

Article

Agent-Based Coordinated Control of Power Electronic Converters in a Microgrid

Maryam Nasri ^{1,*}, Herbert L. Ginn III ² and Mehrdad Moallem ³

¹ Department of Electrical Engineering Technology, Alfred State College of Technology, The State University of New York, Alfred, NY 14802, USA

² Department of Electrical Engineering, College of Engineering and Computing, University of South Carolina, Columbia, SC 29208, USA; ginnhl@cec.sc.edu

³ School of Mechatronic Systems Engineering, Simon Fraser University, Surrey, BC V3T 0A3, Canada; mmoallem@sfu.ca

* Correspondence: nasrim@alfredstate.edu

Abstract: This paper presents the implementation of an agent-based architecture suitable for the coordination of power electronic converters in stand-alone microgrids. To this end, a publish-subscribe agent architecture was utilized as a distributed microgrid control platform. Over a distributed hash table (DHT) searching overlay, the publish-subscribe architecture was identified based on a numerical analysis as a scalable agent-based technology for the distributed real-time coordination of power converters in microgrids. The developed framework was set up to deploy power-sharing distributed optimization algorithms while keeping a deterministic time period of a few tens of milliseconds for a system with tens of converters and when multiple events might happen concurrently. Several agents participate in supervisory control to regulate optimum power-sharing for the converters. To test the design, a notional shipboard system, including several converters, was used as a case study. Results of implementing the agent-based publish-subscribe control system using the Java Agent Development Framework (JADE) are presented.

Keywords: distributed control; smart grids; microgrid; multi-agent system; distributed hash table; JADE



check for updates

Citation: Nasri, M.; Ginn, H.L., III; Moallem, M. Agent-Based Coordinated Control of Power Electronic Converters in a Microgrid. *Electronics* **2021**, *10*, 1031. <https://doi.org/10.3390/electronics10091031>

Academic Editor:
Mohamed Benbouzid

Received: 13 January 2021
Accepted: 20 April 2021
Published: 26 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Microgrids are groups of power energy sources, loads, storage systems, load controllers, and local networks that are created to provide an energy solution for a community and operated as an electrical island or connected to larger scale power grids [1]. A versatile range of power electronic converters can be utilized to connect sources, loads, and power storage systems to the grid in a microgrid. Coordinating a power system requires directing the flow of power between its primary energy sources and loads through power electronic converters [2]. Comparing the smaller size of microgrids to the large scale power grids, they are more flexible in many cases, and different system-level control functions have been used in both islanded connected modes [3].

Some control techniques, including minimizing power losses, balancing load flow, reducing fuel costs, and defining limits for power components like active and reactive power, can be utilized at a system-level [4,5]. A coordinated and comprehensive control system must be deployed to achieve such system optimization goals [6]. In some methods, supervisory control is used as a higher control layer to coordinate local controllers [7]. Considering that the number of active power electronic converters in a microgrid may change, the control topology needs to meet the up-limit time in the number of real-time message exchanges and disregard the number of converters operating at a given moment [8].

Aggregations of centralized and decentralized control systems have been utilized in smart grids [9,10]. More than eighty percent of the twenty sample microgrid projects in North America have used decentralized control systems [11]. It should be noted that

most conventional power control systems behave proactively or responsively to events, whereas recently developed control systems include active control alternatives to their control techniques [12]. The main specifications for choosing a suitable control strategy can be considered the size and status of the power system, including connected and stand-alone modes [13]. A comprehensive and coordinated control model is needed to freely empower each power component's load control to achieve the local and system-level control functions in a distributed power system. Studies performed on distributed control techniques have demonstrated that agent-based technologies can meet the distributed converter control requirements [14].

New software technologies should be utilized to cope with interoperability and distribution [15]. Agent-based technologies are considered a solution for realizing such systems because they were created to address interoperability and distribution [16]. However, agent-based technologies are new, and they have been deployed in only a handful of power system applications due to complications in their control coordination [17–19].

An agent is a software program running in an environment that can autonomously handle changes in that environment [20]. As a group of intelligent entities, agents can communicate and alter their behavior [21]. A multi-agent system (MAS) is a system that includes two or more agents that are capable of communication. Multi-agent control systems can provide higher degrees of reliability and efficiency of power generation and consumption in power systems, including distributed energy resources [22–24]. This is because they work inside a wholly distributed system, where each node can connect to other nodes at one or multiple layers [25]. In MASs, each agent processes portions of data locally and transfers results to destination agents. Thus, the overall computation time can be vastly reduced compared to more centralized control models if the impact of communication does not outweigh the benefits gained through distributed computation.

Moreover, MASs allow extensibility, such as performing new tasks or communicating a new set of data that becomes available, as well as scalability, like when new resources, loads, or interconnections are added to the system [26]. In this work, considering the high level of flexibility, reliability, and extensibility, an agent-based control system is applied to coordinate power electronic converters in a stand-alone microgrid to control a cost function system level.

2. Proposed Multi-Agent Control Architecture

System-level coordination in a microgrid requires a coordinating system that operates within a bounded time so that the system control is deterministic. Thus, to supervise the coordination among control agents in a microgrid, a message count is determined and used as a key metric. The message count must be bounded within a system-level control cycle and proportional to its upper limit. Thus, the impact of control node scaling on message count must be analyzed in order to determine if a particular agent architecture is feasible for microgrid system-level control.

Several agent architectures have been proposed in the literature, but many are not suitable for deterministic control with a tightly bounded message exchange. Several messaging technologies and their associated lookup algorithms were reviewed and evaluated [27] based on the number of required message exchanges. The algorithms, as mentioned earlier, included: (1) belief-desire-intention (BDI) architecture, using a bidding lookup algorithm; (2) publish-subscribe agent technology working on a DHT overlay lookup algorithm; and (3) facilitator agent architecture, using an ordered multicast searching algorithm. Analysis of the required message exchange as a function of the number of control nodes led to selecting publish-subscribe as the appropriate agent technology to coordinate converters in a microgrid.

Publish-subscribe is a MAS messaging pattern that transfers messages from publishers (senders) to the assigned subscribers (receivers). Topic or content are used to categorize published messages into different classes. Subscribers state their interest in some contents or topics beforehand and receive only those messages [28].

Using content rather than the destination to address a message is considered the unique concept of publish-subscribe systems and is called content-based addressing. This addressing method is utilized to split a network into some freely coupled data producers and data consumers. The publish-subscribe messaging pattern is depicted in Figure 1. Topic-based publish-subscribe lookup algorithms, such as the grid quorum based publish-subscribe system (GQPS), decrease the cost of searching for broadly distributed and loosely coupled grids compared to the brokerage architecture; however, searching flexibility is more limited due to the topic-based design [29].

In addition to agent architecture, methods for message routing are needed. DHTs are distributed searching algorithms used as an infrastructure for efficient, expandable resource lookup in distributed peer-to-peer networks. They are a type of distributed searching algorithm. Ferry [30] is an architecture for content-based publish-subscribe services. It also exploits embedded trees in the underlying DHT to collect and classify data for subsequent data discovery services. An overlay network is made on top of an existing network, and this is a suitable solution for supporting distributed algorithms. DHT is an overlay network that is developed on the application layer of TCP/IP Ethernet networks. It uses hash table capabilities to supervise the nodes' join and leave in a wide-area environment. There are sets (key, value) in hash tables that help applications find a value associated with the given key. Although publish-subscribe can be implemented on a centralized overlay network in small-scale systems to reduce costs, it is preferable to use DHT. The latter does not require changes in system architecture for expanding networks over time. Furthermore, there is no single point of failure in DHT as compared to centralized algorithms. Some studies have been done on the application of publish-subscribe over DHT in large-scale distributed systems [31,32].

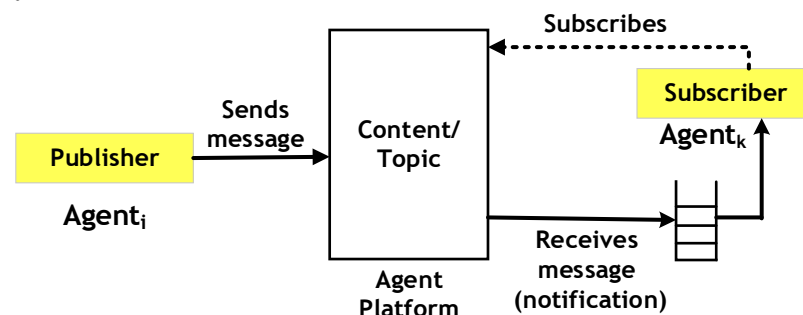


Figure 1. An asynchronous message exchange model designed using content-based publish-subscribe.

