RESEARCH

Open Access

Modeling control strategies for prosumers in a Python-based modular simulation tool



Andrea Schoen^{1,2*}, Jan Ringelstein¹, Denis Mende^{1,2} and Martin Braun^{1,2}

From The 12th DACH+ Conference on Energy Informatics 2023 Vienna, Austria. 4-6 October 2023. https://www.energy-informatics2023.org/

*Correspondence: andrea.schoen@iee.fraunhofer.de

 ¹ Fraunhofer Institute for Energy Economics and Energy System Technology IEE, Joseph-Beuys-Strasse 8, 34117 Kassel, Germany
 ² Department of Energy Management and Power System Operation, University of Kassel, Wilhemshoeher Allee 73, 34121 Kassel, Germany

Abstract

The planned massive increase of producers and consumers such as electric vehicles, heat pumps and photovoltaic systems in distribution grids will lead to new challenges in the electrical power system. These can include grid congestions at the low voltage level but also at higher voltage levels. Control strategies can enable the efficient use of flexibilities and therefore help mitigate upcoming problems. However, they need to be evaluated carefully before their application in the energy system to avoid any unwanted effects and to choose the most fitting strategy for each application. In this publication, a Python-based modular simulation tool for developing and analysing control strategies for prosumers, which uses pandapower (Thurner et al. 2018), is presented. It is intended for sequential simulations and enables detailed operational analyses, which include evaluating the influence on grid situations, the necessary behavior of energy system components, required measurements and communications. This publication also gives an overview of control strategies, existing simulation tools, how the modular simulation tool fits in and illustrates its functionalities in an application example, which further highlights its versatility and efficiency. Time series simulations with the tool allow analyses regarding the effect of control strategies on power flow results. Moreover, the simulation tool also facilitates evaluating the behavior of energy system components (e.g. distribution substations), necessary communications and measurements as well as any faults that might occur.

Keywords: Distribution grids, Power system modeling, Distribution system operation, Prosumers, Control strategies, pandapower

Introduction

Due to Germany's ambitious goals to increase renewable generation to at least 80 % by 2030 and reducing the dependency on fossil fuels, the number of new loads and generators, such as electric vehicle (EV) charging stations, heat pumps (HPs) and photovoltaic (PV) systems, is rapidly increasing (German Federal Government 2023). Control strategies can be useful to help mitigate negative impacts on the grid situation, e.g. by reducing grid congestions. This is especially relevant for loads like EV charging stations



© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http:// creativecommons.org/licenses/by/4.0/.

and HPs. Before implementing any control strategy into the real energy system, it is vital to analyse its behavior and expected influence on the grid. As described in (Schoen et al. 2021a) and shown in Fig. 1, this is a multi-level process starting with a theoretical analysis, followed by a simulation-based analysis and generally completed by lab-based analyses and/or field tests. In this paper, a Python-based modular simulation tool for modeling control strategies for prosumers in distribution grids is presented. It closes the gap in the analysis process by offering a connecting link between the analysis of grid planning aspects and lab-based analysis using co-simulation tools and later field tests. This simulation tool makes use of pandapower power flow controllers, which are originally intended for simulating grid assets that are controlled based on power flows in electrical grids (Thurner et al. 2018; Fraunhofer IEE and University of Kassel 2023a, 2023b). An innovate way of using these power flow controllers in a coordinated, flexible and modular simulation tool in an architecture motivated by co-simulation tools, such as OpSim (Fraunhofer IEE 2023a), is applied and presented in this paper. This enables modeling relevant power system components, such as distribution system operator (DSO), distribution substations and home energy management systems (HEMS), and communications between them. By making use of the pandapower controllers, non-concurrent operations and communications with each other can also be modelled. This allows modeling control strategies and performing detailed operational analyses by means of versatile simulations within the Python-based pandapower framework.

This paper presents the modular simulation tool and is structured as follows: First, an overview of control strategies and their role in grid operation is given in section Grid Operation and Control Strategies for Prosumers, followed by a summary of requirements for simulation tools for operational analyses of control strategies and existing tools in section Simulation Tools for Operational Analyses of Control Strategies. Subsequently, section Modular Simulation Tool for Control Strategies presents an in-depth description of the modular simulation tool including an application example. Finally, the conclusion and outlook are given.

Grid operation and control strategies for prosumers

The large number of low voltage (LV) grids (\sim 500,000 LV grids in Germany (Mende 2022)), which are often operated decoupled from each other, in conjunction with disproportionate expenses for connection/accessibility and remote controllability, has made active grid operation in the lowest voltage levels virtually impossible. First steps to mitigate the influences of a large number of distributed generation systems are implemented by means of distributed and similar plant behavior based on characteristic curves. These



Fig. 1 Modular simulation tool in the multi-level analysis of control strategies (based on (Schoen et al. 2021a))

have also found their way into the corresponding connection guidelines for distributed generation systems (e.g. VDE-AR-N 4105 for the connection to low voltage grids (VDE 2023) or the guidelines for feed-in management in Germany (Bundesnetzagentur 2023)). Due to expected increase in system loading and decreasing distances to operational limits, an active management of generation plants and consumers becomes increasingly relevant, which is why control strategies for decentralized prosumers gain importance.

Control strategies for prosumers

In this paper, the term control strategy refers to controlling consumers, producers or prosumers on the LV level of the electrical power grid according to a pre-defined control behavior. This section focuses on control strategies for EV charging, which are currently of highest interest due to the fast increase in the share of EVs (ADAC 2023) and the potential for shifting and adapting charging processes as a result of the long parking times of an average EV in comparison to its active usage. Additionally, there are also several approaches for controlling PV systems, HPs, battery storage systems and strategies for combinations thereof (Gomes et al. 2022). There is a large variety of control strategies for electric vehicle charging such as control-signal-based strategies and specificationbased strategies (Schoen et al. 2020). The latter can be further divided into power-based and price-based strategies (Schoen et al. 2021a). The DSO often defines the specifications in power-based strategies based on the knowledge about the grid and its operational limits. An example of a power-based strategy is a time-dependent Power Limit, where the charging power limit is determined based on the average annual load profiles (Zhang et al. 2014; Schoen et al. 2023). The power limit is lower in times of high values in the annual load profile and vice-versa. The goal of price-based strategies is often related to minimizing costs for the energy provider and/or for the customer by shifting charging processes (Hildermeier et al. 2023; Birk Jones et al. 2022). The DSO Access introduced in (Schoen et al. 2020) is an example for a control-signal-based strategy, where the DSO can actively curtail the charging powers in case of a critical grid situation (here: red BDEW signal phase (BDEW 2023)).

