

Distributed Negotiation in Future Power Networks: Rapid Prototyping Using Multi-Agent System

M. Chen, M. H. Syed, E. Guillo Sansano, S. D. J. McArthur, G. M. Burt and I. Kockar

Department of Electronic and Electrical Engineering
University of Strathclyde
Glasgow, UK
minjiang.chen@strath.ac.uk

Abstract—Technologies like multi-agent system (MAS) have the capability to deal with future power grid requirements such as frequency management and voltage control under a flexible, intelligent and active feature. Based on web of cells (WoC) architecture proposed by European Liaison on Electricity Committed Towards longer-term Research Activity Integrated Research Programme (ELECTRA IRP), a distributed MAS with distributed negotiation ability for future distributed control (including frequency management and voltage control) is proposed. Each cell is designed as an intelligent agent and is investigated in case studies with constraints, where each agent can only communicate with its neighbouring agents. The interaction logic among agents is according to the distributed negotiation algorithm under consideration by the authors. Simulation results indicate that the WoC architecture could negotiate resources in a distributed manner and achieve successful exchange of resources by coordinating distributed agents. Moreover, the prototype reported in this paper can be extended further for future grids' distributed control regimes. The option of MAS to be exploited for the support of the development and integration of novel power system concepts is explored.

Index Terms—Multi-agent system, negotiation algorithm, rapid prototyping, distributed control, agent communication.

I. INTRODUCTION

The increasing integration of distributed energy resources (DERs) implies dramatic changes in the power grid [1], [2]. The renewable energy sources (RESs) are major providers of DER connected to power distribution grids, such as solar, wind, biomass and hydro. As the nature of a RES is intermittent and stochastic, it stresses the whole power system in terms of voltage and frequency fluctuations [3], [4]. This can lead to occasions where voltage or frequency is outside of statutory or operational limits. In order to solve such unwanted effects in the power grid, intelligent systems techniques play an important role in network management and control. However, today's information and control technologies primarily control power grids in a centralized, hierarchical-oriented structure [5], which has limitations in terms of scalability, computational complexity, and communication [6].

Future power grids will transition from a centralized control structure to more distributed control architectures with varying distributed decision-making functionalities. The benefits of a distributed architecture are the ability for the

electric power grid infrastructures to have local intelligence with local autonomy, self-management and self-healing operation. At the same time, individual prosumers and community energy initiatives can be empowered. These capabilities offer greater flexibility, extensibility, and scalability.

Hence, new software architectures have to enable intelligent grid operation with highly decentralized features. Multi-Agent Systems (MAS) is one of the most relevant intelligent technologies that allows the development of a distributed control system. A MAS system comprises of several intelligent agents [7]. Each agent has its own knowledge, capability to sense and act in the environment, and can autonomously plan its own activities. Agents can communicate with other agents and cooperate together to achieve the global system's objectives within a distributed control structure. In addition, MAS technology can provide fast response to the network changes and supports reconfiguration without affecting the other components of the system [8].

MAS technology has already been modelled and designed in several simulation studies and pilot projects in industrial domains [9], [10]. Additionally, there are many reports about successful examples of MAS rapid prototyping in literature [11]- [13]. Rapid prototyping allows engineers to validate the essential features of the proposed system by means of being able to identify and quickly rectify implementation issues at an early stage in development. In this paper, we present an example of rapid prototyping based on MAS that runs as a distributed platform by means of a distributed algorithm with agent cooperation. To do so, the web of cells (WoC) concept proposed by the ELECTRA IRP project [14] is used. Each cell is designed as an agent that could contain any combination of generation and load capacity. A control algorithm can be integrated into a cell agent as an agent function so that the agent will follow the logic and exhibit certain behaviours. Each agent can only interact with their neighbours to exchange information or coordinate to achieve system goals. The proposed MAS system, consisting of several cell agents, will be tested in case studies to prove the concept of the agent based integrated control algorithm under various conditions. Paper provides how MAS could have utilized for rapid prototyping of future solutions for the purpose of testing that will allow them to be deployed in large scale distribution system.

In this paper, Section II describes a distributed architecture approach, the WoC concept, and how a distributed control algorithm could be used in such architecture. The MAS simulation platform and agent prototyping details are described in Section III. To demonstrate the capability of the proposed MAS system, two case studies are presented in Section IV. Finally, the paper is concluded and key future work is outlined in Section V.

