

Scalable assessment method for agent-based control in cyber-physical distribution grids

ISSN 2398-3396

Received on 9th January 2020

Revised 9th March 2020

Accepted on 1st April 2020

E-First on 19th August 2020

doi: 10.1049/iet-cps.2019.0096

www.ietdl.org

Sonja Happ¹ ✉, Stefan Dähling¹, Antonello Monti¹

¹Institute for Automation of Complex Power Systems, E.ON Energy Research Center, RWTH Aachen University, Mathieustraß e 10, 52074 Aachen, Germany

✉ E-mail: Sonja.Happ@eonerc.rwth-aachen.de

Abstract: This study proposes a scalable method to assess agent-based control concepts and deployment systems for cyber-physical electrical distribution grids. The deployment system of an agent-based control concept is the system that executes the agents and their interactions in a real implementation. Due to the increasing number of controllable loads, generators, and storage systems installed in distribution grids, scalability of control concepts and deployment systems plays a key role. Hence, the assessment method for both has to be scalable itself. For this reason, the proposed method bases on DistAIX, an open-source scalable simulation tool for cyber-physical distribution grids designed for execution on distributed computing clusters. DistAIX is extended with an interface for coupling with the deployment system of the agent-based control approach. The performance of the proposed method is evaluated for an agent-based control concept called SwarmGrid-X and the deployment system cloneMAP. The results demonstrate the feasibility of the proposed method and exemplify the functionality and scalability of SwarmGrid-X and cloneMAP.

1 Introduction

The transition towards a sustainable supply of electrical energy results in a growing amount of controllable devices in distribution systems. Their coordination is enhanced by the involvement of information and communication technology. Devices are capable of coordinated control and manipulation of the physical electrical system. Hence, distribution grids incline towards the integration of continuous physical processes with discrete communication and computational control processes resulting in cyber-physical distribution grids. Their overall behaviour emerges from the physical and cyber interactions of devices.

Agent-based control is a bottom-up concept addressing the large number and diversity of interacting devices in cyber-physical distribution grids [1]. Their complexity becomes manageable by breaking up a complex control task into small, individual, and simpler sub-tasks executed by communicating intelligent *agents*. However, agent-based control makes the analysis of cyber-physical distribution grids, especially large-scale grids, more challenging since it causes more emergent system behaviour.

The counterpart of an agent-based control concept is its deployment system in a real-world implementation. Such a system executes the agent behaviours and enables agent interactions integrated with a real system. In general, it can be a hardware or a software system, or a combination of both [2]. This paper targets deployment systems that implement software agents. Usually, the systems and methods used for the design and testing of an agent-based control concept differ from the ones used for its deployment. To ensure proper integration of an agent-based control concept with its deployment system – also with respect to scalability – a joined assessment of both is essential in advance of deploying a concept in a real system. This is the motivation for the scalable assessment method proposed in this work.

To claim the property of scalability, the behaviour of an agent-based system has to be correct and fulfil all required constraints for large numbers of agents. The amount and processing of agent communication is most critical for the scalability of agent-based systems since the communication channels and processing power of agents are limited resources. This holds for both the agent-based concept and its deployment system. Hence, besides the verification

of functionality, the amount of agent communication is in the focus of scalability assessment.

The contribution of this paper is an assessment method for agent-based control concepts addressing the following four properties:

- Functionality of the concept.
- Scalability of the concept.
- Functionality of the deployment implementation.
- Scalability of the deployment implementation and system.

While the first two properties are assessed based on pure simulation in Part I of the proposed method, the assessment of the second two properties in Part II uses Software-in-the-Loop (SiL) simulation. The software in the loop is the deployment system in combination with the implementation of the agent-based concept specific for that system. The advantage of our approach compared to existing approaches is the continuity of the simulated cyber-physical distribution grid models and the comparability of results of both parts. Results of Part I are the expected results for Part II and can be used for the identification of deviations and problems related to the functionality of the deployment implementation and the scalability of the deployment system.

The simulation tool DistAIX [3] is the core element of both parts of the proposed assessment method. For this work, DistAIX is extended with a generic and real-time capable SiL interface. The generic interface makes our approach applicable to any kind of software agent deployment system – not limited to the one that is used for the case study in this paper. Real-time capability is important because a deployment system is a real continuously executed system without time-stepped execution as in a simulation.

Existing approaches for the design and assessment of agent-based control concepts for distribution systems are discussed in Section 2. The proposed scalable assessment method is introduced in Section 3. Section 4 describes the exemplary use case for which the proposed method is demonstrated in this paper. The results are evaluated in Section 5. Section 6 concludes this paper.

2 Related work

In general, three approaches for the design and assessment of agent-based control in electrical grids can be identified in the literature:

1. Pure simulation [4–7].
2. SiL [8–12].
3. Hardware-in-the-Loop [13, 14].

In approach 1, the agent-based system and the electrical grid are both simulated. Models of both systems and an adequate coupling of these models are required for this approach. This can be achieved by integrating both parts in one simulation [4–6] or by using different modelling and simulation tools [7].

For the SiL approach, agents are executed using a multi-agent platform (MAP) as a deployment system. The electrical grid is the physical environment of the agents and is coupled with the MAP. Popular MAPs for the multi-agent system (MAS) implementation are JADE [8–10, 12] and NetLogo [11]. While JADE works well for small system sizes, it is known to have scalability issues [15]. A SiL simulation allows evaluating the concept under study not only in terms of functionality but also in terms of implementation requirements such as scalability. Hence, utilising SiL simulations researchers can study their agent-based control approaches more realistically compared to pure simulation. However, in order to generate meaningful results, the used toolchain must not introduce additional bottlenecks. To evaluate the scalability of the MAS concept with a SiL simulation, also the simulator, the MAP, and the interface between both have to be scalable.

The interface must induce only a minimal latency and computational load. Either, custom interfaces for specific combinations of simulators and MAPs can be developed as in [8–12], or a generic abstract interface can be used such as the open-source VILLASnode library [16]. It is designed for fast forwarding of data vectors that are optionally translated from one communication protocol to another. During its operation, the library functions refrain from time-costly and non-deterministic operations such as memory allocations to achieve a high throughput and a low latency fluctuation of the transmitted signals. VILLASnode supports many established communication protocols, for example Message Queuing Telemetry Transport (MQTT), nanomsg, User Datagram Protocol (UDP), Transmission Control Protocol/Internet Protocol (TCP/IP), and various custom protocols of real-time digital simulators. It is successfully used for interfacing distributed real-time simulations of electrical grids in several experiments [17–19].

In approach 3, real (embedded) devices are used to deploy and execute agents. The agent environment is either subject to simulation [14] or represented by a real laboratory setup [13] increasing the complexity of the experiment.

