivasan D, Khambadkone AM, Aung HN. Multiagent system for real-time operation of a microgrid in real-time digital simulator. IEEE Transactions on Smart Grid. 2011;**3**(2):925-933. DOI: 10.1109/TSG.2012.2189028

[8] Nunna HSVSK, Doolla S. Multiagent-based distributed-energy-resource management for intelligent microgrids. IEEE Transactions on Industrial Electronics. 2013;**60**(4):1678-1687. DOI: 10.1109/TIE.2012.2193857

[9] Gomez-Sanz J, Garcia-Rodriguez S, Cuartero-Soler N, Hernandez-Callejo L. Reviewing microgrids from a multi-agent systems perspective. Energies. 2014;**7**(5):3355-3382. DOI: 10.3390/en7053355

[10] Abhilash K, Laura EB, Gordon P, Wayne WW. Survey of multi-agent systems for microgrid control. Engineering Applications of Artificial Intelligence. 2015;**45**:192-203. DOI: 10.1016/j.engappai.2015.07.005

[11] Bellifemine F, Caire G, Greenwood D. Developing Multi-Agent Systems with JADE. West Sussex, United Kingdom: John Wiley Press; 2007. DOI: 10.1002/9780470058411

[12] Diwakar Purusothaman SRR, Rajesh R, Bajaj KK, Vijayaraghavan V. Implementation of Arduino-based multi-agent system for rural Indian microgrids. In: Proceedings of the IEEE Innovative Smart Grid Technologies (ISGT Asia, 2013); 10–13 November 2013; Bangalore, India. 2013. pp. 1-5. DOI: 10.1109/ISGT-Asia.2013.6698751

[13] Diwakar Purusothaman SRR, Rajesh R, Bajaj KK, Vijayaraghavan V. Design of Arduino-based communication agent for rural Indian microgrids. In: Proceedings of the IEEE Innovative Smart Grid Technologies (IGST-Asia, 2014); 20–23 May 2014; Kuala Lumpur, Malaysia. 2014. pp. 630-634. DOI: 10.1109/ISGT-Asia.2014.6873865

[14] Perkonigg F, Brujic D, Ristic M. Platform for multiagent application development incorporating accurate communications modeling. IEEE Transactions on Industrial Informatics. 2015;**11**(3):728-736. DOI: 10.1109/TII.2015.2428633

[15] Kumar Nunna HSVS, Srinivasan D. Multiagent-based transactive energy framework for distribution systems with smart microgrids. IEEE Transactions on Industrial Informatics. 2017;**13**(5):2241-2250. DOI: 10.1109/ TII.2017.2679808

[16] Leo Raju MRS, Antony Amalraj M, Ramyaa R, Soundaryaa P, Thavam LD. Micro-grid outage management using multi agent system. Energy Procedia. 2017;**117**:112-119. DOI: 10.1016/j. egypro.2017.05.113

[17] Raju L, Antony Amalraj M, Milton RS. Advanced energy management of a micro-grid using Arduino and multi-agent system. Intelligent and Efficient Electrical Systems. 2018;**118**(10):65-76. DOI: 10.1007/978-981-10-4852-46