II. FUTURE DISTRIBUTED POWER SYSTEMS

The change in generation resources, from large-scale power plants to distributed energy resources, expected in future power systems is also foreseen to drive a move from centralized to more distributed power systems. An example of such an approach has been identified in the ELECTRA IRP WoC concept.

A. Web of Cells Concept

The WoC is a distributed architecture concept aiming to increase the reliability of future power systems by achieving distributed control operation of autonomous regions within the power system, known as cells. A cell is defined as “a group of interconnected loads, distributed energy resources and storage units within well-defined grid boundaries corresponding to a physical portion of the grid and to a confined geographical area” [14]. Each cell in the network has the same level of authority and there is no hierarchical control over cells by a superior entity (i.e. system operator). Hence, distributed negotiation and coordination between cells is a means of delivering secure system-wide operation.

B. Distributed Negotiation Algorithm

With such a distributed approach as the WoC, negotiation between cells will play an important role for the successful operation of the architecture. Negotiation algorithms based on price signals are widely applied in power markets [15], [16]. In contrast, our key objective is to realise distributed control of frequency and voltage within the WoC architecture incorporating the negotiation algorithm. The negotiation between cells can have different objectives and it will mainly be a negotiation between cells with insufficient resources (deficit cells) and cells with surplus resource (excess cells). The resource could be anything required by one cell that a neighbouring one would be able to provide. This negotiation can involve more than two cells that can help to provide the total amount of resources required from a cell in deficit in a distributed manner.

The distributed negotiation algorithm that will be used for the design of the multi-agent system is presented in Fig.1. When a cell is aware of a deficient status of a resource within its cell it will send the deficient quantity of resource to its neighbours (cells electrically interconnected). The neighbouring cells will assess the state of their resources and if they do have available resources a quantity versus price curve will be sent. At this point the deficient cell will have received the curves of the neighbours with resource surplus and it will evaluate them in order to get the cheapest combination. Once the bid (or combination to different neighbours) is sent the

Distributed Negotiation Algorithm

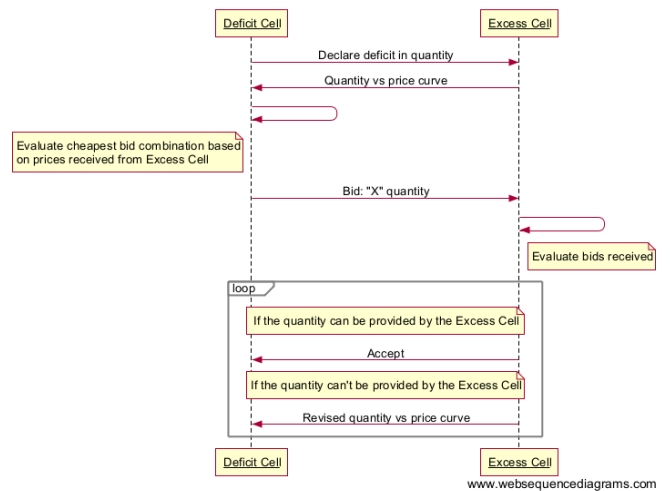


Fig.1 Distributed negotiation algorithm sequence diagram

excess cells will evaluate the bids received (there can be more than one cell that may be requiring assistance). If the quantity can be provided, the excess cell will accept the bid. If the quantity can't be provided because different cells are bidding and the total combined bid is larger than the available amount, the excess cell will send a revised price curve to the deficient cells. This action will be continued until an agreement is achieved. It is assumed that the total amount of resources required will be smaller or equal to the excess of resources.

III. MULTI-AGENT SYSTEM DESIGN WITH DISTRIBUTED NEGOTIATION FUNCTIONALITY

In this section, the MAS platform applied in this paper will be introduced and MAS prototyping will be detailed along with the agent modelling.

A. Agent Simulation Platform

The proposed distributed negotiation algorithm is implemented using the Simulation of Agent Societies 2 (Presage2) framework [17], [18]. Presage is a JAVA based programming environment that provides improved autonomy and agent communication capability. Presage2 contains abstract classes and interfaces for the user to extend. It allows for trialling of MAS prototypes and designing novel advanced applications in electric power systems. In addition, Presage2 offers flexibility to design self-organising systems that meet the requirements of applications in power grid management.