The advantage of the pure simulation approach is the reproducibility of results which is meaningful for systematic assessments of agent-based control. SiL and hardware-in-the-loop approaches benefit from involving real agent deployment systems in the testing of an agent-based approach. Thereby, conclusions on the correctness of the agent implementation and its performance are feasible. The assessment method presented in this paper combines both advantages in one assessment method supporting both systematic assessments of agent-based control for distribution grids with reproducible results and the involvement of an agent deployment system in the assessment. Following the study presented in [6], functionality and scalability are jointly in the focus of the assessment – and not just one of the two characteristics. Often, scalability is not well addressed, since only small use cases of electrical grids with up to 100 nodes or less are studied for a proof of concept [7–14, 20–22].

Following [23], we choose to use an approach based on parallel computing to achieve a scalable simulation of the electrical grid and the agent-based control. The simulation tool DistAIX [24] is the basis of the proposed assessment method for three reasons: First, it is shown in [3] that DistAIX exhibits outstanding scalability compared to other software tools for the simulation of

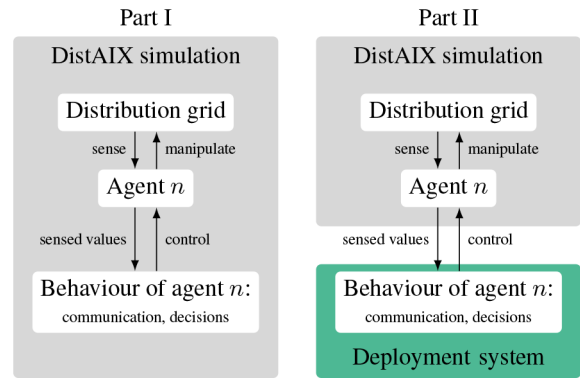


Fig. 1 Two parts of the proposed assessment method

cyber-physical distribution grids. Second, DistAIX allows for an intuitive implementation of agent-based concepts since it utilises agent-based modelling. Third, DistAIX is open source software and enables reproducibility and unobstructed reusability in the research community.

It should be noted that the utilisation of DistAIX comes with the limitation to *radial* grid topologies of any nesting and complexity. Especially in the context of distribution grids, this is only a weak limitation since these predominantly exhibit a radial topology. Meshed electrical grids cannot be simulated with DistAIX, but there exists a theoretical extension of the solution logic that can be integrated into the simulator for the simulation of weakly meshed grids [25]. Moreover, only the electrical models for the static or quasi-static simulation are thoroughly tested while dynamic phasor models that enable more detailed insight into electrical dynamics [26] are under development [3]. DistAIX's interface for electrical models is kept generic and independent of model dynamics. Therefore, even electromagnetic-transient models could be developed in future work. Throughout this work, we use the quasi-static simulation models as the simplest possible case.

3 Proposed scalable assessment method

Fig. 1 sketches both parts of the proposed assessment method. Part I aims at assessing the functionality and scalability of an agent-based control concept by means of pure simulation with DistAIX. The distribution system, the agents, and their behaviours are modelled and simulated by DistAIX. Results of Part I are evaluated regarding the overall system behaviour, and the amount of messages exchanged between agents. The system behaviour serves as an indicator for the functionality of the agent-based concept, while the number of messages indicates its scalability.

For Part II, the agent behaviour is no longer part of the DistAIX simulation. Instead, it is implemented in and executed by a deployment system that can be either a MAP or agents embedded in hardware. The deployment system is interfaced with DistAIX in a SiL simulation. Unlike Part I, DistAIX simulates in real-time in Part II so that the timing of interactions between the behaviour of the agents and the simulated physical system is as realistic as possible. System behaviour results of Part II are evaluated with respect to the functionality of the behaviour implementation. The agent behaviour computation time for growing numbers of agents is an indicator for the scalability of the deployment system itself.

3.1 Part I

The first part of the proposed method is sketched on the left side of Fig. 1. It constitutes the simulative assessment of an agent-based control strategy with DistAIX. Agents can sense and manipulate their local physical environment. They use sensed values of each time step in their behaviour to process them along with received messages from other agents. As a consequence of that processing, their behaviour issues new communication and finds control decisions. Control signals are applied to the electrical component accordingly. Agent behaviours are implemented in C++ classes with interfaces pre-defined by DistAIX agents.

For the simulation of the MAS and the cyber-physical distribution grid, each process of a DistAIX simulation follows the same schedule shown in Fig. 2 [3]. After the initialisation in step 1, a process enters the simulation loop and performs one iteration through this loop for each time step to be simulated. The loop begins with the transmission of agent messages and the execution of agent behaviours. In this step, the agent-based control concept is applied to the messages that are received by agents and the state of the physical system computed in the previous time step (or the initial state in the first time step). Hence, physical and cyber interactions emerge from this step. The power flow of the electrical grid is computed in step 3 to determine the new state of the physical system. Step 4 advances the simulation by one time step to start over again with step 2 if the simulation is not finished. During steps 2 and 3, processes require synchronisation for the computation of the correct result. This is achieved by message passing between processes.

Part I of the proposed method was already successfully applied to an agent-based concept [6]. Since simulation results obtained with DistAIX in Part I are deterministic, scenarios can be tested with parameter variations in a systematic and reproducible way. This supports the design of agent-based control concepts. Furthermore, Part I helps to understand the effects of agent communication on the overall emerging system behaviour and scalability.

3.2 Part II

The second part of the proposed method is sketched on the right side of Fig. 1. Instead of simulating the agent-based concept as agent behaviours in DistAIX, the concept's deployment system is interfaced with agents in DistAIX to exchange sensor and control values. We base the interface implementation on the VILLASnode library as introduced in Section 2. Through the VILLASnode interface that is added to DistAIX, the simulator receives inputs from the deployment system and sends back updates about the physical system state. Both happen in step 2 of the process schedule shown in Fig. 2. The whole assessment setup of Part II is a SiL simulation where the deployment system of the agent-based control concept is the software and the loop is the simulation loop of DistAIX.

3.2.1 Real-time constraint: A major difference between the pure simulation in Part I and the execution of the deployment system in Part II is the real-time execution of the latter without a defined time step. Consequently, DistAIX has to be capable of computing time steps in real-time. Due to the selected iterative method of power flow computation (forward-backward sweeping) and its distributed implementation, DistAIX cannot guarantee deterministic computations times for simulation time steps. The simulator executes faster than real-time in most cases (exceptions are discussed below). It can wait until the real-time step is completed before starting the simulation loop for the next time step. DistAIX has been extended with this feature in step 4 of the process schedule.

There are cases for which DistAIX computes a simulation time step slower than real-time. These are combinations of large scenarios and small simulation time step sizes. For these cases, the time step size has to be longer, or computing resources have to be increased (or both) until DistAIX is capable of real-time or faster than real-time execution of the desired scenario. The final time step size determined by the user has to be compatible with the emergent dynamic behaviour exhibited by the agent-based concept.

The time of day of DistAIX and the deployment system are not synchronised. This eases the setup and execution of both systems, but it also complicates the study of time-wise procedures and interactions between the two. The start of an experiment for Part II of the proposed method is always initiated by DistAIX while the deployment system is already running and waiting for the inputs of the physical system. During the experiment, both systems track their local times to document time propagation. If required, a time of day synchronisation of both systems can be added easily by synchronising the respective computing nodes with the same time

server. However, for the studied use case and performance indicators, this is not required.