Classification of control strategies

Control strategies for prosumers can be further classified using criteria such as: *objective(s), operation architecture, input, control variables, method* and *communication* (Wenderoth et al. 2019). This is helpful for their clear description and an important preparatory step for their implementation into simulation tools. An *objective* in this context refers to the goal of a control strategy. Table 1 gives an overview of relevant *operation architectures*. Operation unit refers to a component, which makes decisions for the operation based on the processed information. A power system component that is controlled

Table 1	Overview of	operation	architectures	ífrom (Wenderoth	et al. 2019	9))

	Local	Decentralized	Distributed	Centralized
Number of operation units	1	> 1	> 1	1
Number of controlled elements	1	> 1	> 1	> 1
Coordination between operation units	N/A	No	Yes	N/A

locally or by another entity is referred to as controlled element. There is also a hierarchical architecture, which refers to the organisation of operation units in several levels that depend on each other and have a clear hierarchy. *Input* refers to the values that are needed to realize a control strategy, which can be measurements or set point values. *Control variables* can also be set point values or physical values. The *method* describes the algorithm or approach that is used within a component to reach the *objective*. Evaluating the *communication* of a control strategy and comparing it with the status quo in the energy system is vital for judging the current practicability of a strategy and determining necessary developments to enable its implementation (Wenderoth et al. 2019).

Simulation tools for operational analyses of control strategies

As shown in Fig. 1, the analysis of control strategies is a multi-level process. Operational analyses regarding the influence of control strategies on selected grid situations and scenarios are an important part of this. They can be used to evaluate the necessary behavior of energy system components, the communication between them, potential variations and faults. This can either be done as part of the simulation-based or lab-based analysis or a combination of both. This section summarizes requirements for tools for such operational analyses, presents existing ones and highlights the relevance and novelty of the modular simulation tool.

Tool requirements for modeling control strategies for operational analyses

The classifiers introduced in section Classification of Control Strategies need to be integrated into a simulation tool to perform effective operational analyses with control strategies. Moreover, such tools also need to be able to consider the hierarchies of the relevant energy system components as mentioned in the Introduction. Figure 2 shows the relevant hierarchy levels for modeling control strategies as introduced in (Schoen et al. 2020): DSO, distribution substation, HEMS and prosumer. The figure represents one element each in a system with one DSO, where several components can be on levels 2, 3 and 4. These hierarchy levels can work together in the different operation architectures shown in Table 1. Generally, the HEMS and prosumer(s) on the lower hierarchy levels are either controlled by the distribution substation or DSO at the upper hierarchy levels. Figure 2 represents the main communication streams of the control strategies, but additional communication streams are also possible (e.g. directly between DSO and



Fig. 2 Relevant hierarchy levels for modeling control strategies (adapted from (Schoen et al. 2020))

HEMS). Moreover, the figure also represents goals of the levels and possible interaction with upper voltage levels. In addition to that, simulation tools also need to be able to correctly take into account the various relevant input parameters, control variables, objectives and necessary communications of the control strategies.

Existing tools for operational analyses

There are already several tools that enable operational analyses at the LV level and can therefore also model control strategies for prosumers. In this section, the tools shown in Table 2 are considered.

A common approach is using multi-agent systems, where the agents can model any type of component of the energy system and therefore allow the implementation of a wide variety of control strategies (Mahela et al. 2022). Reference (Papadopoulos et al. 2013) introduces an example of this, where control strategies for EV charging are implemented and analysed using a multi-agent system (MAS), which models relevant system components as agents, such as the DSO or the EVs. The agent-based tool Simona allows detailed modelling of grid system participants, including the modelling of technical grid equipment, regulation, monitoring and control algorithms (TU Dortmund University 2023; Römer et al. 2019). Moreover, the power distribution system simulation and analysis tool GridLAB-D (Pacific Northwest National Laboratory 2023) also provides capabilities for modeling an agent-based system with a focus on power system modeling.

There are also several other approaches for evaluating the influence of control strategies on electrical grids, based on modeling the electrical grid and adding controllable elements to it. Such modeling and the corresponding execution of time series simulations can be done with tools such as pandapower (Thurner et al. 2018), Grid-Sim (Forschungsstelle fur Energiewirtschaft e.V 2023) or MATLAB (The MathWorks, Inc. 2023). Additionally, control behaviour can also be integrated into such grid modeling approaches by integrating time series data, where the control behavior is already considered. For this, tools such as datafev can be used (Gümrükcü et al. 2023).

Another common modeling approach, which can combine the strengths of the aforementioned ones is the use of co-simulations. Reference (Vogt et al. 2018) gives an

Tool name	Tool type	Reference
MAS for EV control strate- gies	Multi-agent system	Papadopoulos et al. (2013)
Simona	Agent-based simulation environment	TU Dortmund University (2023) and Römer et al. (2019)
GridLAB-D	Power system simulation and analysis tool	Pacific Northwest National Laboratory (2023)
pandapower	Power system modeling tool	Thurner et al. (2018)
GridSim	Power system modeling tool	Forschungsstelle für Energiewirtschaft e.V. (2023)
MATLAB	Programming and numeric computing platform	The MathWorks, Inc. (2023)
datafev	Python package for modeling EV charging	Gümrükcü et al. (2023)
OpSim	Co-simulation tool	Fraunhofer IEE (2023a)
helics	Co-simulation tool	NREL (2023b2023)
beeDIP	Data integration tool	Requardt (2021)

Table 2	Simulation	tool	overview
---------	------------	------	----------

overview of existing co-simulation tools for modeling smart grid solutions. One of these tools is OpSim, which is a test and simulation environment for the development and testing of smart grid solutions, where components of the energy system as well as the electrical grid can be modeled and combined in a co-simulation (Fraunhofer IEE 2023a). Moreover, the project ReCharge (NREL 2023a) highlights the use of the co-simulation tool helics (NREL 2023b) in a project for evaluating EV control strategies. An important part of modeling control strategies is the preparation for their integration into the real energy system. For this, data integration tools such as beeDIP (Requardt 2021) are needed. These tools can also be used to prepare the energy system integration of control strategies, which can also be tested beforehand by emulating components such as the electrical grid or DSO.

Relevance and novelty of developed methodology

The aforementioned tools offer a large variety of possibilities for modeling control strategies for prosumers. Many of them are also able to fulfill the introduced tool requirements. Co-simulation tools such as OpSim also enable the coupling of different software tools within one simulation enabled by its flexible message bus architecture, which allows a wide array of investigations regarding grid operation and any type of control behavior. Modules such as pandapower controllers offer many possibilities regarding the implementation of control algorithms for grid assets in power flow simulations. With the goal of developing a lightweight pandapower-based tool that allows for researching and developing control strategies including agent-like controller interaction, we developed the modular simulation tool, which uses pandapower controllers in a new way that enables an integrated implementation of relevant energy system components, communications between them and a coordination of relevant control behaviors in a pandapower-based Python environment without the need for additional libraries next to pandapower, e.g. to handle the messaging. This provides a tool that enables detailed analyses of control strategies regarding their impact on the electrical grid and aids in the development and testing of control strategies, which can be further integrated into tools, such as co-simulation frameworks or data integration tools.