Figure 2 shows a computational complexity comparison of the selected method with a popular broker method. Using a bidding searching algorithm, Broker agent technology is one of the most often used agent technologies. The graph shows that a combination of publish-subscribe over DHT is a scalable agent technology for coordination among agents in medium-sized microgrids where message count is the primary concern [33]. By deploying this technology, the upper-time limits of message exchange are only slightly affected for a microgrid with a variation from 5 to 50 control nodes.

Assuming a network consisting of N nodes, publish-subscribe can route to the numerically closest node to a given key in less than $\log_{2^b} N$ steps under a regular operation, where N is the number of nodes and 2^b is a parameter with a standard value of 2 [34]. Ratnasamy and Francis [35] presented a formula to calculate the complexity of a string-attribute message as $O(r * \log_{2^b} N)$ where r is the average length of string values. As the number of nodes increases, this value grows much more slowly compared to the number of message counts for the bidding algorithm, which can be calculated as N^2 .

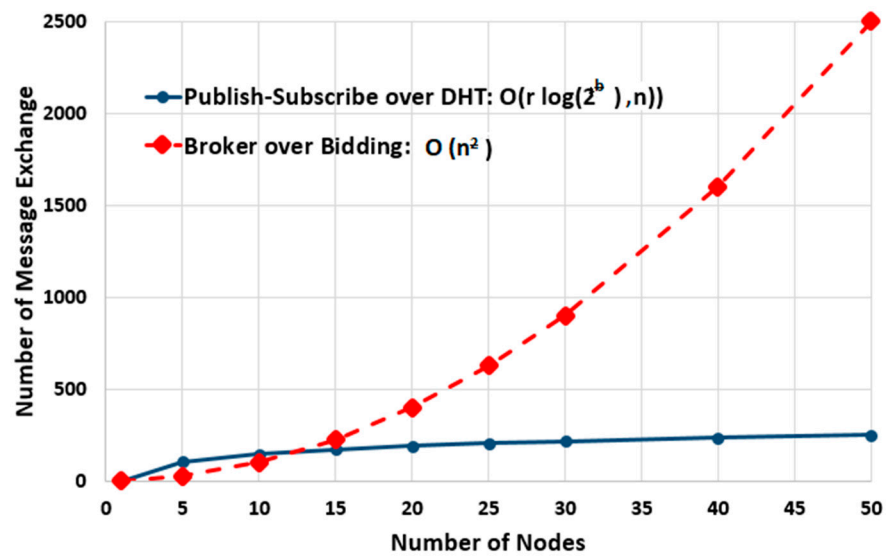


Figure 2. A comparison between two agent technologies’ complexity using message count metrics values: publish-subscribe and bidding algorithm.

3. Coordinated Control for Shipboard Power SYSTEMS

A shipboard power system is a stand-alone microgrid with many power electronic converters coordinated to achieve a range of system-level goals [36]. It is an excellent example of a microgrid that can benefit from a MAS approach to system-level control.

An example shipboard power system, shown in Figure 3, is a subset of a notional DC shipboard distribution system [37]. All significant sources and load centers connected via power electronic converters are denoted as power conversion modules (PCMs) and include energy storage. An electrochemical energy storage system (ESS) and two fuel-based generators have been designed in the microgrid. The ESS can function both as a source or a load depending on the system requirement and battery state of charge (SOC) condition [38]. Loads in the system are clustered into zones.

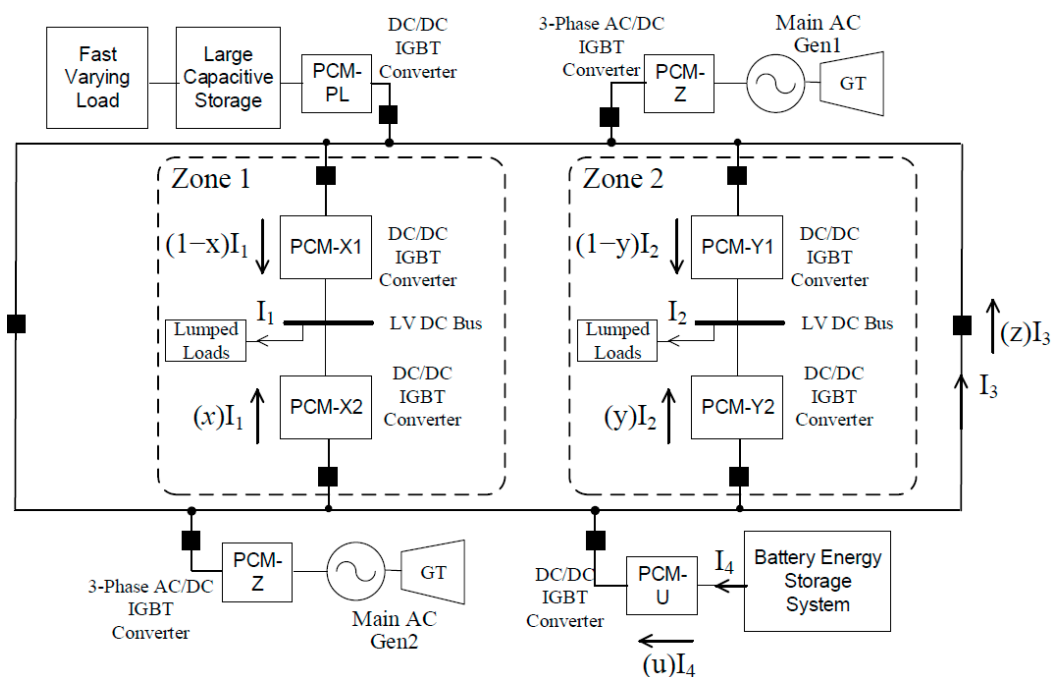


Figure 3. The architecture and electrical model of the notional shipboard microgrid.

The main bus level control is above the zonal level and controls the system level's energy flow. The bus cluster control system controls the bus-tie current for all parallel bus-tie branches connecting two bus clusters. Therefore, system-level control can decide how energy flows across the bus-tie and into each zone. Two main buses form the backbone of the microgrid, the starboard side bus, and the portside bus. Two cross-tie disconnect switches connect them. The battery energy storage system and fast varying load are located on different busses to demonstrate energy flow across the shipboard system.

A master-slave architecture has been applied to zonal level control, where the PCM zonal converter works as a master to control the in-zone voltage, and the PCM slave converter tracks a particular percentage of the master converter's output. System-level control is deployed in the upper level to manage which converter is the master and the sharing percentage between 0 and 100. The PCM converters connected to the load center are considered unidirectional, and they cannot allow energy flow from one bus to another. The fast varying load is high power and highly flexible load that contains its large scale capacitive storage buffer. The rate of change of demand due to charging the capacitive storage is higher than zonal loads.

Each zone or load center and the storage and bus-tie current introduces one control variable. For the system shown in Figure 3, there are four control variables. These variables determine the role of energy storage, branch energy flow, and inter-flow of energy between buses, thereby determining the generators' operating points. A publish-subscribe agent-based control model has been designed to execute a system-level optimization function that determines the distributed converter coordination requirements for both zonal and system levels. Agents allow for much faster processing of the cost function by distributing the computational burden.

4. An Optimal Sharing Algorithm to Deploy in the Proposed Publish-Subscribe MAS

The proposed publish-subscribe MAS system enables a robust and expandable implementation of a system-level optimization based microgrid control. An appropriate distributable optimization algorithm is needed to run on MAS. For the shipboard power system, an optimization algorithm was developed in [39]. The optimization algorithm dynamically identifies the optimal global values for distributed variables to minimize system losses. Optimizing the power-sharing leads to minimizing fuel and distribution losses while applying constraints, such as equipment power ratings. This algorithm can distribute the optimization task among intelligent software modules, making it a perfect candidate for applying the proposed multi-agent control system. The optimization framework is summarized below.

