Flexible Distribution Grid Demonstrator (FLEDGE): Requirements and Software Architecture

Sebastian Troitzsch[1*], Sarmad Hanif[1†], Kai Zhang[1‡], Andrej Trpovski[1§] and Thomas Hamacher[2¶]
[1]TUMCREATE, Singapore, [2]Technical University of Munich (TUM), Garching, Germany
Email: *sebastian.troitzsch@tum-create.edu.sg, †sarmad.hanif@tum-create.edu.sg, ‡kai.zhang@tum-create.edu.sg,
§andrej.trpovski@tum-create.edu.sg, ¶thomas.hamacher@tum.de

***Abstract*—In the distribution grid, flexible resources are becoming an important tool to address challenges from uncertain renewable generation and increasing peak loads. To this end, there exists a need for a dedicated software framework to simulate the active distribution grid operation with flexible resources. The Flexible Distribution Grid Demonstrator (FLEDGE) is presented as a simulation tool which integrates active distribution grid operation with classic power flow studies. This paper outlines the requirements, software architecture and key capabilities of FLEDGE.**

Keywords*—Distribution Grid Simulation, Optimal Power Flow, Distribution Grid Operation, Distribution Grid Planning.

I. Introduction

Traditionally, the distribution grid is considered as a passive element of the power system. This passivity stems from the fact that the traditional distribution grid faced unidirectional power flow from source to the loads. In the modern distribution grid, distributed energy resources (DERs) such as renewable generation and electric vehicle (EV) charging are challenging these traditional premises. Uncertain renewable generation may cause bidirectional power flow and EV charging peak loads may pose issues such as grid congestion and high voltage drops. At the same time, the installation of flexible resources such as battery energy storage systems (BESSs) and flexible loads is creating opportunities to mitigate these challenges.

The integration of flexible resources into the distribution grid is aided by numerical optimization. For example, the operation of flexible resources can be formulated as an optimization problem for the minimization of the energy cost subject to consumer constraints, with the energy price becoming an implicit control signal [1], [2]. Motivated by such advancements, regulators are looking into the introduction of new market frameworks on the distribution grid level to enable the integration of flexible loads [3].

Traditional open source software tools for distribution grid simulation such as OpenDSS [4] and GridLAB-D [5] focus more towards classical studies, i.e., short-circuit, power quality and load flow analysis [6]. Moreover, an important consideration in these tools is to account for multiphase unbalanced grid conditions, where detailed single, two and three phase grid component models are modeled [7]. Notably, the emphasis of traditional tools is on the detailed modelling of all assets of distributions grid, and hence, the optimal operation of flexible resources based on numerical optimization is not taken into consideration in detail.

In the recent past, there have been some efforts to support active distribution grid operation through traditional software tools. For example, OpenDSS has been deployed for renewable hosting capacity studies [8] and GridLAB-D to simulate a retail market with price-responsive flexible resources [9]. However, both OpenDSS and GridLAB-D do not currently contain dedicated optimization modules. Instead, the formulation of the optimization problem is outsourced to dedicated optimization software frameworks, e.g., YALMIP/MATLAB [10] or Pyomo/Python [11]. Then, the solution of the numerical optimization may be checked through a co-simulation interface of the traditional software tools [4], [5]. As an alternative, MATPOWER [12] does incorporate optimal power flow solution methodologies as well as an extensible optimization framework. However, MATPOWER is restricted to the modelling of balanced grids which is not suitable for the inherently multiphase unbalanced nature of the distribution grid.

This paper outlines the Flexible Distribution Grid Demonstrator (FLEDGE), an open source simulation tool for the integration of flexible resources in the distribution grid. FLEDGE aims to provide (1) the models and optimization algorithms required for studying the active distribution grid operation with (2) consideration for multiphase unbalanced grid conditions. The focus of this paper is to discuss the requirements, software architecture and the capabilities of the software framework. Furthermore, the process flow for processing an exemplary optimal operation problem in FLEDGE is presented.

Notation

Let $\mathbb{R}$ and $\mathbb{C}$ be the domains of real and complex numbers. Lowercase letters x denote scalars $\mathbb{R}^{1\times1}$ or $\mathbb{C}^{1\times1}$, bold lowercase letters $\mathbf{x}$ denote vectors $\mathbb{R}^{n\times1}$ or $\mathbb{C}^{n\times1}$ and bold uppercase letters $\mathbf{X}$ denote matrices $\mathbb{R}^{n\times m}$ or $\mathbb{C}^{n\times m}$. The transpose of a vector or matrix is $()^\intercal$ and the complex conjugate is $\overline{()}$.

II. Requirements and Software Architecture

A. Preliminaries

Expert users and non-expert users are distinguished as stakeholders. Expert users are assumed to be trained with the software such that they are able to define test cases and scenarios, whereas non-expert users may be interested in the result of a scenario analysis without comprehending each asset of the distribution grid.