Modular simulation tool for control strategies

This section presents the modular simulation tool and how the pandapower (Thurner et al. 2018) control module is used to build it. First the pandapower control module and the system architecture are presented. Subsequently, the system components are described, including how they can communicate with each other and how the correct simulation sequence is ensured. Additionally, insights into how control strategies can be modeled and analyzed with the system are given. Finally, an application example is presented to highlight the functionalities and some possible applications of the tool.

pandapower control module

The modular simulation tool is implemented in Python and uses pandapower to model the relevant energy system behavior (Thurner et al. 2018; Fraunhofer IEE and University of Kassel 2023a). pandapower is a Python-based tool for modeling, analysing and optimizing power systems. pandapower can be used to model electric grids and all relevant components, such as lines, transformers, switches, loads, generators and others. Additionally, it also allows for time series simulations consisting of subsequent power flow calculations. By making use of the control module, any control behavior can also be integrated into these time series simulations through the use of so-called controllers. Generally, the objective of one controller is adapting the behavior of a singular grid asset. The modular simulation tool uses these pandapower controllers in an innovative way allowing the coordination of different components, which do not necessarily need to be a grid asset, and enables the communication between them in an architecture that is motivated by architectures such as OpSim. Controllers are Python classes, where the control behavior is realized in the method *control step* of the controller, which can adapt any grid parameters based on the implemented control behavior. Before the *control* step can be executed, it needs to be prepared appropriately. The preparatory steps vary depending on the desired control behavior but can include performing an initial power flow, updating input and control variables for the current time step or modifying any grid parameters. Thereafter, the *control_step* can be executed. Its input always includes the current state of the pandapower grid model it is applied to and the current time step. Other relevant parameters can be processed in the *control step* by making use of class variables that need to be defined and/or adapted before the *control_step* is applied. Any control behavior can be realized here, such as a transformer tap position optimization, PV power plant curtailment or adapting the active powers of any load or generator according to a specified control strategy. The latter is most relevant for the application in the modular simulation tool. Generally, controllers are used to adapt the behavior of single grid assets, but it is also possible to add other types of functionalities, e.g. modeling entities like a DSO acting as a central controller, which the modular tool makes use of.

In many cases, there are several controllers in a pandapower grid model that need to be executed in a certain sequence. This can be realized by the *Cascade Control* functionality that is integrated in the pandapower control module. With this functionality, levels and orders can be assigned to controllers. This allows executing controllers and applying their behavior in a specified sequence. Figure 3 represents how the level and order parameters can be used to define the sequence of the controllers within one time step. At first, all controllers of the first level (here: level 0) are executed in the specified order (here: starting with order 0 up to *m*), afterwards all controllers on the next level (here: level 1) are executed in the specified order and so on. Moreover, before a new level is started, all controllers of the previous level have to indicate that they have converged.



Fig. 3 Cascade control (adapted from (Fraunhofer IEE and University of Kassel 2023b))

As long as this is not the case, they are re-executed starting with the lowest-order controller again. This allows for multiple executions of the same controller within a single time step. Once all controllers have been executed, the next time step can be calculated. This allows a variety of different applications of the controllers. The modular simulation tool makes use of these *Cascade Control* functionalities to model hierarchy levels as illustrated in Fig. 2 in its tool architecture and for ensuring the correct simulation sequence. Additionally, the modular simulation tool also enables the communication between controllers and a detailed initialization process for all modeled components, which is uncoupled from their control functionalities.

The pandapower functionalities provide the foundation for the modular simulation tool. It uses the existing capabilities for modeling electrical grids and performing power flow simulations, and the presented functionalities of the control module. While the tool makes use of these functionalities, its tool architecture, components, and especially the communication capabilities, are additional implementations that are done within the modular simulation tool. To this, the tool provides additional implementations: (i) a standardized class for general controllers that do not represent grid assets and are equipped with a mailbox, (ii) a messaging scheme for exchanging information between such controllers, (iii) message send and receive methods and (iv) a scheme that organizes control strategy simulation with seamless integration into the existing pandapower controller architecture. These new implementations are needed to create a tool, where relevant energy system components and their interaction can be modeled with a level of detail that reproduces the real energy system as closely as possible while enabling agent-like controller interaction without the need for additional packages other than pandapower. The following sections present these new implementations.

Tool architecture

The goal of the tool is modeling the components shown in Fig. 2 and also enabling the representation of the architectures presented in Table 1. The tool architecture is motivated by the architecture of OpSim (Fraunhofer IEE 2023a) with the difference that it directly coordinates pandapower controllers. The system consists of different components, which include the *DSO*, the *Substation(s)* and the *HEMS/Prosumers* (Schoen et al. 2021a). Prosumers are modeled as part of the HEMS and can be any type of load or generator that is related to the household. There is also the components are part of. Additionally, the *Simulation Tool Core* is needed to enable all communications between the components by feeding a public message list in a mailbox-like structure. Moreover, it ensures the efficient initialization of the simulation *Control* is required for setting the parameters of the simulation. Figure 4 gives an overview of the components and their interaction.

Figure 5 shows how the components work together during the simulation. It also highlights that the *Simulation Control* and *Simulation Tool Core* are responsible for ensuring the correct simulation structure (including initialization, communication and sequence). The components *Grid Calculation*, *DSO*, *Substation(s)* and *HEMS / Prosumers* are part of the simulation process and are called successively in order to perform



Fig. 4 Simulation tool components (adapted from (Schoen et al. 2021a))





Fig. 6 Three-layered structure of a simulation tool process component

power flow calculations and implement control strategy behavior. Therefore, the terms structure components and process components will be used in the following. Each of the process components is modeled by a Python class with a wide variety of functionalities. To enable an efficient implementation of these components, they are structured into three layers as shown in Fig. 6. The light blue ellipse represents a process component as shown in Figs. 4 and 5. The upper layer *Initialization* is used to initialize the components and their variables. The second layer is the *Controller* itself, which is created in the layer *Initialization*. Depending on the component, several controllers can be created. As the control functionalities can be quite complex, they are allocated to the third level *Control Functionality*. Messages are processed and sent during the initialization of the simulation and in each time step. This is why the *Initialization* and *Control Functionality* layers are equipped with a message box each. The integration of the messaging functionality

using pandapower controllers is a key feature of the modular simulation tool. Here, a method is implemented for each control strategy, which is called when the strategy is active in a simulation to execute the corresponding behavior of the component. Several control strategies can be active in one simulation by combining the different methods appropriately. There are two approaches for realizing the combination of control strategies: A new method can be implemented, which combines the desired functionalities or existing methods are executed after one another. Here, the method of the control strategy with the highest priority is executed last, e.g. when a *Power Limit* and *DSO Access* are active at the same time, the *DSO Access* is executed last to ensure that no additional congestions are caused by the application of the *Power Limit*.