The MAS simulation environment in Presage2 is shown in Fig.2. In the MAS platform environment, agents collaborate

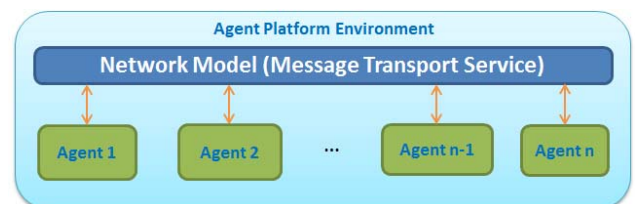


Fig.2 MAS simulation environment in Presage2

with each other through suitable agent communication in the network model. The Network Model is the central switchboard for messaging in the platform, rather than letting the agents communicate with each other directly. Users can add constraints to allow the blocking of messages or modify the messages before delivery. Presage2 has defined its own agent communication language. The message format in Presage2 is:

Message (Performative, Sender, Receiver, Time, Content)

The Message Performative indicates the intention behind the message, such as “require”, “query” and “inform”. The Message Sender and Receiver refer to the agents that are sending and receiving the message respectively. Time of the message is the time at which the message was sent. Message Content is the information that is communicated between the agents. Hence, Presage2 provides the essential infrastructure for agent design, simulation and communications.

B. Agent Modelling

Each agent has simulation states to store required information and it has a message box to receive and send messages. During MAS simulation, each agent uses its behaviours and rules to govern its decision making and to cooperate with other agents. The details of agent model show below will present the cell agent architecture for the resource negotiation. It illustrates the states inside of each agent, the messages that the agent can receive or send, the main functions of the cell agent and how each agent will communicate to reach an agreement. The agent model associated with Presage2 for negotiation algorithm is illustrated in Fig.3. Based on the negotiation algorithm described in Section II, each cell is operating as an agent. The MAS consists of several cell agents that communicate and coordinate with each other to provide amount of resources from a cell in deficit. As a result, each cell agent could present the different behaviours according to negotiation algorithm.

a) *Agent Setup*: Each agent will be initialised at the beginning of the simulation to setup which cell agent is its neighbour. Hence, each cell agent will know which cell agent it can communicate with.

b) *Agent States*: A cell agent has its own resource status such as required amount or resource excess for the next time period, and an associated price curve associated with it. As a result, the agent states include resource status and price curve. Agent states are not only used to store its own information, but also to record other agents’ resources and price curves. Agent states will be updated after each cell agent reaches an agreement with other agents.

c) *Message Inbox/Outbox*: Each cell agent will send its own status to neighbouring agents by messaging, including resource status and price curve for an excess of resource. Hence, each agent will also receive resource status and price curve messages from neighbouring agents. Moreover, a cell agent will send request messages to neighbouring agents with the required resource amount if in deficit for the next time period. Once a request message has been received agents will decide whether to accept or not. If a cell agent can accept requests, it will send accept messages to requesting agents. The neighbouring agents will reply with a message as a confirmation. If it cannot fulfil the amount of resource requested, the cell agent will send an increased price curve message to them.

d) *Agent Functions*: The distributed negotiation algorithm will be integrated into each agent as one of the functions so that it will guide the agent to perform specific actions under differential conditions. Based on the algorithm, a cell agent could be implemented to represent two different behaviours. A cell agent will operate as a deficit agent after checking the amount of resources within its cell still lacking resources for the next time period. On the other hand, a cell that can provide resources to its neighbour agents will be an excess agent. As a result, deficit agents and excess agents will take different actions. For instance, a deficit agent will calculate minimum cost of resources according to the price curve from the excess agent. An excess agent, on the other hand, will check if a received request can be fulfilled or not due to other deficit agent requesting at the same time.

e) *Agent Communicative Actions*: The agent communicative action is triggered when a cell agent needs to send messages to other agents, such as sending request resource amount message and sending a modified price curve of excess resources. However, each cell agent can only communicate with its neighbour cell agents, which means a cell agent is electrically interconnected with its neighbour cell agents.