3.2.2 Interface implementation: The interface between a deployment system and DistAIX is an extension of DistAIX compared to the version presented in [3]. Its purpose is enabling communication between agents in DistAIX and their respective entity in the deployment system. The VILLASnode library is integrated with DistAIX through a dedicated C++ class named *VILLASinterface*. Fig. 3 shows the integration of this class with DistAIX and the VILLASnode library. Dashed arrows mark configuration and management instructions while solid arrows show the data flow. Some communication libraries require explicit initialisation and resuming in each process that uses them. Therefore, each process invokes a method to *start* and *stop* a communication library process-wide at the beginning and end of each simulation, respectively. Internally, the VILLASnode library invokes the required functions of the communication library in use.

As in a regular simulation, agents in DistAIX execute a behaviour. In contrast to the normal simulation case of Part I, this behaviour is a non-intelligent pseudo-behaviour in Part II. The advantage of using a pseudo-behaviour for interfacing an agent in DistAIX with a deployment system is that the schedule of a DistAIX simulation remains the same as for a normal simulation (cf. Fig. 2). This avoids pervasive changes and additions to the code base and execution logic.

The pseudo-behaviour uses an object of the *VILLASinterface* class to invoke the interface management functions *init*, *destroy*, *start*, and *stop* as well as the *write* and *read* operations provided by VILLASnode. After initialising the VILLASnode instance with *init*, the parameters for communication library/protocol X are initialised and all memory allocations for the VILLASnode instance are complete. Afterwards, the VILLASnode instance is started with *start* and the *write* and *read* operations can be used. Through the invocation of the *write* operation, the values sensed in the simulated local physical environment of an agent are dispatched. The *read* operation forwards received data vectors to the pseudo-behaviour. Depending on the paradigm of the communication protocol in use, *read* and *write* operations happen synchronously or asynchronously with the simulation execution. For asynchronous operations, VILLASnode provides a buffer to store received data between two receive invocations of DistAIX. That is why agents in DistAIX can receive multiple messages per simulation time step in the order of their reception by VILLASnode. At the end of an execution of Part II, every pseudo-behaviour terminates its VILLASnode instance by invoking *stop* and the allocated memory of the VILLASnode instance is released through the *destroy* function.

The *VILLASinterface* class is generic and used in the same way for any protocol supported by VILLASnode. The only protocol-specific part of the class is the configuration of the protocol itself, for example network addresses and ports. This configuration is provided by the user and used by processes and agents to configure the communication interface. For each protocol, a small code section is added to the class to map the input protocol parameters to the ones defined in VILLASnode. VILLASnode's flexibility allows for interfacing with different kinds of agent deployment systems in Part II, for example JADE or embedded devices. The interface is not limited to the deployment system used in the case study of this paper.

4 Use case: SwarmGrid-X and cloneMAP

The proposed assessment method is applied to a use case consisting of the agent-based control concept SwarmGrid-X [6] and the open-source deployment system cloud-native Multi-Agent Platform (cloneMAP) [27]. SwarmGrid-X is selected since this concept is already comprehensively introduced and analysed in previous work [6] where it has proven its functionality and scalability in simulations. However, there exists no study yet that assesses an implementation of SwarmGrid-X for a deployment system. Therefore, it is the ideal candidate to challenge the

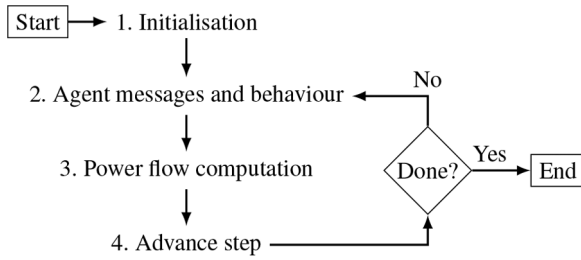


Fig. 2 Schedule of a DistAIX simulation process (adapted from [3])

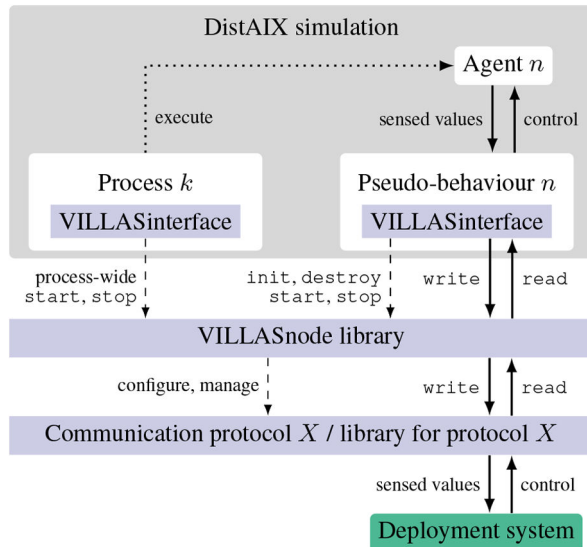


Fig. 3 Interface between DistAIX simulation and deployment system using VILLASnode library

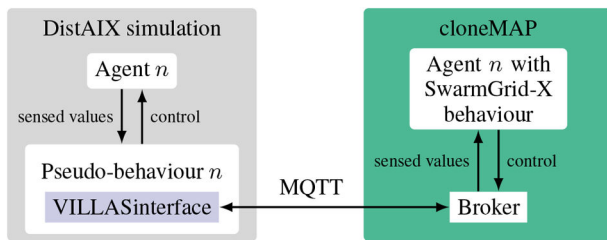


Fig. 4 Use case for Part II of the proposed assessment method including SwarmGrid-X and cloneMAP

assessment method proposed in this paper. The choice of the relatively new MAP cloneMAP instead of a well-known and established MAP such as JADE was made, because cloneMAP's design goals include modularity and scalability, while existing approaches, especially JADE, are already known to have scalability issues [15] and there would be no value in identifying these again with this paper. Hence, cloneMAP is an appropriate choice to demonstrate especially the deployment system scalability assessment of Part II.

For a proof of concept of Part I, SwarmGrid-X is simulated using DistAIX to assess the agent-based concept and the resulting overall system behaviour for one specific scenario. As there exists a previous study of this kind for SwarmGrid-X [6], this part is kept relatively short here. SwarmGrid-X is implemented in DistAIX and cloneMAP, as described in [6]. The DistAIX version serves as a use case for Part I of the proposed method while the cloneMAP version is the use case for Part II.

Fig. 4 shows the use case for the evaluation of Part II of the proposed method using cloneMAP as cloud-based deployment system for SwarmGrid-X. The simulated agents send measurement values to their respective representation in the deployment system every 10 s using the MQTT protocol. Agents executed in cloneMAP determine the components' control values by means of SwarmGrid-X. These control values are sent back to the simulation

via MQTT whenever they change. The VILLASnode library launches one additional thread per DistAIX simulation process for asynchronous MQTT communication independent of the simulation computation. Received messages are queued by VILLASnode instances until they are acquired by the simulation to avoid data loss.