Simulation tool structure components

The components *Simulation Tool Core* and *Simulation Control* are key elements for the simulation tool structure and ensure the initialization, communication and correct simulation sequence.

Simulation tool core

The Simulation Tool Core fulfills the following main tasks:

- Starting the simulation: triggering the initialization of all components and the start of the time series simulation once all components are initialized,
- Handling communication between the components by: adding all incoming messages to the intended inboxes (*process*) and sending all outgoing messages to the intended receiving components (*send*).

Simulation control

Different grids, future scenarios, various control strategies and any combinations thereof can be considered with the modular simulation tool. Additional parameters are also defined, e.g.: the time frame that is to be simulated, where the input data is to be collected from and which power flow results are to be saved and where. All relevant simulation parameters are set in the *Simulation Control*, where they can be easily adapted by the user. When a simulation is started by the *Simulation Tool Core*, the aforementioned parameters are distributed from the *Simulation Control* to the relevant components to initialize the simulation. Once all components are initialized, the *Simulation Tool Core* starts the time series simulation. It is also possible to parameterize the simulation using a Graphical User Interface (GUI), which is implemented using Plotly Dash (Plotly 2023). It can be used to define all of the aforementioned simulation parameters, starting and stopping the simulation and also displaying the results of the simulation.

Simulation tool process components

The process components *Grid Calculation*, *DSO*, *Substation(s)* and *HEMS / Prosumers* all work according to the same principle, where they need to go through an initialization process first before the start of the time series simulation. The properties and behavior of these components will be described in two steps: initialization and time step behavior. During the initialization, the components process the incoming messages and get prepared accordingly. If needed, they also send messages during the initialization to update other components on their status. The initialization varies depending on the components, but the following actions need to be performed for all components in the *process*-method of the *Initialization* layer as shown in Algorithm 1:

- Handle strategy specifications: All parameters related to the active strategy or strategies are processed and handled accordingly. Generally, this means updating the corresponding class variables or adding new ones so the specifications can easily be accessed or updated during the simulation. Some examples for variables that could be set initially (and updated later if applicable) are: operating limits, power limits, price curves, etc.
- Create *Controllers*: The defined number of *Controllers* is created:
 - GridCalculation: Two Controllers GridControl1 and GridControl2 are created.
 - DSO: One Controller is created as it is assumed that one DSO is responsible for the modeled grid.
 - Distribution Substation(s): One Controller per distribution substation in the modeled grid is created.
 - HEMS / Prosumers: One Controller per household is created.

Algorithm 1 Component initialization	
<pre>for message in messageBox do: sender = message[0], receiver = message[1], content = message[2] if receiver == currentComponent then variablesInfo = interpret(content) initVariables = updateVariables(variablesInfo) strategy = initVariables.strategy controllers = createControllers(Core) controllers.addControlFunctionality(Core)</pre>	

The *Controllers* are connected to the *Simulation Tool Core* and the component itself and the *Control Functionality* is added to it. If any other actions need to be performed during the initialization, it is specified explicitly in the following. Otherwise, only the time step behavior of the components is described. Within one time step, each component exhibits the behaviour as shown in Algorithm 2, which is realized in the layer *Control Functionality*. The *if*-query can be extended depending on the number of implemented strategies. Any variables that were previously initialized can be updated here.

Algorithm 2 Time step behavior
<pre>for message in messageBox do: sender = message[0], receiver = message[1], content = message[2] if receiver == currentComponent then timestepVariablesInfo = interpret(content) timestepVariablesInfo = interpret(content)</pre>
<pre>timestepVariables = updateVariables(timestepVariablesInfo) if strategy == strategyA then perform strategyAbehavior(timestepVariables) else if strategy == strategyB then perform strategyBehavior(timestepVariables) [] send new messages to new receiver</pre>

Grid calculation

An important task of the Grid Calculation is setting up the following input data:

- A base grid model (grid topology, existing consumers and producers, line/transformer standard types, measurement data (transformer drag pointer)),
- Placement and specifications of additional consumers and producers according to the desired simulation scenario (e.g. high PV penetration),
- Time series data for all consumers and producers (with a fitting resolution, e.g. 15 min, and at least for the time frame that is to be simulated).

Within the initialization phase of the simulation, this component performs the following tasks in addition to the general initialization tasks:

- Set up the grid model: The grid model is set up according to the specified grid parameters. It can be directly loaded with all scenario and time series information already added to it if it has been previously configured and saved. Alternatively, a new grid model can be configured according to the specified grid parameters. This means that future consumers and producers (incl. the corresponding time series information) are added to the base grid model.
- Handle which results are to be saved where: In detail, this means that the output writer of the pandapower time series module is configured.
- Start the time series simulation: Once all components are configured, the time series simulation is started as stated above. The actual corresponding pandapower function is located in the *Grid Calculation* component. After this call, pandapower takes over the control flow and calls the controllers according to the fixed cascade sequence as shown in Fig. 3.

In each time step, the Grid Calculation controllers perform the following tasks:

• GridControl1: This *Controller* is called at the very beginning of a time step. It performs a power flow calculation and sends the required results to the receiving components. In most cases, this is the *DSO*, which takes care of any further handling of this information.

• GridControl2: This is the last *Controller* that is called in a time step. It updates the control variables in the grid model (mostly active and/or reactive power set points of consumers and/or producers) so that they can be considered for the initial power flow in the following time step.

DSO

In each time step, the *DSO* handles incoming information from the *Grid Calculation*. Depending on the active strategy or strategies, the *DSO* makes decisions regarding the behavior of other components, i.e. *Distribution Substation(s)* and/or *HEMS / Prosumers*. The results of these are then sent to the corresponding recipients. Algorithm 3 shows an example of a potential strategy behavior after handling the incoming information for the *DSO Access* strategy. Here, it is assumed that the DSO can determine the elements *criticalGridElements* in the grid which violate the operating limits. In the simulation, this information is sent from the *Grid Calculation* to the *DSO* and added to the *timestepVariables* at the beginning of the time step when incoming messages are interpreted as shown in Algorithm 2.