IV. CASE STUDY SIMULATION

In this section, the prototype of the distributed algorithm developed using MAS will be tested for the purpose of reserve procurement in a WoC architecture by means of two case studies. The first case study comprises three cells, while the second has four cells. Both case studies will be simulated for two different sets of price curves. The price curves in general are based on the resources that are present within the cell and the formation of the price curves is out of scope of this paper. The price curve is increased by 0.1 £/MW each time when the required amount of resource cannot be fulfilled by the excess

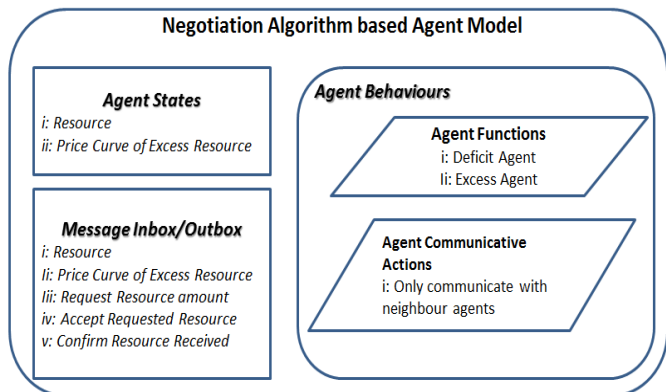


Fig.3 Agent model for negotiation algorithm

agent - this action is dependent upon the cell operator. In this study, it is assumed that there are always sufficient reserves available within the network, i.e. there will always be excess reserves available somewhere to cover any reserve requirements of cells in deficit. The message sent between the cells includes resource amount and price associated with it in the form of a curve.

A. Three Cell Network Study

The three cell interconnection diagram is shown in Fig.4. Cell 1 and Cell 3 are connected to Cell 2. In this case, Cell 1 and Cell 3 have excess resources of 100 MW and 150 MW respectively. Cell 2, in contrast, requires 150 MW. In order to meet the required reserve amount, Cell 2 requires 150 MW from Cell 1 and/or Cell 3. The portion of reserve procured from each excess agent will depend on their prices and availability, and also on the minimising cost function that the deficit agent has. For simulation two different price curves have been chosen that would represent two different times of the day or different composition of resources available during the two periods.

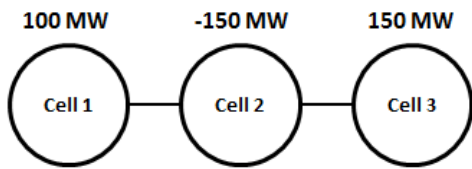


Fig.4 Three cell interconnection diagram

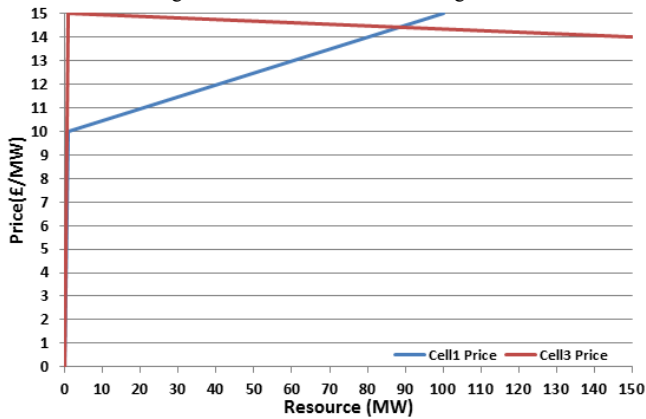


Fig.5 Three Cell Network Study: Case A price curves for cell 1 and 3

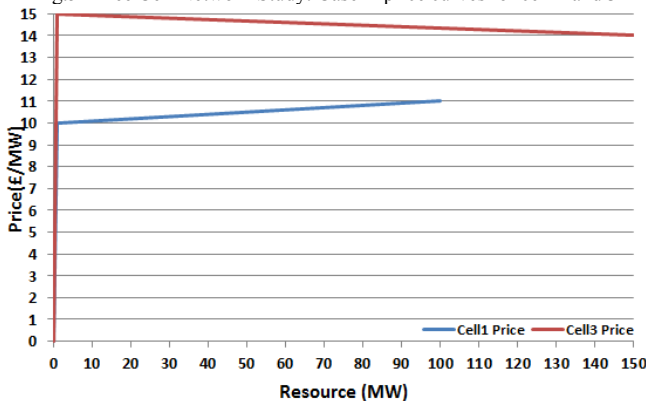


Fig.6 Three Cell Network Study: Case B price curves for cell 1 and 3

1) *Case A*: The price curves for Cell 1 and Cell 3 in this case are depicted in Fig.5. After all cell agents send their amount of reserves and price curve of reserves to neighbour agents for the next time period, Cell 2 will try to minimise its cost by calculating the required reserves from Cell 1 and Cell 3 based on each price curve. The solution determined by Cell 2 is to request 35 MW from Cell 1 and 115 MW from Cell 3. It is obvious that Cell 1 and Cell 3 accept the request from Cell 2 in this case without increasing price as no other cells are bidding for their reserves and they can provide the amount that has been asked for.