For both parts of the assessment, a simulation time step of 1 s is used. This choice was made based on the evaluation of results presented in [6], which indicate that this time step is sufficient to capture relevant characteristics of SwarmGrid-X. However, the proposed assessment method is not limited to this time step. In case of a control concept with a more dynamic behaviour, also smaller time steps and the use of other electrical models, e.g. dynamic phasor models (see Section 2), are possible. The choice of MQTT is based on its widespread use, e.g. in the Internet of Things (IoT) applications. Due to the use of VILLASnode, also other protocols are available. The update rate of one update every 10 s was found experimentally. A higher update rate would not change the results of Part II significantly (cf. Section 5.2.2), but would induce unnecessarily high network traffic in the execution system. The highest possible update rate depends on the size of the scenario under study, the resources of the execution system, and the communication protocol in use. It can be determined experimentally, as done for the use case of this paper in Section 5.2.1.

4.1 Agent-based control concept: SwarmGrid-X

SwarmGrid-X is an agent-based control concept for flexible components in distribution grids introduced by the authors in [6]. One agent is assigned to each flexible component. The structure of the distribution grid, consisting of multiple voltage levels, i.e. high, medium and low voltage, is mapped to a holonic architecture. The agent topology and control architecture can be seen in Fig. 1, published in [6]. Substation agents represent the underlying grid in the superior grid and act as a single prosumer. Similarly, the substation agent represents the superior grid to the agents in the underlying grid. This method ensures that communication occurs only among agents in the same grid and with the substation. Consumers and producers negotiate the amount of active and reactive power that can be consumed and produced. Thereby, power is balanced as local as possible using the available flexibility. A swarm concept is introduced to enable the agents to find each other. Consumer agents find producers close to their own location. If more producers are needed to cover the demand, those with a larger distance are added to the swarm by a recruiting protocol. The directory facilitator (DF), a typical component in MAS applications, is used for that purpose. The swarm shrinks again in case producers have not been contacted in a while.

The authors show in [6] that the concept can mitigate voltage band violations introduced by high penetration of renewable power resources. This is mainly achieved by improved reactive power management compared to typical grid codes. Also, the provision of a requested amount of reactive power to the transmission system for ancillary services is demonstrated. Due to the agent-based approach, SwarmGrid-X works completely decentralised without the need for a central coordinator. SwarmGrid-X is used in this work to exemplarily evaluate the proposed assessment method. While SwarmGrid-X aims specifically at a local balancing of power and mitigation of voltage band violations, the proposed assessment method can be applied to any agent-based control concept independent of the control target.

4.2 Deployment system: cloneMAP

The open-source platform cloneMAP [27] is a MAP designed to exploit features of modern cloud-computing to achieve high scalability and fault-tolerance. It is implemented in Go and consists of four modules that are deployed as Docker containers to a Kubernetes cluster [28]. The four modules are (1) the core module responsible for MAS and agent creation, monitoring, and termination, (2) the DF module for service discovery, (3) the IoT module comprising an MQTT broker and (4) a logging module for storing application output and debug logging. Agents interact with

Table 1 Installed powers in LV grids of basic scenario.
Table adapted from [6]

	Rural LV grid		Urban LV grid	
	Qty	Power (capacity)	Qty	Power (capacity)
Load	175	204 kW	175	353 kW
EV	26	104 kW	35	195 kW
PV	35	390 kW	23	138 kW
Wind	5	108 kW	0	—
CHP	25	325 kW	16	222 kW
HP	0	—	14	387 kW (322 kWh)
Battery	35	118 kW (301 kWh)	21	65 kW (166 kWh)

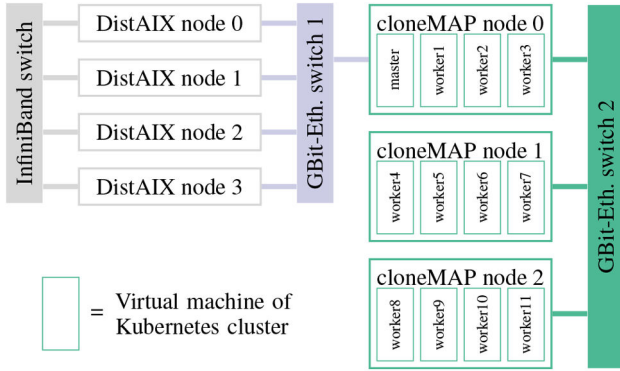


Fig. 5 Computing hardware and network setup for the use case

the modules by means of a REST API. The modules are implemented such that they are stateless. All state information is stored in distributed databases. As a result, the modules are horizontally scalable, preventing possible bottlenecks. Moreover, agents are deployed in groups of *agencies*. The number of agents per agency results from the number of agencies and the total number of agents. The distribution of agents to agencies and the modular design of cloneMAP lead to a microservice architecture [29]. Kubernetes orchestrates the single components deployed as Docker containers. Using a Kubernetes cluster enables easy up-scaling of single components and fault-tolerance by means of Kubernetes' monitoring features [30]. For all scenarios, the number of agents per agency is determined such that one agency per Kubernetes worker node is created. cloneMAP serves as an example deployment system for this work. Note that it could be replaced by any other MAP (e.g. JADE).

4.3 Implementation of SwarmGrid-X on cloneMAP

For Part II of the proposed assessment method, SwarmGrid-X is transferred to cloneMAP as an example of a real-world implementation of an agent-based control concept. The implementation is realised as close as possible to the original agent behaviour implementation in DistAIX. The DF module of cloneMAP serves as DF used during swarm forming in SwarmGrid-X. The main difference is that each agent is executed continuously in cloneMAP while the execution in DistAIX is performed with a time step.

In each DistAIX time step, all agents execute the agent behaviour described above. As a result, events and agent actions can only occur at discrete times in DistAIX. Moreover, multiple executions of a pure DistAIX simulation (Part I) always lead to the same results, because events such as a change of the physical system or the receiving of agent messages always happen in the same order. In cloneMAP, the negotiation behaviour is executed whenever the agent's state changes, i.e. new sensor values from the component or messages from other agents are received. Since the execution of agents is not synchronised, the order of events is influenced by non-deterministic constraints, for example the scheduling of agent threads to CPUs or the IP-based messaging. This leads to a time-wise non-deterministic behaviour of SwarmGrid-X in cloneMAP. Hence, the numerical results of

multiple executions of the same scenario can slightly differ, even though SwarmGrid-X steers the overall system behaviour towards a similar outcome in all executions. The reason for this is that the goal of every agent remains the same in all executions, while the points of time at which negotiations take place are slightly different. As a result, slightly different contracts are negotiated, differing in the contract partners and the contracted amount of power, to fulfil the local agent goal of a local power balance. This leads to resembling but not exactly equal system-level results for consecutive executions.