Algorithm 3 DSO example: DSO Access behavior
perform DSOAccessBehavior(timestepVariables)
criticalGridElements = timestepVariables.criticalGridElements
loadsToBeCurtailed = getLoadsToBeCurtailed(criticalGridElements)
sendMessage(sender = $^{-1}$ DSO', receiver = 'Distribution Substation(s)', content = {'critical loads': loadsToBe-
Curtailed})

Distribution substation(s)

If applicable, the *Distribution Substation(s)* component refines the decisions made by the *DSO* after processing the incoming information. This is mostly done based on the additional information about the grid section corresponding to the substation. An example of this could be the fair distribution of charging contingents to individual charging stations in a strategy where the *DSO* allocates charging contingents to substation areas based on the grid situation (Schoen et al. 2020). At the end of a time step, the decisions are sent to the corresponding recipients, e.g. *HEMS/Prosumers* in the grid section. Continuing the *DSO Access* example, Algorithm 4 shows a potential behavior of the *Distribution Substation(s)* component, where it forwards the information *loadsToBeCurtailed* from the *DSO* to the *HEMS*.

Algorithm 4 Distribution Substation(s) example: DSO Access behavior				
perform DSOAccessBehavior(timestepVariables)				
loadsToBeCurtailed = timestepVariables.loadsToBeCurtailed				
sendMessage(sender = 'Distribution Substation(s)', receiver = 'HEMS', content = {'critical loads': loadsTo-				
BeCurtailed})				

HEMS/Prosumers

First, all incoming information is processed. Depending on the active strategy or strategies and available information, the control variables are then updated here, e.g. the charging powers are curtailed based on an active charging power limit in the time step. Finally, the updated control variables are sent back to the *Grid Calculation* for their use in the next time step. In the *DSO Access* example, the *HEMS/Prosumers* component can exhibit the strategy behavior as shown in Algorithm 5, where the *newChargingPower* is determined.

Algorithm 5 HEMS / Prosumers example: DSO Access behavior
perform DSOAccessBehavior(timestepVariables)
loadsToBeCurtailed = timestepVariables.loadsToBeCurtailed
if householdID in loadsToBeCurtailed then
newChargingPower = curtailChargingPower(originalChargingPower)
end if
$sendMessage(sender = 'HEMS', receiver = 'GridCalculation', content = \{'New P values'; \{'elementID'; house-\}$
holdID, 'chargingPower': newChargingPower}})

In addition to the shown functionality, the *HEMS/Prosumers* component also takes care of handling any energy that was curtailed and gradually increasing the charging power after a curtailment. This avoids starting a cycle of mitigating a violation in one time step and causing one again in the subsequent one by increasing the charging power too quickly, while the overall grid situation is still close to operating limits. As one controller per household in the considered grid is created, this behavior takes place for each household individually. Each controller is referenced to a household, where the uncontrolled behavior is portrayed by the corresponding load/generation profile. Depending on the desired behavior in the controlled case, the *HEMS/Prosumers* component is defined accordingly.

Sequence and communication

This section provides a closer look at the simulation sequence as shown in Fig. 5 and how the components communicate with each other. The *Cascade Control* functionality is used within this simulation tool to ensure the correct sequence of the process components within each time step of the time series simulations that are performed with it. Table 3 presents the hierarchy of the simulation tool process components.

The levels and orders are used to ensure the correct sequence in a time step. The *Grid Calculation* component has two main tasks and is therefore called twice: (i) performing the initial power flow at the very beginning of a time step and (ii) updating any grid parameters (e.g. charging powers) if they were changed during the time

Component	Level	Order	
Grid Calculation	0 and 5	1 and 1	
DSO	1	1	
Distribution Substation(s)	2	1, 2,, m ₁	
Profile Update	3	1	
HEMS/Prosumers	4	1, 2, m ₂	

Table 3 Hierarchy of the simulation tool process components (adapted from (Schoen et al. 2021a))

step because of the control strategy behavior. In addition to all components shown in Figs. 4 and 5, there is one more functionality needed for the time series simulations: updating all loads and generators with the corresponding profile data (active power, and— if applicable—reactive power) in every time step. This is done internally in the time series simulation and is referred to as *Profile Update* here. This happens after the *DSO* and *Distribution Substation(s)* are called as their behavior is to be based on the results of the previous time step to model the behavior of the real energy system as closely as possible, where decisions of the DSO or other actors can generally only be made based on what has already happened in the electrical grid. By the time any decisions are made, some time has generally passed and the grid situation has changed. The sequence of the components, especially *Profile Update*, is supposed to model this behavior.

Each of the components in Table 3 (except for *Profile Update*) is equipped with a message box as shown in Figs. 4 and 6 to enable receiving messages from other components. Each component can also send messages to any other component. Both of these functionalities are realized by making use of the Simulation Tool Core, which can distribute messages between the components. While communication between all components is theoretically possible, any type of communication behavior (e.g. which components communicate with each other in which frequency) can be implemented to model different existing and future communication infrastructures in the energy system. As one of the applications of this system is developing and testing control algorithms with the pandapower framework before their integration into co-simulation platforms such as OpSim (Fraunhofer IEE 2023a), the modular simulation tool aims to mimic message-bus-based communication. Each component is equipped with a pipe-like structure that is implemented as a public message list which is fed by the Simulation Tool Core. There is no need for synchronisation since the controller components are executed subsequently during the time series simulation by pandapower callbacks, with the order fixed by the level/order setup. The data model for the messages is kept simple to ensure an efficient implementation process and to keep the focus of the system on evaluating the grid impact of control strategies. The versatility and convenient interpretability of Python Strings is used for this purpose. The sender and receiver need to be specified as well as the content itself. The message should contain a keyword describing its content, e.g. "Critical loads", and and the corresponding information in a format that can be integrated into a String and interpreted easily later on. The receiver then splits these messages into its elements (sender, receiver, content), checks if it has been delivered correctly, and if so, interprets the message accordingly. The OpSim framework uses a similar format, but there it is augmented with a time stamp to allow for conservative synchronization. Figure 7 shows a communication example with four components taking place within one time step. In this example *Component A* sends two messages, one intended for *Component* B and one for Component C. The Simulation Tool Core distributes the messages to the intended recipients. These handle the messages internally and both send one message intended for Component D. The handling of all messages is done by the Simulation Tool Core and the correct component sequence is ensured by the defined level and order of the components.