In a shipboard power system, losses are mainly due to fuel usage inefficiency, and a much smaller amount is due to line losses. The cost due to fuel usage inefficiency of an i_{th} generator can be expressed as

$$C_i = aP^2 + bP + c \quad (1)$$

$$C_{MG}(x) = a_{MG}I_{MG}^2 + b_{MG}I_{MG} + c_{MG} \quad (2)$$

$$C_{AG}(x) = a_{AG}I_{AG}^2 + b_{AG}I_{AG} + c_{AG} \quad (3)$$

where a_{MG} , b_{MG} , c_{MG} , a_{AG} , b_{AG} , and c_{AG} are constant dependent values for the main and auxiliary generators. Voltage magnitudes are nearly constant, allowing cost expressions in terms of current. The distribution losses are expressed as

$$C_{Dis}(x) = C_{Dis1}(x) - C_{Dis2}(x) + C_{Dis3}(x) - C_{Dis4}(x) + Const \quad (4)$$

where

$$\begin{aligned} C_{Dis1}(x) &= x^2 I_1^2 (R_1 + R'_1 + R_2 + R'_2) + y^2 I_2^2 (R_2 + R'_2) + z^2 I_3^2 (R_2 + R'_2 + R_3) + u^2 I_4^2 (R_2 + R_4 + R_v) \\ C_{Dis2}(x) &= 2x I_1^3 (I_1 R'_1 + I_1 R'_2 + I_2 R'_2 + I_5 R'_1 + I_5 R'_2) + 2y I_2 (I_1 + I_2 + I_5) R'_2 + 2z I_3 (I_1 + I_2 + I_5) R'_2 \\ C_{Dis3}(x) &= 2xy I_1 I_2 (R_2 + R'_2) + 2xz I_1 I_3 (R_2 + R'_2) + 2yz I_2 I_3 (R_2 + R'_2) \\ C_{Dis4}(x) &= 2xu I_1 I_4 R_2 + 2yu I_2 I_4 R_2 + 2zu I_3 I_4 R_2 \end{aligned}$$

where C_{Dist} is the distribution loss, I_1 is zone 1 load current, I_2 is zone 2 load current, I_3 is the maximum allowed cross-tie (inter-bus) current, I_4 is the maximum charging/discharging current of the ESS, I_{PL} is pulsed load charging current, and the line resistances and converter losses are lumped using R_i values.

An overall cost function is a linear form of the various loss components expressed as

$$C_{sys}(\mathbf{x}) = C_{MG}(\mathbf{x}) + C_{AG}(\mathbf{x}) + C_{Dist}(\mathbf{x}) \tag{5}$$

The variables ‘ x ’, ‘ y ’, ‘ z ’, ‘ u ’ are cost functions that display the flow of energy through various paths and must be optimized by a distributed control system to ensure minimum losses. The current limits on the supply and constraints on the ramp rate can be expressed as

$$g_1(\mathbf{x}) = \Delta I_{gm} \leq (\Delta I_{gm})_{\max} \tag{6}$$

$$g_2(\mathbf{x}) = \Delta I_{gx} \leq (\Delta I_{gx})_{\max} \tag{7}$$

$$g_3(\mathbf{x}) = xI_1 + yI_2 + zI_3 - uI_4 \leq (I_{gm})_{\max} \tag{8}$$

$$g_4(\mathbf{x}) = (1 - x)I_1 + (1 - y)I_2 + I_5 - zI_3 \leq (I_{gx})_{\max} \tag{9}$$

where $(I_{gi})_{\max}$ denotes the i th generator current maximum value, and ΔI_{gi} is the rate of change of the generator current within the generator controller’s time steps.

The control variables are discretized to utilize the mixed integer quadratic programming (MIQP) optimization algorithm. Furthermore, the equation is reformulated as n one-dimensional search equations that help to make the system distributable.

$$C_{sys}(\mathbf{x}) = z_1^2 + z_2^2 + z_3^2 + z_4^2 = \sum_{i=1}^n z_i^2 \tag{10}$$

As ‘ z_1 ’ is an independent variable, it has been considered the primary variable to initiate a search tree, where ‘ z_2 ’, ‘ z_3 ’, and ‘ z_4 ’ act as consecutive followers. All variables are squared in the modified cost function to make a parabolic form of the total loss that helps many search trees be pruned away. These pruned variables can be the right or left side of the absolute minimum of a variable, depending on the constraints. Figure 4 displays a search tree along with a pruning and optimal solution. The dashed lines indicate values that are pruned without introducing approximation into the optimum solution [40]. A search tree is initiated with each existing ‘ z_1 ’ value. Then another span of ‘ z_2 ’ and corresponding loss component for every ‘ z_1 ’ is calculated. Values of ‘ z_2 ’ would also be pruned as was done for ‘ z_1 ’, and so on for the other variables.

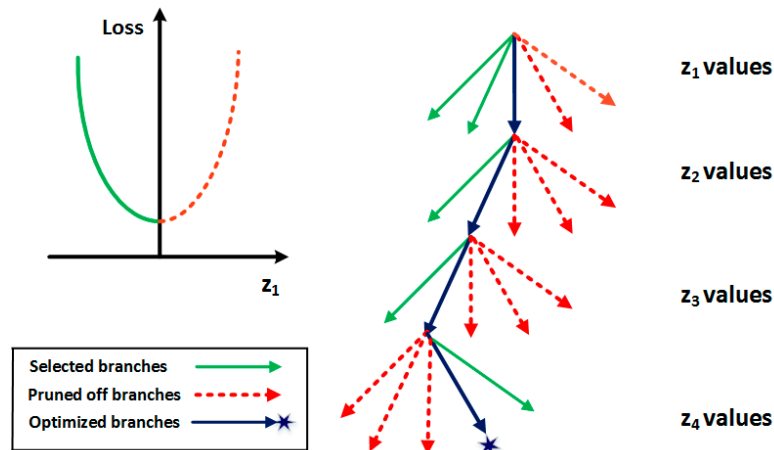


Figure 4. A search tree using an MIQP (Mixed Integer Quadratic Programming) optimization algorithm with pruning.

5. Publish-Subscribe MAS for the Shipboard Microgrid

An agent-based system was designed considering the optimization algorithm described in Section 4. The application of the publish-subscribe model increases the convergence rate of the cost function solution. This is accomplished by running different search trees simultaneously in local agents and exchanging the data for finding the globally optimal values. The distribution of the workload allows for better system expandability within the system control time limits. The top level of the search tree for the MIQP formulation of the cost function forms the main tree branch per variable. Thus, the number of agents is selected equal to the number of control variables, and four agents, designated $Ag1$, $Ag2$, $Ag3$, and $Ag4$, use individual search trees to optimize values of ' z_1 ', ' z_2 ', ' z_3 ', and ' z_4 '.

In a content-based model, the publisher can express its interest to subscribers by specifying a period of defined values over different attributes. The content-based publish-subscribe is made of (attribute, operator, value) tuples, where operators can be one of ($<$, $=$, $>$, \leq , \geq). Since $Ag1$, which associates with the first search tree level, determines the start of several search trees, it was chosen as a subscriber. The other agents calculate ' z_i ' values for their search trees and publish them to subscriber agents.

$$x = f(z_1) \quad (11)$$

$$y = f(z_1, z_2) \quad (12)$$

$$z = f(z_1, z_2, z_3) \quad (13)$$

$$u = f(z_1, z_2, z_3, z_4) \quad (14)$$

Each search tree is denoted by a tuple, where the definition of subscriber agents are

$$S1 = Ag1 = \{(x, y, z, u) | (x = z_1) \wedge (y = z_2) \wedge (z = z_3) \wedge (u = z_4)\} \quad (15)$$

Definition of publisher (event) agents are

$$P1 = Ag2 = \{(m, n, o, p) | (m \leq z_1) \vee (n \leq z_2) \vee (o \leq z_3) \vee (p \leq z_4)\} \quad (16)$$

The mathematical descriptions of the content-based publish-subscribe agent structure were developed based on the system design described in Figure 5. As outlined in reference [41], for a given event, using a publish-subscribe algorithm can quickly eliminate large amounts of non-matching subscriptions and focus on a small subset of possibly matching subscriptions. The searching area of the subscriber agent ($Ag1$) is decreased from the large set of topics in each period to the smaller subset of $S1$. For example, $Ag1$ subscribes to the shared data among ' z_i ' values called $S1$ (15). The publisher agent ($Ag2$) can publish their service request area of $P1$ if they meet equation conditions (16).