The reproducibility of results in Part I constitutes an inevitable feature for a systematic evaluation of the control concept under study, i.e. SwarmGrid-X. However, the execution of SwarmGrid-X on a platform such as cloneMAP in Part II of the proposed method yields a more realistic system-level behaviour since the described non-deterministic effects would also occur in real-world execution. This underlines the importance of the proposed assessment method.

4.4 Scenarios

We use Scenario 1, defined in [6] as a basic scenario. It consists of two 400 V low-voltage (LV) grids with the same topology (177 buses, also used in [3, 31]) which are coupled to a 20 kV medium-voltage (MV) grid. The installed powers for both grids are listed in Table 1. Cable lengths of both grids vary between 10 and 40 m. One of the two LV grids is a power-consuming urban grid that supplies many consumers such as single- and multi-family houses, electric vehicles (EVs), and heat pumps (HPs). The second LV grid is a rural grid exhibiting a temporary generation surplus from photovoltaic (PV) and wind generation and less power consumption than the urban grid.

Based on the basic scenario, larger scenarios are derived for scalability assessments. For up-scaling, the basic scenario is duplicated and connected N times to the same HV/MV substation.

$$A(N) = N \cdot 585 + N \cdot 2 + 1 \quad (1)$$

The number of communicating agents A of an up-scaled scenario is computed according to (1) where $N \cdot 585$ denotes the number of communicating agents except for substations and the term $N \cdot 2 + 1$ represents the number of substations contained in the scenario.

4.5 Computing and network infrastructure

The evaluation of the proposed method involves both high-performance computing for DistAIX and cloud computing for cloneMAP. Fig. 5 sketches on how computing nodes are interconnected for this purpose. DistAIX uses a maximum of four identical nodes with 24 CPUs and 128 GB RAM each. These nodes are interconnected with an InfiniBand network that is used for inter-process communication during a simulation. They run a CentOS 7 operating system with a 4.15.0 Linux kernel. DistAIX is executed directly on the computing nodes without any virtualisation applied.

The computing infrastructure of cloneMAP consists of three identical nodes with 16 CPUs and 64 GB RAM each. They run an Ubuntu 16.04 operating system with a 4.4.0 Linux kernel and use a dedicated Gbit-Ethernet switch (switch 2 in the figure) for internal cloneMAP network traffic. DistAIX nodes and one of the cloneMAP nodes are interconnected via another Gbit-Ethernet switch (switch 1 in the figure). This switch is not a dedicated switch for the interconnection of the two subsystems, but a general-purpose switch that also routes non-related network traffic of other clients. It is used for the communication between the simulator and the deployment system via the MQTT protocol.

The Kubernetes cluster used for the execution of cloneMAP consists of one master and 11 worker virtual machines (VMs). Each cloneMAP node deploys four VMs with an equal share of CPUs and RAM (4 vCPUs and 12 GB RAM). The master VM is always deployed on cloneMAP node 0 since this node has the network interface to switch 1. All cloneMAP modules are randomly deployed as Docker containers in one worker VM,

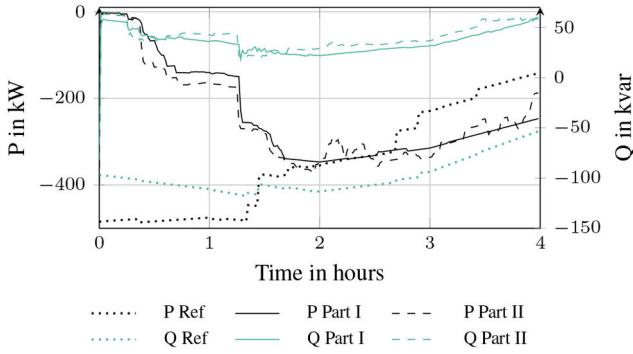


Fig. 6 Active and reactive power of the basic scenario

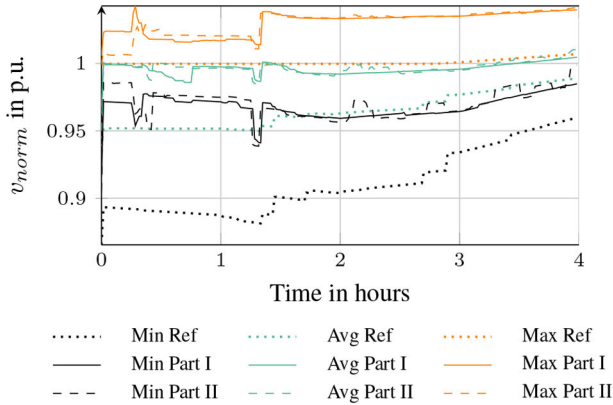


Fig. 7 Normalised voltages of the basic scenario

except for the IoT module which is always deployed on a worker running on cloneMAP node 0 to enable fast communication between master VM and MQTT broker. Eclipse Mosquitto is used as single-instance MQTT broker in the IoT module of cloneMAP with disabled Nagle algorithm [32] to avoid negative effects on the latency at the cost of a potentially lower throughput for small packet sizes. For all experiments in this paper, each worker VM deploys one agency so that in total 11 agencies are used.

5 Evaluation of use case

This section introduces performance indicators for both parts of the proposed assessment method and discusses the results obtained for the use case.

5.1 Part I: SwarmGrid-X simulated with DistAIX

For Part I, the basic LV assessment scenario introduced in Section 4.4 is simulated with DistAIX for a period of 4 h with a time step size of 1 s. As a reference to the results of SwarmGrid-X, a reference behaviour identical to the one used in [6] is simulated with DistAIX for the same scenario as introduced in Section 4.4. In the reference behaviour, agents do not communicate and find their set-points based on the conditions in their local physical environment as well as currently valid German guidelines (VDE-AR-N 4105). To evaluate the functionality of SwarmGrid-X, the active (P) and reactive power (Q) at the slack node as well as the normalised node voltages v_{norm} are compared to those of the reference behaviour. Since the functionality of the agent-based concept has already been studied extensively in [6], its assessment is kept short here and serves as a proof of concept of Part I of the proposed method.

The scalability of SwarmGrid-X is evaluated based on the maximum and average number of sent messages per agent for 1 s and the complete 4 h period. Substation agents are considered separate of all other agents due to their extraordinary task in SwarmGrid-X of imparting between voltage levels. The amount of transmitted messages is a key factor for the scalability of an agent-based concept since the physical links and computing resources available for communication are limited in real systems and

Table 2 Number of sent messages per agent in SwarmGrid-X

Sent messages per agent except substation agents					
N	Agents	Avg./s	Max./s	Avg./4 h	Max./4 h
1	585	0.017	89	244.55	4218
2	1170	0.017	89	238.39	3321
5	2925	0.017	89	245.95	4292
10	5850	0.018	89	250.04	4189

Sent messages per substation agent

N	Substations	Avg./s	Max./s	Avg./4 h	Max./4 h
1	3	4.790	301	67,544.33	110,932
2	5	5.472	301	77,159.80	104,886
5	11	6.639	468	93,619.82	114,158
10	21	7.597	486	107,125.91	161,616

therefore require thoughtful exploitation. The scalability assessment uses scenarios resulting from $N = 1$, $N = 2$, $N = 5$, and $N = 10$ applied to the up-scaling method introduced in Section 4.4.