2) *Case B*: The price curves for Cell 1 and Cell 3 in this case are shown in Fig.6. Cell 2 follows the same procedure to calculate the required amount of reserve to be procured from Cell 1 and Cell 3 with their new set of price curves. As Cell 1's price of reserve in this case is lower than in price curve A and Cell 3 keeps the same price curve, Cell 2 requests 100 MW from Cell 1 and 50 MW from Cell 3. Again, Cell 1 and Cell 3 both accept the request from Cell 2 without increasing price due to the lack of other neighbours and having sufficient reserves.

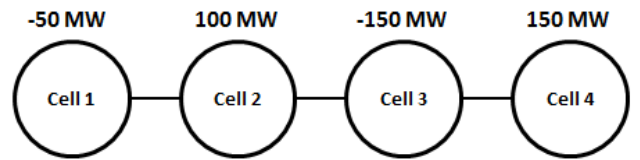


Fig.7 Four cell interconnection diagram

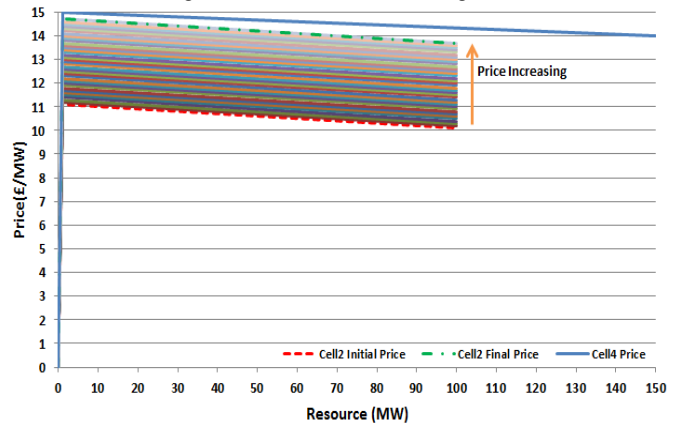


Fig.8 Four Cell Network Study: Case A price curves for cell 2 and cell 4

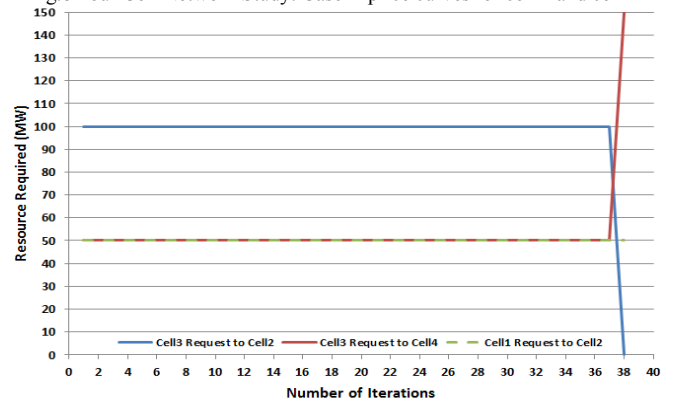


Fig.9 Four Cell Network Study: Case A iteration results for cell 1 and cell 3

B. Four Cell Network Study

An extra cell is added in this case study. The four cell diagram is shown in Fig.7. Cell 1 and Cell 3 both lack 50 MW and 150 MW reserves respectively. Cell 2 and Cell 4 both have excess reserve of 100 MW and 150 MW respectively. Cell agents follow the same procedure of sending their own reserve information and price curves of excess reserves to neighbour agents. However, interactions between each agent are greater here due to different reserve conditions and price curves. Simulations of two different price curves are presented below.