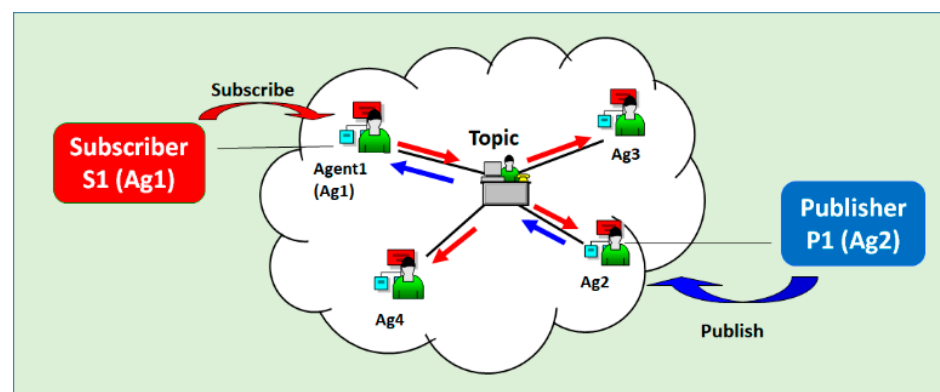


Figure 5. Application of Publish-Subscribe for Finding Global Optimization Ratios.

6. Developing an Integrated Platform for Evaluating the Publish-Subscribe Agent Model

An agent platform was designed using the Java Agent DEvelopment Framework (JADE, version 3.7, Telecom Italia Lab, Torino, Italy) compatible with the publish-subscribe architecture, and able to communicate with hardware and software applications such as MATLAB R2018a, The MathWorks, Inc., Natick, MA, USA. JADE provides a set of internationally recognized standards, defined by the Foundation for Intelligent Physical Agents (FIPA) [42]. FIPA, the standards organization for agents and multi-agent systems, supports common forms of messages exchanged by two or more agents. Three key mandatory roles in an agent platform: are identified as the agent management system (AMS), agent communication channel (ACC), and directory facilitator (DF) [43].

In this article, publish-subscribe messaging was adopted as an asynchronous communication method. A subscriber agent sends a request for some variables to a publisher agent. A message queue was defined for each agent where the JADE runtime places received messages from other agents. Whenever a message is received in the message queue, the subscriber agent is notified of any matching content while ignoring all non-matching contents. Subsequently, the publisher agent activates an action method to start communication.

A separate Java class was created for each agent to facilitate the message exchange with other agent classes and implement the publish-subscribe architecture. Figure 6 illustrates an agent-based system design, where eight converters are associated with four agents through the agent platform. Furthermore, DF and AMS, two facilitator agents from the FIPA standard, run in the main container. DF enables a directory displaying which agents are available on the platform, while AMS helps to create and destroy other agents, destroy containers, and stop the platform.

As a case study, a notional DC shipboard microgrid was developed and implemented in MATLAB. Although MATLAB is convenient for modeling optimization algorithms in microgrids, it cannot be used to implement the multithreaded agent architectures developed in other languages, such as JADE. There were constraints on the number of concurrent message exchanges between a microgrid modeled in MATLAB and the MAS platform developed in JADE. Therefore, MATLAB was coupled with a customized S-Function toolbox called multi-agent control for MATLAB/Simulink R2018a, The MathWorks, Inc., Natick, MA, USA named MACSim (Mendham & Clarke 2005a, University of York).

MACSim toolbox is described in an article [44] by Mendham and Clarke. They created it as a medium to facilitate the communication between an agent-based system designed in C/C++ or Java and Simulink models. As shown in Figure 7, MACSim has a client-server architecture, where the client part is developed in Simulink using an S-function, and the server is programmed in the JADE platform [45–47]. In this paper, the MACSim block was customized to integrate the microgrid hardware models and MAS software parts.

Within the MAS architecture, a coordinator agent (*AgCo*) was developed to enable communication between Simulink ports and the agent platform in JADE using a DF agent service. When the *AgCo* receives a data array from Simulink, it sends the data to the agents who have subscribed to this content. When these agents have processed the data, they send it back to the *AgCo*, to be transmitted to the microgrid model in Simulink. Routing tables are located in *AgCo* and are used and updated during the optimization processes.

Figure 8 shows the flow of optimization variables in a MAS system interfaced with the shipboard microgrid model. The agent model becomes activated upon receiving load values through a converter, and then the JADE platform creates four individual agents, *Ag1*, *Ag2*, *Ag3*, and *Ag4*, upon a trigger being received through *AgCo*. As shown in Figure 9, these agents use individual search trees to optimize ' z_1 ', ' z_2 ', ' z_3 ', and ' z_4 '. Since all of the agents run simultaneously, they obtain the optimization values of ' z_i ' from input values such as I_1 , I_2 , I_{PL} , I_3 , and I_4 that display the current values of zone 1, zone 2, pulsed load, bus-tie, and ESS in Figure 3, respectively. The four agents communicate and exchange data to optimize data and find the minimum ' z_i ' values using the publish-subscribe design. This agent calculates ' x ', ' y ', ' z ', ' u ' values and sends them back to Simulink through an *AgCo*.

After receiving the confirmation of data delivery, each agent terminates and finishes its life cycle. Finally, failure to complete the process by any agent for any reason could result in system instability. To keep the system stable, it should operate within a deadline for any search tree values. Since only search tree branches within constraint boundaries are searched, it is guaranteed that the system always operates within the constrained operating region. The result of this failure mode is simply that the system operation is sub-optimal unless all agents meet time deadlines for completion of the cost function search.

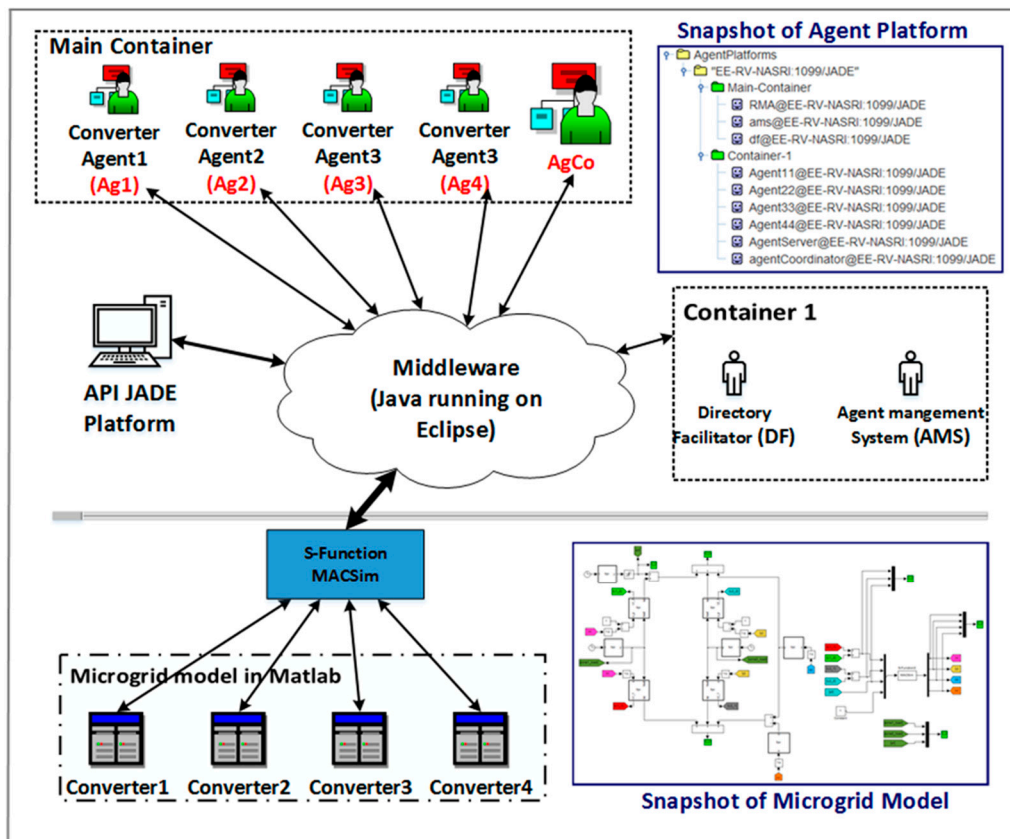


Figure 6. The integrated agent-based system, including a microgrid modeled in Simulink and MAS (Multi-agent System), programmed in JADE (Java Agent DEvelopment Framework), and communicating through the MACSim toolbox.