5.1.1 Functionality: Results for active and reactive power at the slack node of the basic scenario are shown in Fig. 6 and the respective maximum, average, and minimum normalised voltages are shown in Fig. 7. In the first 15 min of the simulated period, the rural and urban LV grids manage to balance active power consumption and production through SwarmGrid-X, resulting in a power demand close to zero. This is achieved by activating the available flexibility of batteries and combined heat and power (CHP) systems. Since most of the batteries are discharged after 1.25 h, the power demand of the overall system increases by ~ 100 kW at this time. From then on, there is not sufficient active power flexibility available to alter and decrease the power demand to that of the reference behaviour. In the reference behaviour, EVs charged with full power until their batteries are fully charged yielding a decreasing power demand of EVs towards 4 h. In SwarmGrid-X, the accumulated power demand of all EVs is almost constant throughout the entire period resulting in the larger active power demand of the overall system starting at 2.75 h compared to the reference behaviour.

SwarmGrid-X uses reactive power flexibility successfully to improve the voltage quality of the system. While the overall grid behaves capacitive for the reference behaviour (a negative sign of reactive power) and exhibits a violation of the 10% voltage band at 0.9 pu for a duration of almost 1.5 h (see Fig. 7), SwarmGrid-X yields an inductive system behaviour and prohibits any voltage band violation. Furthermore, the voltage band used by this concept is significantly more narrow than the one of the reference behaviour underlining the meaningful activation of reactive power flexibility through SwarmGrid-X. The voltage drop at 1.25 h is a consequence of the active power demand increases at this time. SwarmGrid-X manages to compensate the drop within a couple of minutes by activating reactive power flexibility of charging EVs. For a more detailed discussion of the functionality of SwarmGrid-X, the reader is referred to [6] where a similar methodology as the one of Part I of the proposed assessment method is applied to study the concept's capabilities including its capability of following a reactive power set-point provided by the transmission system.

5.1.2 Scalability: According to the results provided in Table 2, the maximum and average amount of sent messages per second and the complete 4 h period remain relatively constant for increasing scenario sizes and the majority of agents excluding substation agents. This proves good scalability properties of SwarmGrid-X and is a consequence of local agent communication in each LV grid. The course of sent messages over time in Fig. 8 underlines that the amount of communication caused by SwarmGrid-X is influenced by the magnitude of changes in the physical system. Each cross in the figure represents the accumulated sent messages

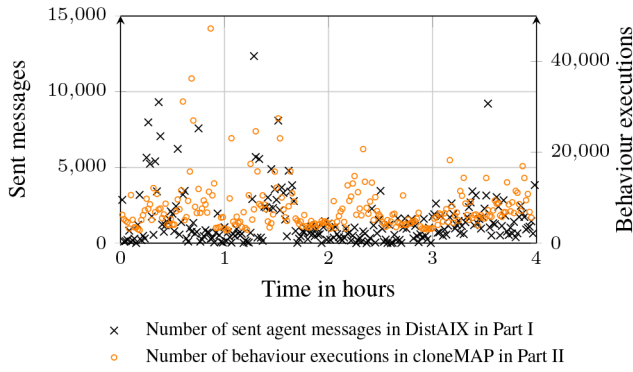


Fig. 8 Number of sent agent messages for Part I (DistAIX) and number of behaviour executions for Part II (cloneMAP) accumulated per minute

Table 3 MQTT RTTs in ms

N	DistAIX processes	DistAIX nodes	Avg.	95th
1	8	1	45	78
2	16	2	128	162
5	40	4	293	461
10	40	4	688	1165

of 1 min. In times of constant conditions of the physical system, less messages are transmitted (see the period from 0.75 to 1.25 h) whereas the significant change in active power at 1.25 h results in a maximum of transmitted messages and entails an increase of communication until the consequences of the change are compensated.

Compared to the rest of the agents, substation agents send considerably more messages (up to a factor of 428 more in 4 h). They negotiate with potentially all agents in the underlying LV grid and other agents in the MV grid. As indicated by the results in Table 2, the average and maximum values of messages per second and 4 h increase with increasing scenario size. The communication of substations agents is identified as the major limitation factor for the scalability of SwarmGrid-X. However, the increase factor of the number of sent messages per substation agent is considerably smaller than the increase factor for the scenario size. Improvements of SwarmGrid-X should attempt to reduce the number of sent messages of substation agents to further improve the scalability of the agent-based control concept. For now, the concept is implemented as it is in Part II of the proposed assessment method.

5.2 Part II: SwarmGrid-X deployed with cloneMAP

Part II of the proposed assessment method is applied to SwarmGrid-X deployed in cloneMAP. For the identification of a feasible time step for the real-time execution of DistAIX, the constraints of the interface between deployment system and simulator for the given setup of computing hardware and network infrastructure (cf. Section 4.5) are assessed. This assessment measures and evaluates the round-trip-time (RTT) of MQTT messages transmitted from the representation of an agent in cloneMAP to the agent in DistAIX and back to cloneMAP. All agents transmit such messages in parallel yielding the worst-case communication effort between platform and simulator. DistAIX does not run in real-time for this assessment and cloneMAP does not execute any agent behaviour. The average RTT over 1000 measurements per agent is evaluated in addition to the maximum RTT of 95% of all messages (95th).

The functionality of SwarmGrid-X deployed with cloneMAP is assessed by a comparison of DistAIX results obtained for parts I and II. Active and reactive power at the slack node as well as node voltages of the basic scenario are compared to identify and understand deviations between the results of parts I and II.

The computation time for an agent behaviour serves as a performance indicator for the scalability of cloneMAP itself (minimum, average, and maximum). It is used here since this time has a major influence on the response time of cloneMAP to a

change in the physical system. Similar to the scalability assessment of SwarmGrid-X in Part I, up-scaled assessment scenarios for $N = 2$, $N = 5$, and $N = 10$ are used. However, for Part II these scenarios are executed in real-time, that means for a real-time duration of 4 h each. It is expected that larger scenarios increase the utilisation of the cloneMAP computing nodes and thereby increase the computation time for agent behaviours. Identifying the characteristics of this increase is the target of the scalability assessment.

Moreover, the frequency of agent behaviour execution over time is evaluated. Since the agents execute the negotiation behaviour only if their state changes, it can be expected, that the number of executions increases at times with significant changes in the physical system. This assessment is compared to the one for agent messages in Part I.