1) *Case A:* The price curves for Cell 2 and Cell 4 can be found in Fig.8. After all cell agents send their amount of reserves and price curve of reserves to neighbour agents for the next time period, Cell 1 and Cell 3 start to request reserve from

Cell 2 and Cell 4. Since Cell 2 is the only neighbour of Cell 1, Cell 1 always requires 50 MW from Cell 2 no matter the price of the reserve. Cell 3, however, needs to balance reserve between Cell 2 and Cell 4 to meet minimum cost. The first request for Cell 3 is to require 100 MW from Cell 2 and 50 MW from Cell 4. As a result, Cell 2 increased its price and sent the increased price to Cell 1 and Cell 3 due to 150 MW exceeding its own reserve capacity. Cell 4 is the only neighbour of Cell 3 and it always accepts the request from Cell 3 so that Cell 4 keeps the same price curve. The cell agents' communications is portrayed in Fig.10, which illustrates "Agents Setup", "Agents Send States", "Agents Negotiating" and "Agents Reach Agreement". From the agent communication diagram, Cell 1 and Cell 3 continued to

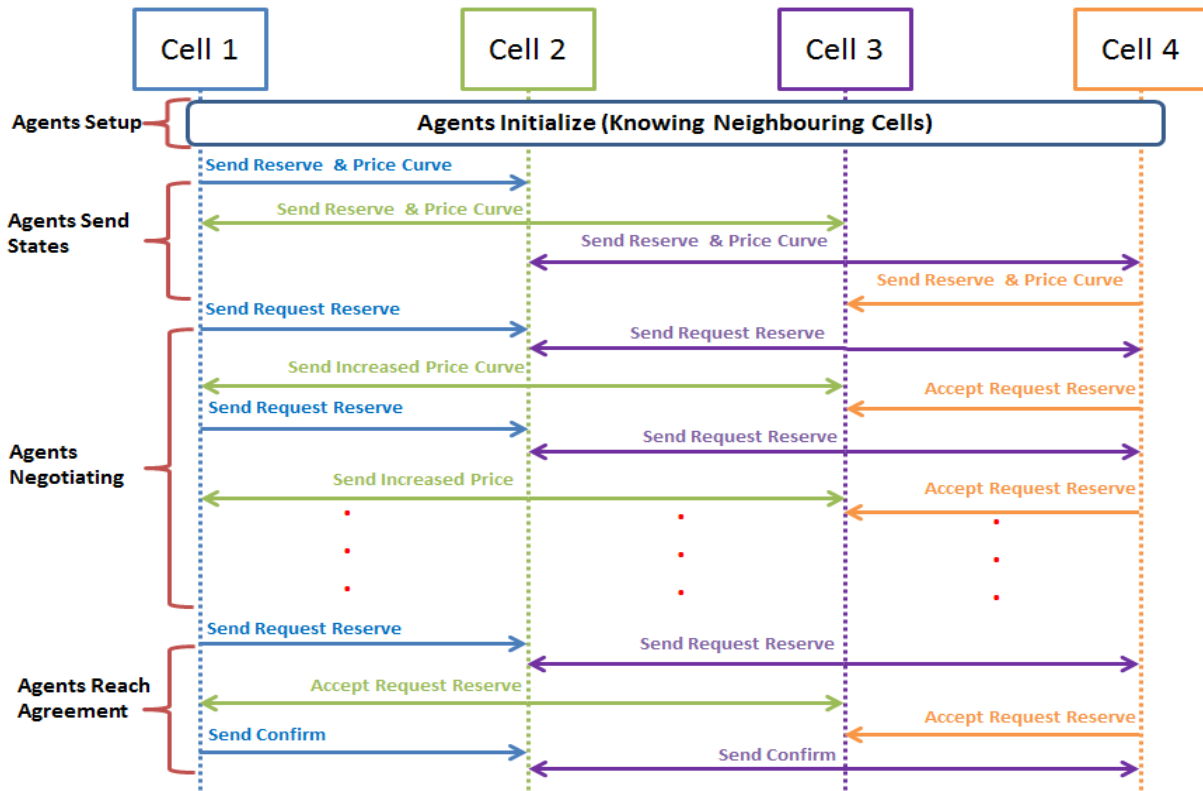


Fig.10 Four Cell Network Study: Case A agent communication diagram

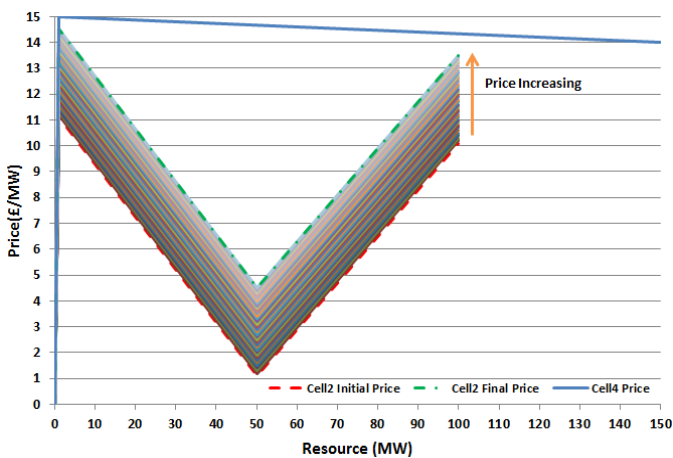


Fig.11 Four Cell Network Study: Case B price curves for cell 2 and cell 4

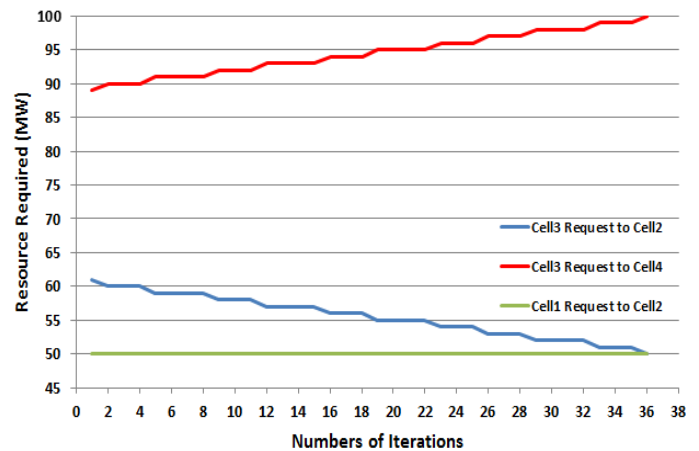


Fig.12 Four Cell Network Study: Case B iteration results for cell 1 and cell 3

negotiate with Cell 2 and Cell 4 by sending requests until an agreement is reached. Finally, Cell 3 received 150 MW from Cell 4. Cell 1 is satisfied by receiving 50 MW of reserve from Cell 2. The total number of interactions between the cells in order to achieve an agreement is 38 that can be found in Fig.9. Furthermore, the increasing price curve for each interaction of Cell 2 is also shown in Fig.8.