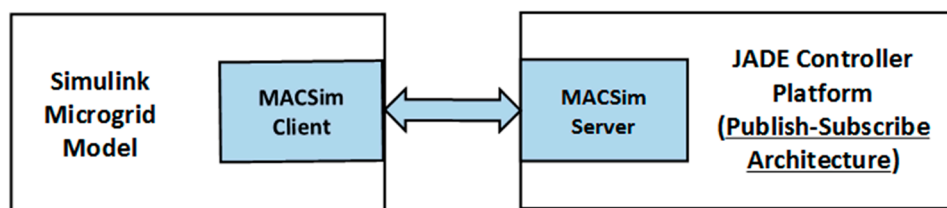


Figure 7. Structure of MACSim toolbox.

Figure 10 shows a picture of communication between the agents. All of the agents mentioned above register to the DF in the JADE platform to receive their unique agent number and find their local data provider. In the following, each agent does the required calculations before processing those received from other agents. Finally, the agents send the results to their subscribed agents.

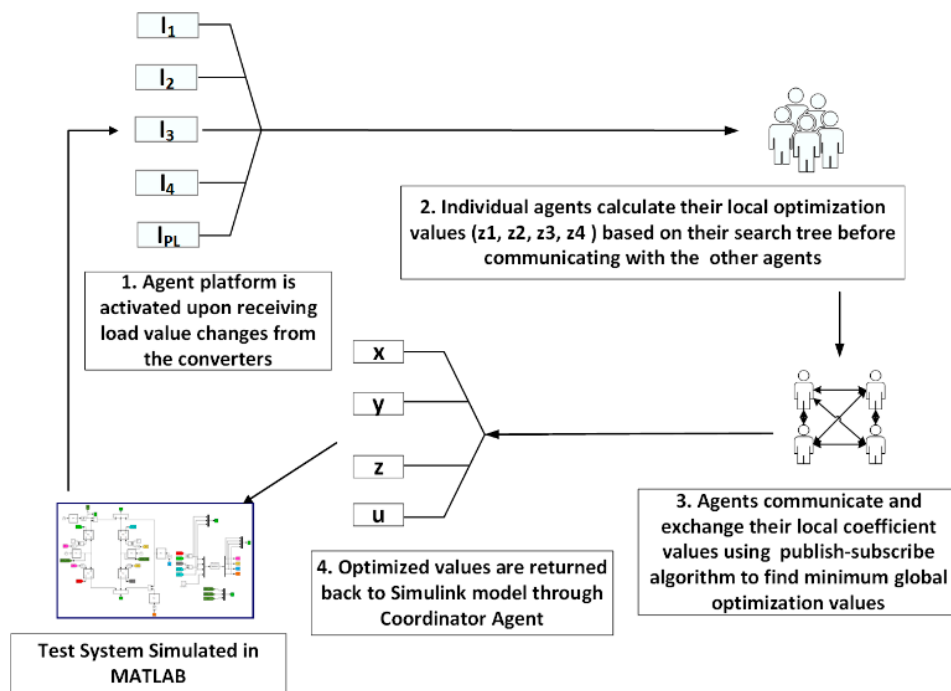


Figure 8. The flow of optimization variables in MAS (Multi-agent System).

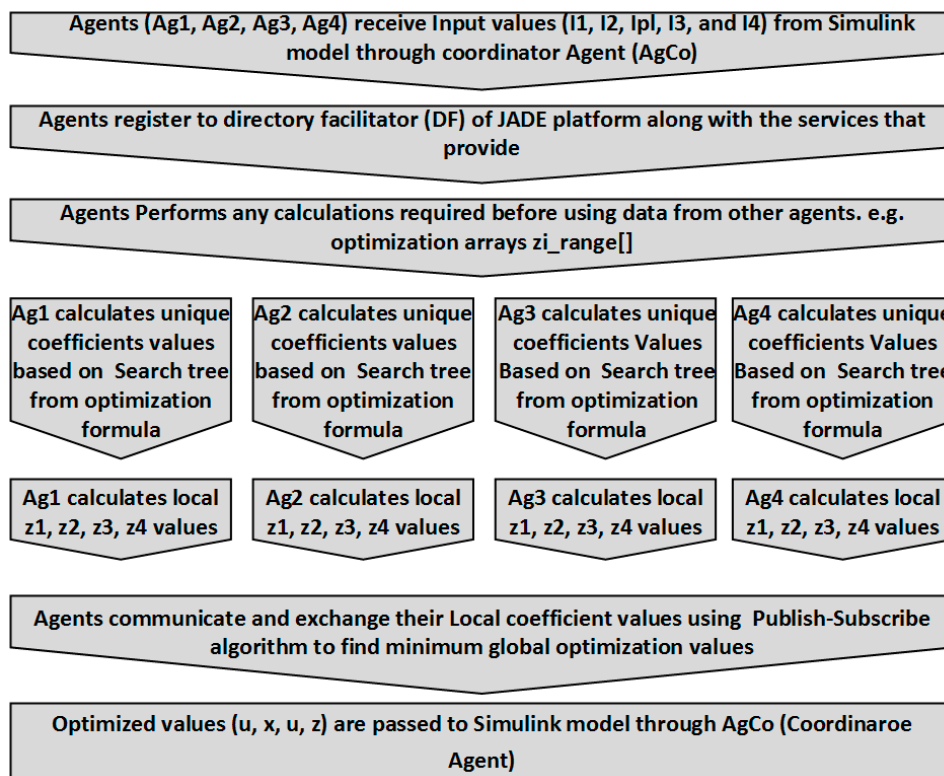


Figure 9. Concurrent optimization algorithm using four agents.

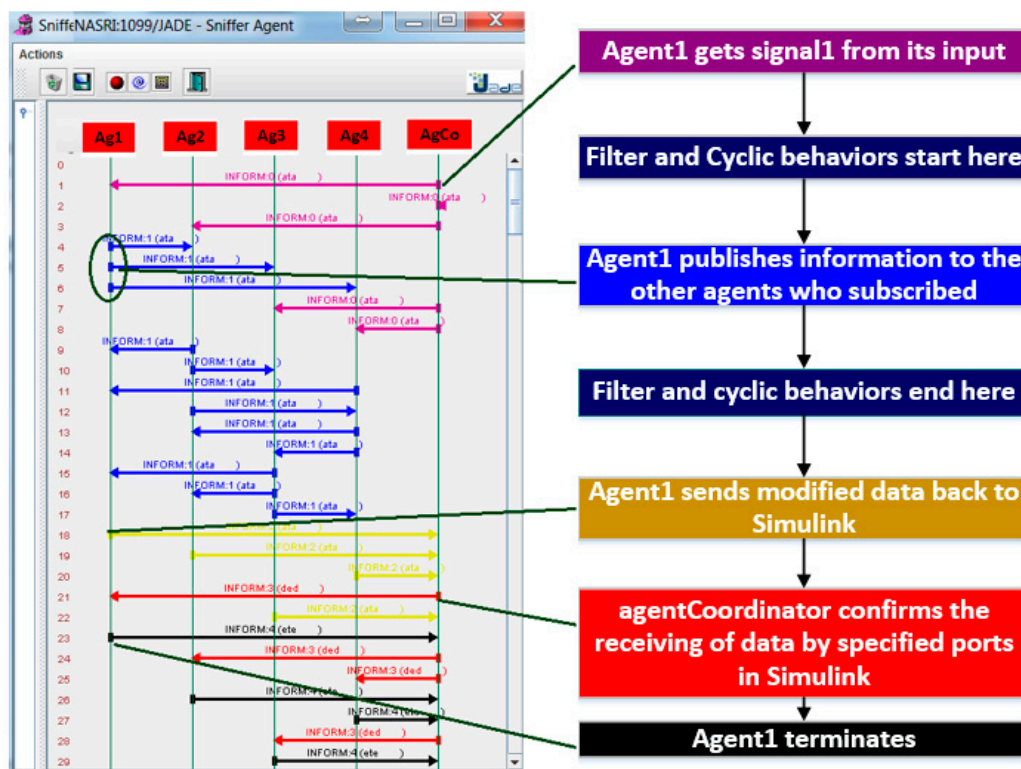


Figure 10. A snapshot of the JADE (Java Agent Development Framework) platform while running message exchange among agents.