Fig. 7 Communication example with four components

Implementation of control strategies

As the correct simulation sequence and communication are ensured as described before, the implementation can focus on: behavior of the components, the question which components need to communicate and specification of information that needs to be exchanged between them. A theoretical evaluation of the control strategies according to the classifiers described in section Classification of Control Strategies (Wenderoth et al. 2019) immensely supports the preparation of the implementation as it inherently tackles the aforementioned points. An important additional part of the preparation is defining individual component functionalities according to the expected behavior of the real system. Moreover, when planning the communication between the components, the data availability in the real system should be taken into account. Different data availabilities and qualities can also be considered and compared in this process. After this preparatory process, the derived functionalities can be efficiently implemented into the simulation tool components, tested and improved. Furthermore, it is also possible to extend the system with additional components if the control behavior cannot be realized with the existing ones alone. This can be done by implementing a new component with the structure as described in section Tool Architecture and adapting the levels and orders of all components accordingly. It is possible to flexibly do this for one strategy but keeping the participating components and their levels and orders unchanged for any other strategy.

Application example

This section presents an application example to illustrate the functionalities and possible applications of the modular simulation tool. For this purpose a distribution grid model with nine substations, 1062 customers and a total line length of 59 km is used. It is configured for a possible future scenario with randomly distributed 182 PV systems, 580 HPs and 651 EV charging stations at the LV level. The grid model and the time series data are based on the LV3 data of the benchmark data set *SimBench* (Fraunhofer IEE and University of Kassel 2023c; Meinecke et al. 2020) and the strategies *Power Limit* and *DSO Access* described in Control Strategies for Prosumers are used. Table 4 shows the result of the application of selected classifiers from section Classification of Control Strategies.

The properties identified in Table 4 are implemented within the modular simulation tool to enable the analysis of these strategies. The main difference between both approaches in terms of *objective* and *method* is the preventative nature of the *Power*

Classifier	Power Limit	DSO Access
Objective(s)	Avoidance of critical grid situations	Congestion management
Operation Architecture	Centralized	Centralized
Input	Average load profile based on historic data	Current state of the grid
Control Variables	Charging powers	Charging powers
Method	Preventative reduction of charging powers accord- ing to power limit	Reactive curtailment of charging stations

Table 4 Application of selected classifiers to Power Limit and DSO Access

Table 5 Evaluation of communications between DSO, Substation(s) and HEMS

	Power Limit		DSO Access		
	Information	Possible frequency	Information	Possible frequency	
DSO to Substation(s)	Power limit curve	Min. 1 × per year	Curtailment control signal	When a critical grid situ- ation occurs	
Substation(s) to HEMS	Power limit curve	Min. 1 \times per year	Curtailment control signal	When a critical grid situ- ation occurs	
HEMS to Substation(s)	Power values	Min. 1 \times per year	Bus voltages, power values	In every time step	
Substation(s) to DSO	Power val- ues summary per feeder	Min. 1 \times per year	Bus voltages, power val- ues, line and transformer loadings	In every time step	

Limit, which aims at avoiding critical grid situations by reducing the charging powers during times when a critical grid situation, such as line overloadings, is expected. The DSO Access is of a reactive nature as it intervenes to mitigate congestions that already occurred. Additionally, the communications between components is also modeled and evaluated in the system. For the Power Limit, this means that the DSO needs to determine the time-dependent power limit curve based on its knowledge about the average load profile in the grid area and communicate the power limit curve to the HEMS. This can be done once for an entire year or updated more regularly, e.g. based on large deviations from the standard load profile. The latter functionality might lead to more precise results but significantly increases the need for measurements and communication in the real system. As far as the DSO Access is concerned, continuous measurements of the grid state, e.g. bus voltages, and communication between the DSO and HEMS are needed for the strategy. Table 5 gives an overview of the resulting communications, excluding any communication to and from the Grid Calculation component that are needed for simulation purposes only. In addition to helping evaluate how the real system needs to operate to enable the integration of these strategies, the table also allows conclusions regarding the effect of any communication faults. Here, it is assumed that all communication between DSO and HEMS is done with the support of the Distribution Substation(s) but a direct communication is also possible. Generally, the Power *Limit* can be operated by updating the power limit curve only once a year. The advantage of this yearly update is that it does not require much additional communication or measurement infrastructure and is therefore also resilient to any fault occurring in that infrastructure. As opposed to that, constant communication and a large amount of measurements are needed for the implementation of the *DSO Access*. This enables the active curtailment but requires additional implementation effort. It also makes the system prone to communication or measurement faults, so functionalities for dealing with such faults need to be implemented.

The modular simulation tool also enables evaluating the effect of control strategies on power flow results. Figure 8 shows the LV level of the considered grid on the left and the time series results of a critical time frame on an evening in February of the simulated scenario year on the right. The orange markers on the left indicate the location of the corresponding grid elements (transformer: diamond, line: line, bus: circle), where the highest loadings or lowest voltage occurred in that time frame. The results of the application example demonstrate how the simulation tool can aid in the analysis of control strategies on grid elements by providing the possibility to analyse the influence on all grid assets and relevant parameters, either for a summary of all assets or with a more detailed view. The figure also shows that both strategies can improve the grid situation, with the combination of both of them having the largest possible impact. The bus voltage and line loading violations can be completely mitigated by the strategies and even though the transformer loading violation cannot be completely mitigated at all times, it is still reduced notably by at most 30 % in the simulated time frame. As the considered variant of the DSO Access only intervenes after a critical grid situation has already occurred and only curtails the charging powers to 50 % of their initial value, the Power *Limit* manages to achieve more favorable effects here. Since the focus of this paper is the methodological description, a further detailing of the concrete simulation results is not shown here.

Conclusion and evaluation

The modular simulation tool uses pandapower controllers in an innovative way, which ensures the correct simulation sequence and enables the modeling of hierarchies and communication between components without the need for additional message bus packages. The layered component scheme also allows for a structurized implementation of control functionalities. Combining these capabilities creates a versatile and powerful