2) *Case B*: The price curves for Cell 2 and Cell 4 are presented in Fig.11. Cell 4 keeps the same price curve as in Case A. As the negotiation process begins, Cell 2 and Cell 4 check if the received reserve requests can be accepted or not. Cell 1 achieves the solution to receive 50 MW from Cell 2. Cell 3 receives 50 MW from Cell 2 and 100 MW from Cell 4. The total numbers of interactions between the cells to reach an agreement were 36. The results can be found in Fig 12. The increasing price curve of Cell 2 during the negotiation process is illustrated in Fig.11.

V. CONCLUSION AND FUTURE WORK

This paper described a rapid-prototype MAS implementation for negotiation of resources within distributed power systems. Using the negotiation algorithm presented and implemented, two case studies have demonstrated that the cell agents achieve the expected decision when different reserve conditions are applied. Cell agents can follow algorithm logic to interact with neighbours in order to achieve system goals by reaching agreements. The Cell agent can determine the accurate amount of resources to meet minimum cost when it operates as a deficit agent. Moreover, the cell agent successfully modifies its price curve when it is implemented as an excess agent. This research demonstrated that negotiation algorithms for distributed power systems architectures (such as the WoC) can be rapidly realised and tested in a MAS platform. Furthermore, these MAS's can be further applied as a distributed platform to support related distributed control with negotiation (such as for frequency management and voltage control). For the next step, more cell agents will be added to investigate more complex case studies and different algorithms will be researched and implemented investigating the interaction between real time controls and operational planning.

ACKNOWLEDGMENT

This work has been part funded through EPSRC HubNet (grant ref: EP/I013636/1), the Energy Technology Partnership, and the European Commission under the FP7 project ELECTRA IRP (grant no: 609687). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the European Commission.