7. Test Case Results

Figures 11 and 12 display how running the shipboard system model in MATLAB changes the converters’ load values and activates associated agents through the MACSim block. This trigger initializes the optimization process in the JADE platform. Variables ‘x’, ‘y’, ‘z’, and ‘u’ represent the zone 1 ratio, zone 2 ratio, bus-tie status, and storage system commands and become regulated upon receiving the value change for input loads. The zonal converter values of ‘x’ and ‘y’ may vary between 0 and 1, with a step size of 0.1, since their current flow is unidirectional. Variables ‘z’ and ‘u’ are associated with bidirectional currents and may have values in the range of -1 to $+1$, with a step size of 0.2 to keep a uniform ten steps for all control variables.

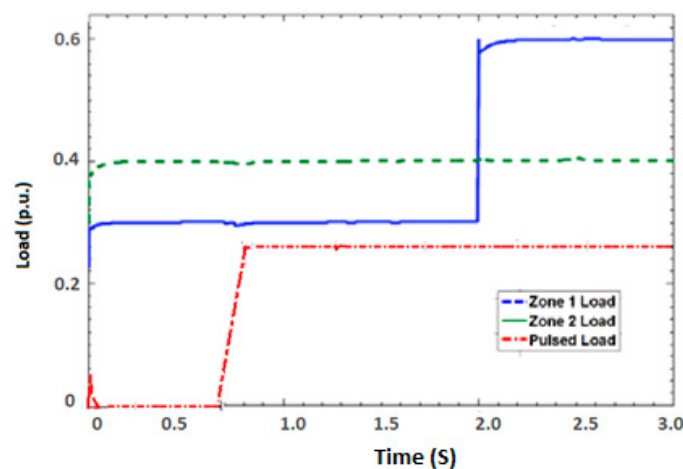


Figure 11. Load changes in the case study.

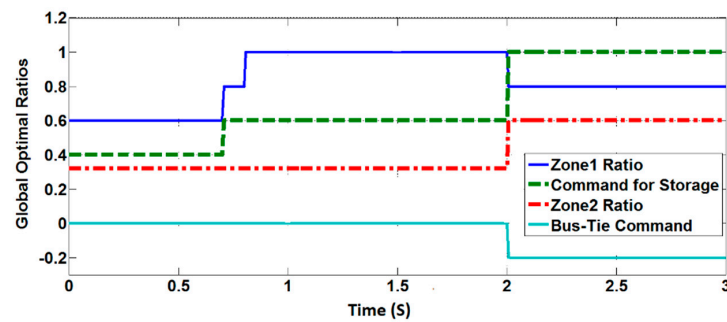


Figure 12. Converter operating points dictated by the agent-based controller.

At the time of 0.7 s, the fast varying load storage system is quickly charged, leading to the new loading conditions. Thus, the battery storage system doubles the supply and zone 1 sharing variable values to guarantee the minimum system loss. At a time of 2 s, the zone 1 load changes, causing the supervisory control to allocate the entire system.

Figure 13 shows that the sharing of load between zone 1 converters controlled by the zone 1 ratio is demonstrated by ‘x’. Figures 12 and 13 display that increasing the ‘x’ value causes changes in the corresponding load sharing. At around 0.8 s, ‘x’ becomes 1, leading one converter to carry 100 percent of load in zone 1, while the other converter has no load.

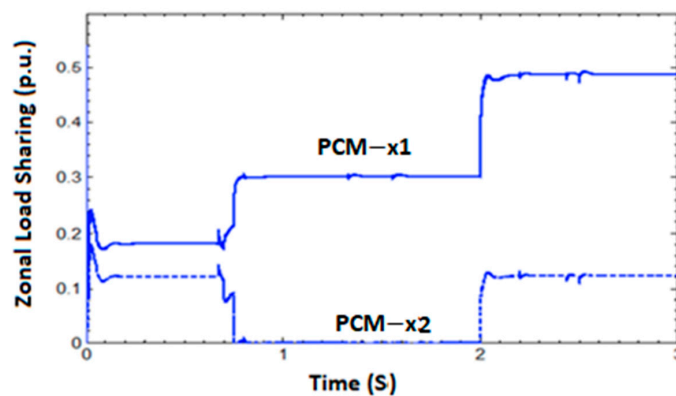


Figure 13. Load sharing in zone 1.

Figure 12 illustrates that the zone 2 sharing ratio indicated by ‘y’ stays constant between 0 and 2 s. As seen in Figure 14, the value of ‘y’ increased from 0.3 to 0.6 at the time of 2 s proves that the load sharing in zone 2 strictly followed these operating points. To validate the accuracy of simulation results, a centralized control method was implemented [39] with the same optimization algorithm and without utilizing agent technology, and the results exactly matched those shown in Figure 12.

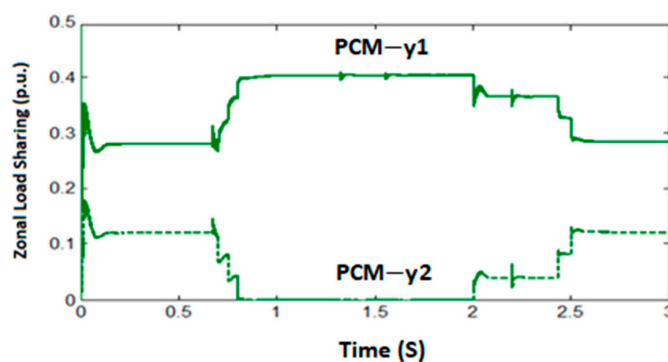


Figure 14. Load Sharing in zone 2.

Finally, to evaluate the proposed multi-agent system’s efficiency, system losses were computed for the setpoints determined by the multi-agent system compared to a loading-based sharing scheme, such as droop-based sharing [39], set in this case to the rating-based equivalent sharing shown in Figure 15.

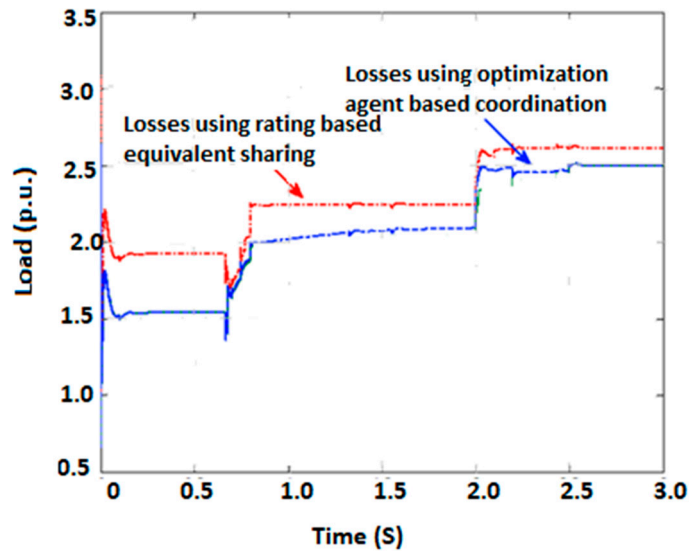


Figure 15. Losses for agent-based optimization versus equivalent sharing.

Figure 16 compares the number of message exchanges from the simulation with numerical analysis. There is a nearly constant difference between the number of predicted and actual message exchanges due to the message’s overhead caused by the AgCo agent. In some of the similar research studies, the number of message exchanges was calculated with decimal accuracy [29,45], but the results were rounded to the closest integer value. Accounting for this additional message overhead, the simulation results confirmed the numerical analysis outcomes calculated based on publish-subscribe MAS architecture. The graph shows that the number of message exchanges has a minimum value when the number of agents is limited to four.