5.2.1 Interface constraints: Table 3 contains the measured average and 95th RTTs in milliseconds and the number of DistAIX nodes and processes involved in the experiments. The required MQTT thread per simulation process limits the number of simulation processes per node to a maximum of 12 to ensure proper parallelism of communication threads and simulation. With increasing scenario size, the RTT between cloneMAP and DistAIX agents increases. This behaviour is expected as the single-instance MQTT broker and the single network interface to cloneMAP node 0 are potential bottlenecks for network traffic. The average RTT is below 0.7 s for all tested scenarios and the values of 95th support the conclusion that the large majority of RTTs is below 1.165 s. With respect to the update rate of 10 s that simulated agents in DistAIX use to send measurement updates to cloneMAP (cf. Section 4), the results in Table 3 indicate that a much higher update rate is feasible with MQTT and the given setup. For a different communication protocol, the RTT experiment would have to be repeated to determine the highest possible update rate.

Because this experiment is a worst-case communication scenario between cloneMAP and DistAIX, a real-time simulation time step of 1 s is save to use in all following experiments. A real-time step of 1 s means that an incoming message is processed by the respective agent in DistAIX latest 1 s after its reception. For the application of SwarmGrid-X, this is sufficient, as previous works achieved positive results with the same time step [6]. If required, the RTT can be improved by using multiple network interfaces to access the cloneMAP nodes and by deploying a distributed MQTT broker instead of a single-instance variant.

5.2.2 Functionality: The dashed lines in Fig. 6 show the active and reactive power results for SwarmGrid-X executed with cloneMAP for Part II. They follow the course of the results obtained with DistAIX in Part I with some deviations. Reactive power is closer to the simulation result than active power. On average, the active power at the slack node obtained in Part I is 8.67 kW higher than the result of Part II while the reactive power is on average 4.63 kvar smaller for Part I. The existing deviations are no errors. They are caused by a slightly different timing in cloneMAP with respect to the receipt of agent messages. This again results in other contracts for single agents compared to the simulation and hence in a deviant flexibility usage throughout the 4 h period. Due to the non-deterministic behaviour of agent message transmission in cloneMAP, both approaches find different and likewise correct results. In further studies, the agents acting differently can be identified along with the contracts leading to the differences.

Fig. 7 contains the node voltages obtained with Part II. Similar to the slack results, these show deviations from the results of Part I but follow their course in general. It is especially notable, that the voltage drop at 1.25 h entails almost the same voltages in parts I and II. This means that the substation agents behave similarly to their detection and solution.

We verified the benefit of Part II of the proposed assessment method for debugging during several experiments carried out prior to the one that provided the results of Figs. 6 and 7. Comparing the results obtained with cloneMAP to the ones of Part I enabled the detection of flaws in cloneMAP, for example a race condition in

Table 4 Behaviour execution times in μs

N	Directory facilitator instances	Min.	Max.	Avg.
1	1	33	5,103,829	274
2	1	34	5,095,336	243
5	1	32	5,035,606	386
10	1	n/a	n/a	n/a
10	3	29	23,064,799	752

agent message transmission. Using the results of Part I as a reference proved to be essential for narrowing down implementation mistakes.

5.2.3 Scalability: The minimum, maximum, and average execution times of the agent behaviour for all agents and the entire experiment period of 4 h are used as metrics for the scalability assessment of cloneMAP and the implementation of SwarmGrid-X. Table 4 shows the execution times for an increasing scenario size. The behaviour execution time reveals the reaction time of an agent to changes of the component's state or to messages from other agents. It includes only the time required for the decision making regarding agent actions but not the time required to execute the actions. Possible agent actions are a change of the component's set points or further exchange of messages with other agents. Minimum and maximum values remain almost unchanged for most scenario sizes. Only for the biggest scenario, the maximum increases due to increased utilisation of the computational nodes of cloneMAP. An increasing scenario size leads to a higher number of agents and hence, also to more interactions among agents. Both require more computing resources. Since only 4 vCPUs per Kubernetes worker are available, the possible parallelism of agent behaviour and operating system execution is limited. The average value varies between 243 and 752 μs . The available computing resources restrict the performance for larger scenarios. In the pure DistAIX simulation of Part I, the behaviour is executed only every second due to the simulation time step. In cloneMAP it remains below 1 ms for all scenarios which is sufficient for SwarmGrid-X. Therefore, even bigger scenarios could be executed with the cloneMAP setup shown in Fig. 5.

Table 4 also reveals the number of DF instances used for each scenario. Due to the microservice approach used for cloneMAP, the single modules can be scaled horizontally. For a large scenario size, a single instance DF becomes a bottleneck. This is observable in row five of Table 4, where the execution of Part II fails if only one DF instance is used (marked by n/a in the table). Scaling the DF to three instances increases the possible scenario size that can be executed. This is an important outcome of Part II of the proposed assessment method. It shows that the scalability of the DF is a crucial feature of the MAP chosen for implementation. Therefore, other MAPs, which do not implement a scalable DF, are not suitable for the control concept if it is applied to such large scenarios. A pure simulation as in Part I cannot lead to the same conclusion.

Fig. 8 shows the total amount of behaviour executions of all agents accumulated per minute for the entire experiment period. Similar to the number of sent messages in Part I, the number of behaviour executions correlates with changes in the physical system. As a result, the number of behaviour executions increases during the same time periods as the number of sent messages. During times of a rather constant physical system behaviour, the number of behaviour executions remains low. This reveals the importance of studying cyber and physical systems together as both influence each other and hence, also influence the performance of a control concept such as SwarmGrid-X.

6 Conclusion and outlook

The proposed two-part assessment method enables large-scale testing of functionality and deployment of given distributed agent-based control approaches for distribution grids. This paper successfully demonstrates that parts I and II of the method can be applied to an agent-based control concept and a deployment system

intended for real-world execution of the concept. Distribution grids of up to 3563 buses with 5871 intelligent agents ($N = 10$) are tested successfully with the method proving its scalability. The results support the development and analysis of the agent-based concept itself with respect to its functionality and scalability (Part I). Additionally, they allow for conclusions on the feasibility of the implementation of the concept with a real deployment system (Part II).

Due to the conceptual generality of the proposed method, it can be used with other deployment systems and control concepts. The flexibility provided by the VILLASnode library as well as the pseudo behaviour approach in DistAIX are the basis for a generic coupling with cloud-based applications related to distribution grid monitoring and control.

7 Acknowledgment

The authors gratefully acknowledge the financial support provided by the German Federal Ministry of Economic Affairs and Energy in the research project AGENT under grant no. 03ET1495A and the German Federal Ministry of Education and Research in the research project ENSURE under grant no. 03SFK1CO.