Fig. 8 Grid topology with critical elements (orange) and corresponding time series results

tool for modeling control strategies for prosumers and for performing simulations with them. In addition to allowing analyses regarding the effect of control strategies on power flow results, it also facilitates evaluating the behavior of energy system components, necessary communications and measurements and any faults that might occur. Moreover, its system architecture that inherently takes care of the correct simulation sequence and communications, enables an efficient implementation process that can also be used for pre-development and testing of control approaches prior to their use in other environments (e.g. lab tests). Despite allowing a straightforward implementation of control strategies and efficient time series simulations, the simulation scope is still fairly limited in terms of the amount of different scenario configurations that can be considered. Different approaches are needed to determine critical grid situations with and without control strategies for a large number of different grid configurations when this information is to be used as the basis for robust grid planning results. Grid planning approaches that enable the consideration of control strategies, either in time-series-based (Schoen et al. 2021b) planning or simultaneity-factor-based (Schoen et al. 2023) planning are well-suited for this purpose. The operational analyses with the modular system and both grid planning approaches together enable a holistic simulation-based analysis. This is vital for developing and analysing control strategies but should be followed by lab and/or field tests before their real-life application. This enables the consideration of additional aspects such as communication protocols and data models, measurements and real user interaction. As this is out of the scope for the modular simulation tool, non-sequential simulations cannot be performed with the modular simulation tool and results cannot be directly implemented into lab testing equipment or used in field tests. However, it can immensely aid in the development of control algorithms and other relevant functionalities that can then be efficiently integrated into systems such as OpSim or beeDIP, which can fulfill the aforementioned tasks. Despite these limitations, the modular simulation tool is a powerful tool for developing and analysing control strategies for prosumers in distribution grids. It builds on the functionalities of the pandapower control module and applies it in a way that allows for the efficient implementation and testing of control strategies regarding various aspects. These developments can then be further used in other environments as well. Additionally, it enables detailed operational analyses in the system itself. The influence on grid parameters can be studied in simulations and different evaluations regarding resilience in the power system, available communication infrastructure and measurements are also possible.

Outlook

In the project Ladeinfrastruktur 2.0, the modular simulation tool is used in the multi-level analysis of control strategies for electric vehicles (Fraunhofer IEE 2023b). The modular simulation tool can be extended by any additional Python-based components or functionalities. Neural-network-based state estimation can already be included in some strategies (Liu et al. 2022). The tool could also further be supplemented by also integrating forecasting functionalities. This could include the process of forecasting data delivery, interpretation and the resulting determination of control signals to be communicated to different prosumers. While the system is not directly related to the Smart Grid Architecture Model (SGAM), it can be used to model and analyze use cases related

to it, especially concerning the function, information and component layer (CEN-CENELEC-ETSI Smart Grid Coordination Group 2023). If other types of tools are to be used together with the implemented control strategies, the pre-developed and tested control behavior can be integrated into co-simulation tools like OpSim, where further investigations are possible, e.g. the interaction with real-time power simulators, eventbased simulations or distributed simulation across several locations (Fraunhofer IEE 2023a). Especially in time-critical applications, the possibility to also include the effect of communication times and delays might be an interesting field for further applications of the modular simulation tool, similar to the project EriGrid 2.0 (Montoya et al. 2018). These aspects will also be investigated in the new project ReCODE, in which cosimulation frameworks are used to couple electric grid simulations with communication network simulators. In terms of grid operation and operational planning, more timecritical applications such as curative congestion management currently are discussed and under investigation. Similarly, the coordinated control and flexibility provision for upstream networks might be a research field in which the modular simulation tool could be applied in. Additionally, the definitely needed level of detail in modelling and simulation (e.g. communication delays etc.) in order to get a certain result in operational planning or long term planning studies has to be evaluated, which is the main research guestion of the project MotiV (Fraunhofer IEE 2023c; Mende et al. 2022).

Acknowledgements

The authors would like to thank their colleagues at the Fraunhofer IEE and the University of Kassel, especially Frank Marten, for the fruitful discussions regarding the modular simulation tool and their helpful insights.

About this supplement

This article has been published as part of Energy Informatics Volume 6 Supplement 1, 2023: Proceedings of the 12th DACH+ Conference on Energy Informatics 2023. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-6-supplement-1.

Author contributions

AS and JR worked on the implementation of the modular simulation tool. AS wrote the manuscript with the help of JR, DM and MB.

Funding

This work presents results of the project Ladeinfrastruktur 2.0 which is funded by the German Federal Ministry for Economic Affairs and Climate Action (formerly known as the German Federal Ministry for Economic Affairs and Energy) under grant no. "0350048A", based on a decision of the Parliament of the Federal Republic of Germany. It also presents selected developments of the project Motiv, which is funded under grant no. "03El1023A/B".

Availability of data and materials

The application example is based on grid and profile data from the SimBench project. More information on the project and the data can be found here: https://simbench.de/en/. The simulation tool implementation uses pandapower, which is available as an open source Python package here: https://www.pandapower.org/. The documentation will be updated with information on how users can create their own modular simulation tool using pandapower.

Declarations

Competing interests

The authors declare that they have no competing interests.

Published: 19 October 2023

References

ADAC (2023) Jeder Fünfte Neuwagen Ist Ein Elektroauto. https://www.adac.de/news/neuzulassungen-kba/ Accessed 26 Mar 2023

BDEW, German Association of Energy and Water Industries: Smart Grid Traffic Light Concept (2023) https://www.bdew. de/media/documents/Stn_20150310_Smart-Grids-Traffic-Light-Concept_english.pdf Accessed 17 Mar 2023 Birk Jones C, Vining W, Lave M, Haines T, Neuman C, Bennett J, Scoffield DR (2022) Impact of Electric Vehicle customer response to Time-of-Use rates on distribution power grids. Energy Reports 8

Bundesnetzagentur: Leitfaden Einspeisemanagement (2023) https://www.bundesnetzagentur.de/DE/Fachthemen/Elekt rizitaetundGas/ErneuerbareEnergien/Einspeisemanagement/start.html Accessed 04 Apr 2023

CEN-CENELEC-ETSI Smart Grid Coordination Group: CEN-CENELEC-ETSI Smart Grid Coordination Group—Smart Grid Reference Architecture. https://energy.ec.europa.eu/system/files/2014-11/xpert_group1_reference_architecture_0. pdf Accessed 19 Jun 2023

Forschungsstelle für Energiewirtschaft e.V. (2023) GridSim-Electric Grid and Energy System Model for Distribution Grids. https://www.ffe.de/en/tools/gridsim-electricity-grid-and-energy-system-model-for-distribution-grids/ Accessed 18 Mar 2023