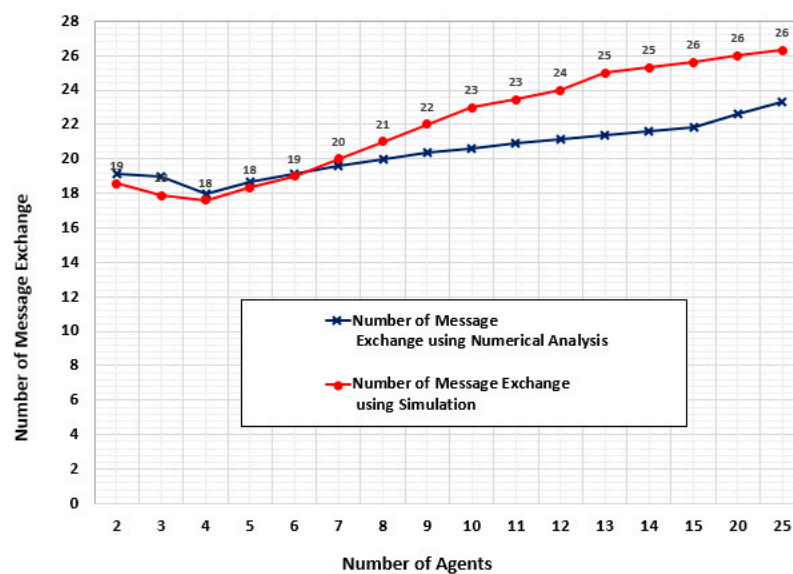


Figure 16. Comparison between the number of messages exchanged in the numerical analysis and simulation.

8. Conclusions

When optimizing multiple control parameters in an autonomous microgrid, controllable devices, such as power electronic converters, should be closely coordinated using an appropriate supervisory control technology. This paper presents a novel method for coordinating power converters in a microgrid using a publish-subscribe agent architecture.

Figure 17 displays a summary of the technologies used in the development of the proposed coordinated control systems.

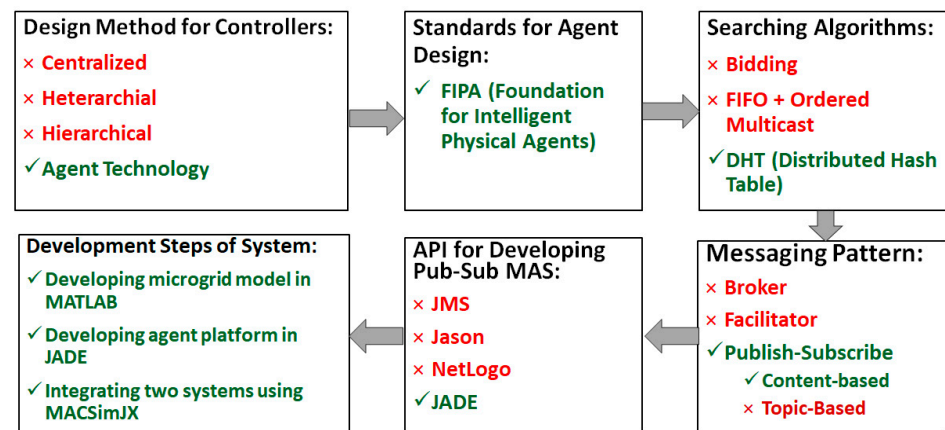


Figure 17. Technologies used in the development of the coordinated control system.

The content-based publish-subscribe agent technology was modeled using the JADE platform and implemented for a notional shipboard microgrid. The content was defined for each publisher and subscriber based on the required data for running an optimization algorithm within each agent.

In the case study, the results of applying the publish-subscribe agent-based system for coordination of power converters were presented and analyzed. The publish-subscribe agent-based method presented limits message exchange so that deterministic system control is achievable. Thus, it can be applied for coordination among power electronic converters in a microgrid with tens of converters. The case study presented here demonstrates such coordination by adjusting converter operating points within a bounded time frame, based on an optimization cost function triggered by system load flow changes.

Author Contributions: Conceptualization, H.L.G.III; methodology, M.N., H.L.G.III; validation, M.N., H.L.G.III, M.M.; formal analysis, M.N., H.L.G.III, M.M.; investigation, M.N., H.L.G.III; resources, M.N.; data curation, M.N.; writing—original draft preparation, M.N.; writing—review and editing, M.N., H.L.G.III, M.M.; visualization, M.N., H.L.G.III; supervision, H.L.G.III, M.M.; project administration, H.L.G.III. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets may be available from the corresponding author on a reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bakken, J.D. *Smart Grids: Clouds, Communications, Open Source, and Automation*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2014.
2. Dehnavi, G.; Ginn, H.L. Distributed Load Sharing Among Converters in an Autonomous Microgrid Including PV and Wind Power Units. *IEEE Trans. Smart Grid* **2018**, *10*, 4289–4298. [[CrossRef](#)]
3. Olivares, D.E.; Mehrizi-Sani, A.; Etemadi, A.H.; Cañizares, C.A.; Iravani, R.; Kazerani, M.; Hajimiragha, A.H.; Gomis-Bellmunt, O.; Saeedifard, M.; Palma-Behnke, R.; et al. Trends in Microgrid Control. *IEEE Trans. Smart Grids* **2014**, *5*, 1905–1919. [[CrossRef](#)]
4. Manbachi, M.; Nasri, M.; Shahabi, B.; Farhangi, H.; Palizban, A.; Arzanpour, S.; Moallem, M.; Lee, D.C. Real-Time Adaptive VVO/CVR Topology Using Multi-Agent System and IEC 61850-Based Communication Protocol. *IEEE Trans. Sustain. Energy* **2014**, *5*, 587–597. [[CrossRef](#)]