Optimization problems are distinguished into planning and operation problems. Optimal planning problems aim to determine the design, sizing or expansion plan for the distribution grid [13], whereas optimal operation problems focus on deriving the operation schedules, control signals or set points for flexible resources [1].

B. Problem statement

In the advent of DERs, the distribution system operator (DSO) seeks to study the deployment of flexible resources in the distribution grid and the the benefit of cost-optimal planning and operation methodologies to manage its assets. Based on this analysis, the DSO may plan its future investments and day-to-day operations.

C. Requirements

To make FLEDGE practically and scientifically relevant, the key requirements were identified through 1) interaction with government agencies, companies and research entities in the distribution grid sector in Singapore 2) and conducting an in-depth domain analysis, i.e., evaluating existing software and identifying relevant gaps required for operating active distribution grid. Furthermore, the requirements were also tuned by 3) prototyping, i.e., a working example was set up to identify key challenges. The requirements are grouped into functional and non-functional, i.e, qualitative, requirements.

1) Functional requirements:

- The software computes planning and operation problems for the distribution grid in the presence of DERs.
- The software comprises a framework to represent and compare different (numerical optimization) problem formulations for both planning an operation. This includes the ability to specify different objective functions, mathematical component models and constraints.
- The software can interface numerical optimization solvers as a tool for computing the planning and operation problems.
- The software comprises mathematical component models for the distribution grid, fixed loads, flexible loads, BESSs, EV charging and renewable generation.
- The software allows adding more mathematical component models at a later stage.
- The software allows computation of results on a scenario basis. A scenario defines the test case, problem type, problem formulation and solution parameters. The test case defines the distribution grid along with its connected DERs. The problem type is either planning or operation. The problem formulation describes which objective function, mathematical component models and constraints are being considered. Solution parameters define which solution algorithm is applied.
- The expert user is provided with an interface for a tabular input format to define scenarios along with the required test case and component model parameters.
- The expert user is provided with programmable interface for instructing the software to compute results for the scenarios from within a high-level programming language.
- The expert user is provided with results for each scenario in a tabular output format comprising the time series data of all state variables of all components.
- The non-expert user is provided with the ability select and view results for scenarios which have been defined by expert users.
- The non-expert user is provided with results for each scenario in a graphical user interface through visual representation of the distribution grid and its operational performance, i.e. nodal voltage, line loading, line losses and dispatched load and generation.

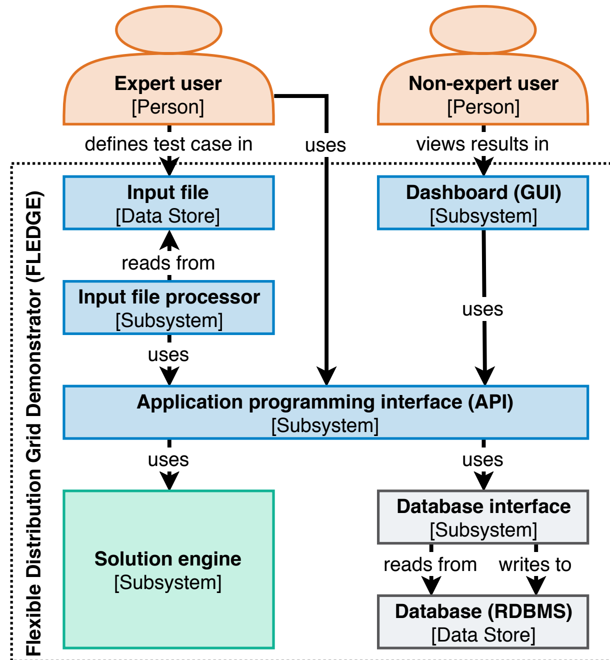

Fig. 1. General architecture

2) Non-functional requirements:

- The solution mechanism is highly scalable, i.e., distributed optimization techniques [14] are utilized to improve the computational performance for large-scale test cases.
- The solution mechanism accounts for multiphase unbalanced loading conditions in the distribution grid.
- The framework specifies a tabular input definition format for test case data, i.e., distribution grid layout, grid component parameters, DER connection points and DER parameters.
- The definition of the test case is separated from the mathematical definition of component model such that the test case is easily replaceable.
- The software makes it easy to reconfigure and re-run scenarios. Hence, the performance of the solution procedure is a major concern.
- The software should not depend on third-party software that would incur additional license fees.
- The software documentation enables expert users to comprehend the mathematical foundation of models and algorithms.
- The software documentation enables non-expert users to operate the software independently given a set of predefined scenarios.

D. Software architecture

1) General architecture: Based on the requirements, the fundamental subsystems of FLEDGE are defined in fig. 1. The input file is the interface for expert users to define test cases and scenarios. The dashboard is the graphical user interface for non-expert users to view the scenario results. The application programming interface (API) is the interface which enables the expert user to interact with FLEDGE. The test cases, scenarios and results are stored in a relational database managment system (RDBMS) which is integrated through the database interface. The solution functionality is bundled into the solution engine, which comprises all mathematical models and solution algorithms.