- Fraunhofer IEE and University of Kassel: pandapower Documentation (2023a) https://pandapower.readthedocs.io/en/ latest/index.html Accessed 03 Mar 2023
- Fraunhofer IEE and University of Kassel: pandapower Control Module (2023b) https://pandapower.readthedocs.io/en/ latest/control.html Accessed 03 Mar 2023
- Fraunhofer IEE and University of Kassel (2023c) SimBench. https://simbench.de/en/ Accessed 25 Mar2023
- Fraunhofer IEE (2023a) OpSim: Test- and Simulation-environment for grid control and aggregation strategies. url:opsim. net/en Accessed 17 Mar 2023
- Fraunhofer IEE (2023b) Ladeinfrastruktur 2.0. https://iee.fraunhofer.de/de/projekte/suche/laufende/ladeinfrastruktur2-0. html Accessed 03 Mar 2023
- Fraunhofer IEE (2023c) MotiV—Modelltiefe in Verteilnetzen. https://www.iee.fraunhofer.de/de/projekte/suche/2020/ motiv.html Accessed 05 Apr 2023
- German Federal Government (2023) "We're Tripling the Speed of the Expansion of Renewable Energies". https://www. bundesregierung.de/breg-de/themen/klimaschutz/amendment-of-the-renewables-act-2060448 Accessed 04 Apr 2023
- Gomes I, Bot K, Ruano MG, Ruano A (2022) Recent techniques used in home energy management systems: a review. Energies 15(8)
- Gümrükcü E, Ahmadifar A, Yavuzer A, Ponci F, Monti A (2023) datafev—a Python framework for development and testing of management algorithms for electric vehicle charging infrastructures. Softw Impacts 15:100467
- Hildermeier J, Burger J, Jahn A, Rosenow J (2023) A review of tariffs and services for smart charging of electric vehicles in Europe. Energies 16(1)
- Liu Z, Ringelstein J, Ernst M, Requardt B, Zauner E, Baumbush K, Wende-von Berg S, Braun M (2022) Monitoring of lowvoltage grids using artificial neural networks and its field test application based on the beeDIP-platform. In: 6th E-Mobility Power System Integration Symposium (EMOB 2022), 2022, 94–99
- Mende D (2022) Modellierung Von Maßnahmen der Leistungsflusssteuerung in Einer Nichtlinearen Mathematischen Optimierung zur Anwendung Im Operativen Engpassmanagement Elektrischer Energieversorgungssysteme: Zugl.: Hannover, Univ., Diss., 2021. Fraunhofer Verlag, Stuttgart
- Mahela OP, Khosravy M, Gupta N, Khan B, Alhelou HH, Mahla R, Patel N, Siano P (2022) Comprehensive overview of multiagent systems for controlling smart grids. CSEE J Power Energy Syst 8(1):115–131
- Mende D, Schmiesing J, Brantl J, Wang H, Mora E, Geiger D, Pape C, Drauz SR, Majidi M, Braun M (2022) Modelltiefe in Verteilnetzen: Szenariobasierte Evaluation des Analyseumfangs und Komplexitätsreduktion für Netzstudien. In: Tagung Zukünftige Stromnetze
- Meinecke S, Sarajlić D, Drauz SR, Klettke A, Lauven L-P, Rehtanz C, Moser A, Braun M (2020) SimBench—a benchmark dataset of electric power systems to compare innovative solutions based on power flow analysis. Energies 13(12)
- Montoya J, Brandl R, Vogt M, Marten F, Maniatopoulos M, Fabian A (2018) Asynchronous Integration of a real-time simulator to a geographically distributed controller through a co-simulation environment. In: IECON 2018—44th Annual Conference of the IEEE Industrial Electronics Society, pp. 4013–4018
- NREL (2023a) Hierarchical Engine for Large-scale Infrastructure Co-Simulation (HELICS). https://helics.org/ Accessed 18 Mar 2023
- NREL (2023b) Charging Infrastructure Technologies: Smart Electric Vehicle Charging for a Reliable and Resilient Grid (RECHARGE). https://www.energy.gov/sites/default/files/2021-06/elt202_bennett_2021_o_5-14_752pm_KS_TM. pdf Accessed 18 Mar 2023
- Pacific Northwest National Laboratory (2023): GridLAB-D -The Next-Generation Simulation Software. https://www.gridl abd.org/ Accessed 20 Jun 2023
- Papadopoulos P, Jenkins N, Cipcigan LM, Grau I, Zabala E (2013) Coordination of the charging of electric vehicles using a multi-agent system. IEEE Trans Smart Grid 4(4):1802–1809
- Plotly Technologies Inc. (2023) Plotly Dash. https://dash.plotly.com/introduction Accessed 19 Mar 2023
- Requardt B (2021) Architekturen und Verfahren für modulare Pilotsysteme und Erweiterungen von Netzleitstellen. PhD thesis, University of Kassel
- Römer C, Hiry J, Kittl C, Liebig T, Rehtanz C (2019) Charging control of electric vehicles using contextual bandits considering the electrical distribution grid. arxiv:1905.01163
- Schoen A, Ringelstein J, Spalthoff C, Braun M (2020) Identifikation und Definition von Betriebsführungsstrategien für die Elektromobilität. In: 16. Symposium Energieinnovation, 12.-14.02.2020, vol. 2021. Graz, Austria
- Schoen A, Ringelstein J, Baumbusch K, Braun M (2021a) Multi-level Analysis of Control Strategies for Electric Vehicles. In: CIRED 2021–the 26th International Conference and Exhibition on Electricity Distribution, 2021, 1470–1474
- Schoen A, Ulffers J, Maschke H, Junge E, Bott C, Thurner L, Braun M (2021b) Considering control approaches for electric vehicle charging in grid planning. In: ETG Congress 2021
- Schoen A, Ulffers J, Maschke H, Mueller L, Braun M (2023) Integrating control strategies for electric vehicles into a simultaneity-factor-based grid planning approach. In: ETG Congress 2023
- The MathWorks, Inc. (2023) MATLAB and Simulink for Microgrid, Smart Grid, and Charging Infrastructure. https://www. mathworks.com/solutions/electrification/microgrid-smart-grid-charging-infrastructure.html Accessed 18 Mar 2023

- Thurner L, Scheidler A, Schäfer F, Menke J-H, Dollichon J, Meier F, Meinecke S, Braun M (2018) pandapower—an open source python tool for convenient modeling, analysis and optimization of electric power systems. IEEE Trans Power Syst 33(6):6510–6521
- TU Dortmund University (2023); Institute of Energy Systems, Energy Efficiency and Energy Economics: agent-based Energy System Modeling and Simulation. https://ie3.etit.tu-dortmund.de/labs-tools/simona/ Accessed 05 Apr 2023
- VDE (2023) VDE-AR-N 4105 Anwendungsregel:2018–11 Generators Connected to the Low-voltage Distribution Network. https://www.vde-verlag.de/standards/0100492/vde-ar-n-4105-anwendungsregel-2018-11.html Accessed 28 Mar 2023
- Vogt M, Marten F, Braun M (2018) A survey and statistical analysis of smart grid co-simulations. Appl Energy 222:67–78 Wenderoth F, Drayer E, Schmoll R, Niedermeier M, Braun M (2019) Architectural and functional classification of smart grid solutions. Energy Informat 2(S1)
- Zhang L, Jabbari F, Brown T, Samuelsen S (2014) Coordinating plug-in electric vehicle charging with electric grid: valley filling and target load following. J Power Sour 267:584–597

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[™] journal and benefit from:

- Convenient online submission
- ► Rigorous peer review
- Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at > springeropen.com