5. Manbachi, M.; Sadu, A.; Farhangi, H.; Monti, A.; Palizban, A.; Ponci, F.; Arzanpour, S. Real-Time Co-Simulation Platform for Smart Grid Volt-VAR Optimization Using IEC 61850. *IEEE Trans. Ind. Inform.* **2017**, *17*, 1392–1402. [[CrossRef](#)]
6. Ginn, H.L.; Ponci, F.; Monti, A. Multi-agent control of PEBB based power electronic systems. In Proceedings of the 2011 IEEE Trondheim in PowerTech, Trondheim, Norway, 19–23 June 2011; pp. 19–23.
7. Dehnavi, G.; Ginn, H.L. Concurrent distributed control of all power components in an autonomous microgrid. In Proceedings of the 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, USA, 22–26 July 2012; pp. 1–8. [[CrossRef](#)]
8. Dehnavi, G.; Ginn, H.L. Load sharing among converters in an autonomous microgrid in presence of wind and PV sources. In Proceedings of the 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 24–27 February 2013; pp. 1–6.
9. Momoh, A.; Meliopoulos, S.; Saint, R. *Centralized and Distributed Generated Power Systems-A Comparison Approach*; Power Systems Engineering Research Center, Arizona State University: Tempe, Arizona, 2012.
10. Foruzan, E.; Algrain, M.C.; Asgarpoor, S. Decentralized Controller Design for Microgrids in Islanded and Grid-Connected Modes. In Proceedings of the IEEE International Conference on Electro Information Technology, Lincoln, NE, USA, 14–17 May 2017.
11. Hossain, E.; Kbalci, E.; Bayindir, R.; Perez, R. Microgrid testbeds around the world: State of art. *Energy Convers. Manag.* **2014**, *86*, 132–153. [[CrossRef](#)]
12. Petrecca, G. *Energy Conversion and Management*; Springer: Berlin, Germany, 2014; Volume 86, p. 326.
13. Dondossola, G.; Deconinck, G.; Garrone, F.; Beitollahi, H. Testbeds for Assessing Critical Scenarios in Power Control Systems. In *Critical Information Infrastructure Security, Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5508.
14. Sharma, D.D.; Singh, S.N.; Lin, J.; Foruzan, E. Agent-based distributed control scheme for distributed energy storages based on forecast uncertainties. In Proceedings of the 2017 IEEE Power & Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017; pp. 1–5. [[CrossRef](#)]
15. Nasri, M.; Farhangi, H.; Palizban, A.; Moallem, M. Multi-agent control system for real-time adaptive VVO/CVR in Smart Substation. In Proceedings of the 2012 IEEE Electrical Power and Energy Conference, San Diego, CA, USA, 22–26 July 2012; pp. 1–7. [[CrossRef](#)]
16. Dwyer, A.; Nielsen, R.; Markushevich, N.; Stangl, J. Load to voltage dependency tests at B.C. Hydro. *IEEE Trans. Power Syst.* **1995**, *10*, 709–715. [[CrossRef](#)]
17. Bellifemine, F.; Poggi, A.; Rimassa, G. JADE—A FIPA-compliant agent framework. In Proceedings of the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM), London, UK, 19–21 April 1999; pp. 97–108.
18. Bellifemine, F.; Caire, G.; Greenwood, D. *Developing Multi-Agent Systems with JADE*; Textbook; Wiley: Hoboken, NJ, USA, 2007.
19. Sycara, K.; Pannu, A.; Williamson, M.; Zeng, D.; Decker, K. Distributed intelligent agents. *IEEE Expert* **1996**, *11*, 36–46. [[CrossRef](#)]
20. Nasri, M. RCPC: A Multi-Agent System for Coordinated Control of Power Electronic Converters in Microgrids. Ph.D. Thesis, School of Mechatronics System Engineering, Simon Fraser University, Surrey, BC, Canada, 2016.
21. McArthur, S.D.J.; Davidson, E.M.; Catterson, V.M.; Dimeas, A.L.; Hatziargyriou, N.D.; Ponci, F.; Funabashi, T. Multi-Agent Systems for Power Engineering Part II. *IEEE Trans. Power Syst.* **2007**, *22*, 1743–1759. [[CrossRef](#)]
22. Saleem, A.; Honeth, N.; Nordstrom, L. A case study of multi-agent interoperability in IEC 61850 environments. In Proceedings of the 2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe), Gothenburg, Sweden, 11–13 October 2010; pp. 1–8.
23. Harp, S.; Brignone, S.; Wollenberg, B.; Samad, T. SEPIA. A simulator for electric power industry agents. *IEEE Control. Syst.* **2000**, *20*, 53–69. [[CrossRef](#)]
24. Sharma, D.D.; Singh, S.N.; Lin, J.; Foruzan, E. Agent-Based Distributed Control Schemes for Distributed Energy Storage Systems Under Cyber Attacks. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2016**, *7*, 307–318. [[CrossRef](#)]
25. Ahmad, I.; Kazmi, J.H.; Shahzad, M.; Palensky, P.; Gawlik, W. Co-simulation framework based on power system, AI and communication tools for evaluating smart grid applications. In Proceedings of the IEEE PES Innovative Smart Grid Technologies, Asian Conference, Bangkok, Thailand, 3–6 November 2015.
26. Green, S.; Hurst, L.; Nangle, B.; Cunningham, P. *Software Agents: A Review*; Technical Report TCD-CS-1997-06, Technical Report of Trinity College; University of Dublin: Dublin, Ireland, 1997.
27. Tolbert, L.M.; Qi, H.; Peng, F.Z. Scalable multi-agent system for real-time electric power management. In Proceedings of the 2001 Power Engineering Society Summer Meeting, Conference Proceedings (Cat. No.01CH37262), Vancouver, BC, Canada, 15–19 July 2001; Volume 3, pp. 1676–1679.
28. Tarkoma, S. *Publish/Subscribe Systems, Design and Principals, Chapters 7 & 8, Textbook*; Wiley Press: Hoboken, NJ, USA, 2012.
29. Sun, Y.; Qiao, X.; Tan, W.; Cheng, B.; Shi, R.; Chen, J. A low-delay, light weight publish/subscribe architecture for delay-sensitive IOT services. In Proceedings of the IEEE Proceedings International Conference on Web Services (ICWS), Santa Clara, CA, USA, 28 June–3 July 2013; pp. 179–186.
30. Zhu, Y.; Hu, Y. Ferry: A P2P-Based Architecture for Content-Based Publish/Subscribe Services. *TPDS IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 672–685. [[CrossRef](#)]
31. Castro, M.; Druschel, P.; Kermarrec, A.-M.; Rowstron, A. Scribe: A large-scale and decentralized application-level multicast infrastructure. *IEEE J. Sel. Areas Commun.* **2002**, *20*, 1489–1499. [[CrossRef](#)]

32. Gupta, A.; Sahin, O.D.; Agrawal, D.; Abbadi, A.E. Meghdoot: Content-based publish/subscribe over P2P networks. In Proceedings of the International Middleware Conference, Toronto, ON, Canada, 18–20 October 2004; pp. 254–273.
33. Nasri, M.; Hossain, R.; Ginn, H.L.; Moallem, M. Distributed control of converters in a DC microgrid using agent technology. In Proceedings of the 2016 Clemson University Power Systems Conference (PSC), Clemson, SC, USA, 8–11 March 2016.
34. Rowstron, A.; Druschel, P. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In Proceedings of the 8th IFIP/ACM Int. Conf. Distributed Systems Platforms (Middleware 2001), Heidelberg, Germany, 12–16 November 2001; pp. 329–3503.
35. Ratnasamy, S.; Francis, P.; Handley, M.; Karp, R.; Shenker, S. A scalable content-addressable network. *ACM SIGCOMM Comput. Commun. Rev.* **2001**, *31*, 161–172. [[CrossRef](#)]
36. D’Agostino, F.; Kaza, D.; Martelli, M.; Schiapparelli, G.-P.; Silvestro, F.; Soldano, C. Development of a Multiphysics Real-Time Simulator for Model-Based Design of a DC Shipboard Microgrid. *Energies* **2020**, *13*, 3580. [[CrossRef](#)]
37. Doerry, N. Naval Power Systems. *IEEE Electr. Mag.* **2015**, *3*, 12–21. [[CrossRef](#)]
38. Pang, S.; Farrell, J.; Du, J.; Barth, M. Battery state-of-charge estimation. In Proceedings of the 2001 American Control Conference, Arlington, VA, USA, 25–27 June 2001.
39. Hossain, R.; Ginn, H.L. Real-Time Distributed Coordination of Power Electronic Converters in a DC Shipboard Distribution System. *IEEE Trans. Energy Convers.* **2017**, *32*, 770–778. [[CrossRef](#)]
40. Nasri, M.; Hossain, R.; Ginn, H.L.; Moallem, M. Agent-based real-time coordination of power converters in a DC shipboard power system. In Proceedings of the 2015 IEEE Electric Ship Technologies Symposium (ESTS), Old Town Alexandria, VA, USA, 21–24 June 2015.
41. Aguilera, M.K.; Strom, R.E.; Sturman, D.C.; Astley, M.; Chandra, T.D. Matching events in a content-based subscription system. In Proceedings of the ACM Symposium on Principles of Distributed Computing, Atlanta, GA, USA, 29 April–6 May 1999; pp. 53–61.
42. The Foundation for Intelligent Physical Agents, Specifications. 8 June 2005. Available online: <http://www.fipa.org/specs/fipa00023/index.html> (accessed on 25 April 2021).
43. FIPA 98 Specification, Agent Security Management. 23 October 1998. Part 10, Version 1.0. Available online: <http://www.fipa.org/specs/fipa00020/OC00020A.html> (accessed on 25 April 2021).
44. Mendham, P.; Clarke, T. Clarke, Macsim: A Simulink Enabled Environment for Multi-Agent System Simulation. In Proceedings of the 16th IFAC World Congress, Prague, Czech Republic, 3–8 July 2005.
45. Robinson, C.R.; Mendham, P.; Clarke, T. MACSimJX: A tool for enabling agent modeling with Simulink using JADE. *J. Phys. Agents* **2005**, *4*, 1–7. [[CrossRef](#)]
46. Raju, L.; Milton, R.S.; Mahadevan, S. Multiagent Systems Based Modeling and Implementation of Dynamic Energy Management of Smart Microgrid Using MACSimJX. *Sci. World J.* **2016**, *2016*, 9858101. [[CrossRef](#)]
47. Raju, L.; Milton, R.S.; Mahadevan, S. Application of Multi-Agent Systems in Automation of Distributed Energy Management in Micro-grid using MACSimJX. *Intell. Autom. Soft Comput.* **2018**, *24*, 483–491. [[CrossRef](#)]