REFERENCES

- [1] M. Liserre, T. Sauter, and J. Hung, "Future energy systems: Integrating renewable energy sources into the smart power grid through industrial electronics," *IEEE Industrial Electronics Magazine*, vol. 4, no. 1, pp. 18–37, Mar. 2010.
- [2] G. Spagnuolo et al., "Renewable energy operation and conversion schemes: A summary of discussions during the seminar on renewable energy systems," *IEEE Industrial Electronics Magazine*, vol. 4, no. 1, pp. 38–51, Mar. 2010.
- [3] A. Lugmaier, H. Brunner, B. Bletterie, F. Kupzog, and A. Abart, "Intelligent distribution grids in respect of a growing share of distributed generation," in *Proc. 2007 19th International Conference on Electricity Distribution*, pp. 1–4.
- [4] J. Mutale, "Benefits of active management of distribution networks with distributed generation," in *Proc. 2006 IEEE PES Power Systems Conference Exposition*, pp. 601–606.
- [5] A. Sabbah, A. El-Mougy, and M. Ibnkahla, "A survey of networking challenges and routing protocols in smart grids," *IEEE Trans. Industrial Information*, vol. 10, no. 1, pp. 210–221, Feb. 2014.
- [6] E. M. Davidson, V. M. Catterson, and S. D. J. McArthur, "The role of intelligent systems in delivering the smart grid," in *Proc. 2010 IEEE PES General Meeting*, pp. 1–6.
- [7] M. Wooldridge, *An Introduction to Multi-Agent Systems*, Hoboken, NJ, USA: Wiley, 2002.
- [8] P. Leitão, "Agent-based distributed manufacturing control: A state-of-the-art survey," *Engineering Application of Artificial Intelligent*, vol. 22, no. 7, pp. 979–991, Oct. 2009.
- [9] M. Pěchouček and V. Mařík, "Industrial deployment of multi-agent technologies: Review and selected case studies," *Autonomous Agents Multi-Agent System*, vol. 17, no. 3, pp. 397–431, May 2008.
- [10] P. Vrba et al., "Rockwell automation's holonic and multi-agent control systems compendium," *IEEE Trans. Systems, Man, Cybernetics*, vol. 41, no. 1, pp. 14–30, Jan. 2011.
- [11] V. Hilaire, A. Koukam, P. Gruer, and J. Müller, "Formal specification and prototyping of multi-agent systems," in *2000 International Workshop on Engineering Societies in the Agents World*, pp. 114–127.
- [12] W. Vasconcelos, D. Robertson, C. Sierra, M. Esteva, J. Sabater, and M. Wooldridge, "Rapid prototyping of large multi-agent systems through logic programming," *Annals of Mathematics and Artificial Intelligent*, vol. 41, no. 2, pp. 135–169, Aug. 2004.
- [13] T. Pedireddy and J. M. Vidal, "A prototype multi-agent network security system," in *Proc. 2003 2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, pp. 1094–1095.
- [14] L. Martini, L. Radaelli, H. Brunner, C. Caerts, A. Morch, S. Hanninen, and C. Tornelli, "ELECTRA IRP approach to voltage and frequency control for future power systems with high DER penetrations," in *Proc. 2015 23rd International Conference on Electricity Distribution*, pp. 15–18.
- [15] S. Ruthe, C. Rehtanz, and S. Lehnhoff, "On the problem of controlling shiftable prosumer devices with price signals," *International Journal of Electrical Power & Energy Systems*, vol. 72, pp. 83–90, Nov. 2015.
- [16] H. F. Wedde, S. Lehnhoff, C. Rehtanz, and O. Krause, "Intelligent agents under collaborative control in emerging power systems," *International Journal of Engineering, Science and Technology*, vol. 2, no. 3, pp. 45–59, Sep. 2010.
- [17] B. Neville, and J. Pitt, "Presage: A programming environment for the simulation of agent societies," in *2009 Programming Multi-Agent Systems*, pp. 88–103.
- [18] S. Macbeth, D. Busquets, J. Pitt, "Principled operationalization of social systems using Presage2," in *Modeling and Simulation-Based Systems Engineering Handbook*, CRC Press, pp. 43–66, 2014.