8 References

- [1] Vrba, P., Marik, V., Siano, P., *et al.*: 'A review of agent and service-oriented concepts applied to intelligent energy systems', *IEEE Trans. Ind. Inf.*, 2014, **10**, (3), pp. 1890–1903
- [2] Leitão, P., Karnouskos, S., Ribeiro, L., *et al.*: 'Smart agents in industrial cyber-physical systems', *Proc. IEEE*, 2016, **104**, (5), pp. 1086–1101
- [3] Kolen, S., Dähling, S., Isermann, T., *et al.*: 'Enabling the analysis of emergent behavior in future electrical distribution systems using agent-based modeling and simulation', *Complexity*, 2018, **2018**, pp. 1–16
- [4] Chassin, D.P., Fuller, J.C., Djilali, N.: 'GridLAB-D: an agent-based simulation framework for smart grids', *J. Appl. Math.*, 2014, **2014**, pp. 1–12
- [5] Kryonidis, G.C., Kontis, E.O., Chrysochos, A.L., *et al.*: 'A simulation tool for extended distribution grids with controlled distributed generation'. 2015 IEEE Eindhoven PowerTech, Eindhoven, Netherlands, 2015
- [6] Dähling, S., Kolen, S., Monti, A.: 'Swarm-based automation of electrical power distribution and transmission system support', *IET Cyber-Phys. Syst., Theory Appl.*, 2018, **3**, pp. 212–223
- [7] Appino, R.R., Listmann, K.D., Chicco, G.: 'Agent-based model for distributed optimization of microgrids operation'. Proc. of the 2017 IEEE PES ISGT Europe Conf., Torino, Italy, 2017
- [8] Aquino-Lugo, A.A., Klump, R., Overbye, T.J.: 'A control framework for the smart grid for voltage support using agent-based technologies', *IEEE Trans. Smart Grid*, 2011, **2**, (1), pp. 173–180
- [9] Colson, C.M., Nehrir, M.H.: 'Comprehensive real-time microgrid power management and control with distributed agents', *IEEE Trans. Smart Grid*, 2013, **4**, (1), pp. 617–627
- [10] Mangiatordi, F., Pallotti, E., Panzieri, D., *et al.*: 'Multi agent system for cooperative energy management in microgrids'. 2016 IEEE Int. Conf. on Environment and Electrical Engineering (EEEIC), Florence, Italy, 2016, pp. 1–5
- [11] Rodemann, T., Eckhardt, T., Unger, R., *et al.*: 'Using agent-based customer modeling for the evaluation of EV charging systems', *Energies*, 2019, **12**, (15), p. 2858
- [12] Nizami, M.S.H., Hossain, J., Fernandez, E.: 'Multi-agent based transactive energy management systems for residential buildings with distributed energy resources', *IEEE Trans. Ind. Inf.*, 2020, **16**, pp. 1836–1847
- [13] Belkacemi, R., Feliachi, A., Choudhry, M.A., *et al.*: 'Multi-agent systems hardware development and deployment for smart grid control applications'. 2011 IEEE Power and Energy Society General Meeting, Detroit, USA, 2011, pp. 1–8
- [14] Constantin, A., Löwen, A., Ponci, F., *et al.*: 'Design, implementation and demonstration of embedded agents for energy management in non-residential buildings', *Energies*, 2017, **10**, (8), p. 1106
- [15] Mengistu, D., Tröger, P., Lundberg, L., *et al.*: 'Scalability in distributed multi-agent based simulations: the JADE case'. 2008 Second Int. Conf. on Future Generation Communication and Networking Symposia, Sanya, China, 2008, pp. 93–99
- [16] FEIN Aachen e.V.: 'VILLASnode – real-time capable simulation gateway' (Aachen, Germany, 2020). Available at <https://www.fein-aachen.org/projects/villas-node/>, accessed 9 March 2020
- [17] Vogel, S., Mirz, M., Razik, L., *et al.*: 'An open solution for next-generation real-time power system simulation'. 2017 IEEE Conf. on Energy Internet and Energy System Integration (EI2), Beijing, China, 2017
- [18] Monti, A., Stevic, M., Vogel, S., *et al.*: 'A global real-time superlab: enabling high penetration of power electronics in the electric grid', *IEEE Power Electron. Mag.*, 2018, **5**, (3), pp. 35–44
- [19] Mirz, M., Vogel, S., Schäfer, B., *et al.*: 'Distributed real-time co-simulation as a service'. 2018 IEEE Int. Conf. on Industrial Electronics for Sustainable Energy Systems (IESES), Hamilton, New Zealand, 2018, pp. 534–539
- [20] Gubina, A.F., Medved, T., Prislán, B., *et al.*: 'A method for optimal schedule balancing in distribution network using multi agent control strategy'. 13th Int. Conf. on the European Energy Market (EEM), Porto, Portugal, 2016, pp. 1–5

- [21] Hu, S., Xiang, Y., Liu, J., *et al.*: 'Agent-based coordinated operation strategy for active distribution network with distributed energy resources', *IEEE Trans. Ind. Appl.*, 2019, **55**, (4), pp. 3310–3320
- [22] Shirazi, E., Jadid, S.: 'Autonomous self-healing in smart distribution grids using multi agent systems', *IEEE Trans. Ind. Inf.*, 2019, **15**, pp. 6291–6301
- [23] Kuschel, C., Rüdte, U.: 'High-performance simulation of nation-sized smart grids', *Int. J. Parallel Emergent Distrib. Syst.*, 2017, **32**, pp. 647–668
- [24] FEIN Aachen e.V.: '*DistAIX framework – scalable simulation of cyber-physical power distribution systems*' (Aachen, Germany, 2020). Available at <https://www.fein-aachen.org/projects/distaix/>, accessed 9 March 2020
- [25] Shirmohammadi, D., Hong, H.W., Semlyen, A., *et al.*: 'A compensation-based power flow method for weakly meshed distribution and transmission networks', *IEEE Trans. Power Syst.*, 1988, **3**, (2), pp. 753–762
- [26] Demiray, T., Andersson, G., Busarello, L.: 'Evaluation study for the simulation of power system transients using dynamic phasor models'. 2008 IEEE/PES Transmission and Distribution Conf. and Exposition: Latin America, Bogota, Colombia, 2008
- [27] FEIN Aachen e.V.: '*cloneMAP – cloud-native multi-agent platform*' (Aachen, Germany, 2020). Available at <https://www.fein-aachen.org/projects/clonemap/>, accessed 9 March 2020
- [28] Bernstein, D.: 'Containers and cloud: from LXC to Docker to Kubernetes', *IEEE Cloud Comput.*, 2014, **1**, (3), pp. 81–84
- [29] Salah, T., Zemerly, M.J., Yeun, C.Y., *et al.*: 'Iot applications: from mobile agents to microservices architecture'. 2018 Int. Conf. on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 2018, pp. 117–122
- [30] Sayfan, G.: '*Mastering Kubernetes: large scale container deployment and management*' (Packt Publishing, Birmingham, United Kingdom, 2017)
- [31] Kolen, S., Isermann, T., Dähling, S., *et al.*: 'Swarm behavior for distribution grid control'. Proc. of the 2017 IEEE PES ISGT Europe Conf., Torino, Italy, 2017
- [32] Nagle, J.: 'Congestion control in IP/TCP internetworks', *ACM SIGCOMM Comput. Commun. Rev.*, 1984, **14**, (4), pp. 11–17