The API provide methods for reading and updating the test case, scenario and result data as well as methods for invoking the solution engine. It provides a layer of abstraction to ensure that changes in one subsystem don't incur changes in others, e.g., the database interface can be changed without having to change the solution engine. Consequently, the API must be relatively stable throughout the software development to maintain compatibility.

The input file is imported to the database through the input file processor which interfaces with the API to import the test case and scenario data. The input file format may be changed by amending this processor, independent from the internal database structure. The dashboard is is the graphical user interface to present the non-expert user with scenario results. This subsystem also relies on API functions to load the results data. With this architecture, additional user interfaces can easily be deployed at a later point by interfacing the API.

A RDBMS is chosen as the internal storage solution, because it enables more efficient reading, writing, updating interactions then a file system. A persistent storage solution is preferred over in-memory storage to enable reviewing the results at a later time without repeatedly invoking the computationally intensive solution engine. Furthermore, the database can easily be centralized to allow concurrent access across multiple machines, e.g. to allow collaborative work on the test cases and scenarios.

2) Solution engine architecture: The solution engine is the core subsystem which is concerned with implementing the mathematical routines for solving planning and operation problems as in fig. 2. At the top level, the classes for planning and operation problems are distinguished, as both problem types yield a different organization of the optimization problem. These classes construct the respective problem based on subordinate classes for the electric grid model and DER models as well as optimization solvers and simulation solvers. The operation problem class further considers a class for aggregators to adequately model the separated concerns between the DSO and aggregators [1].

The electric grid model classes implement the methods for deriving the power flow approximations for the optimization problem. Similarly, DER model classes provide methods for deriving the mathematical models for renewable generation, EV charging, flexible loads, fixed loads and BESSs. DER models take the form of 1) time series models for non-controllable assets and 2) state space models in the case of flexible resources. In the operation problem, DER model instances are not created in the central problem, but rather within the aggregator subproblem.

Optimization solver classes either 1) directly implement the optimization algorithm or 2) provide the interface to an existing solver, e.g., CPLEX or Gurobi.

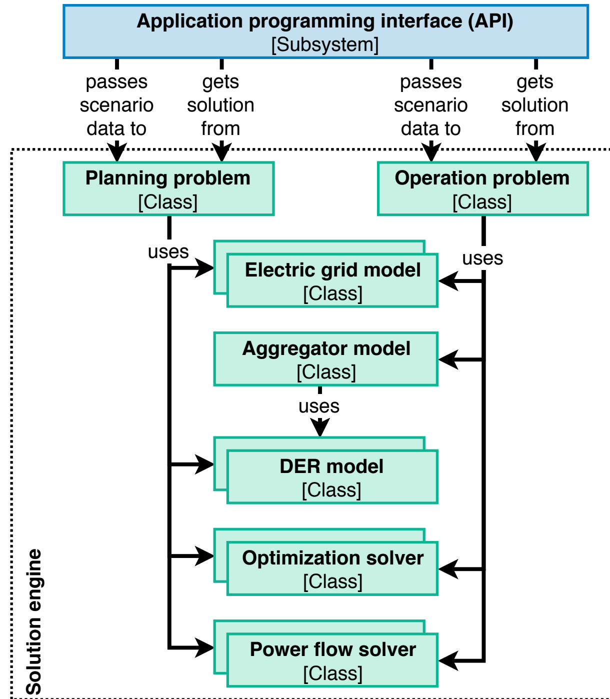

Fig. 2. Solution engine architecture

Power flow solver classes implement solvers for the exact equations of the distribution grid power flow. The exact power flow solution is desired as a benchmark for the optimal power flow solution which was obtained with an approximate model. These classes may interface existing power flow solvers, e.g., OpenDSS.

III. Use case: Operation problem

Figure 3 describes the process flow for an exemplary operation problem in FLEDGE. The input files for defining the test case and scenario have already been imported to FLEDGE and the program is invoked by a call to the API seeking the solution of the scenario. First, the test case and scenario data is loaded through the database interface and an object of the operation problem class is instantiated. The test case and scenario data is passed to the operation problem during creation. Within the operation problem, the electric grid model, aggregator and DER model objects are instantiated, where each DER model object is associated with one aggregator. Once all models have been derived, the optimization problem is constructed and passed to the optimization solver. Lastly, the solution of the optimization problem is used to calculate the exact power flow. All results are stored in the database.

A. Electric grid model

To model the electric grid, first the nodal admittance matrix $\mathbf{Y}$ and incidence matrix $\mathbf{H}$ are constructed for the power delivery assets, i.e., lines and transformers, according to [7], [15]. To this end, the power flow can be solved, i.e., finding the nodal voltages based on the nodal loading. Considering a fixed root node voltage $\mathbf{v}_0$, the no-load node voltages are computed by $\mathbf{w} := \mathbf{Y}_{LL}^{-1}\mathbf{Y}_{L0}\mathbf{v}_0$, where $\mathbf{w}$ is the no-load nodal voltage vector excluding the root node, $\mathbf{Y}_{LL}$ is the

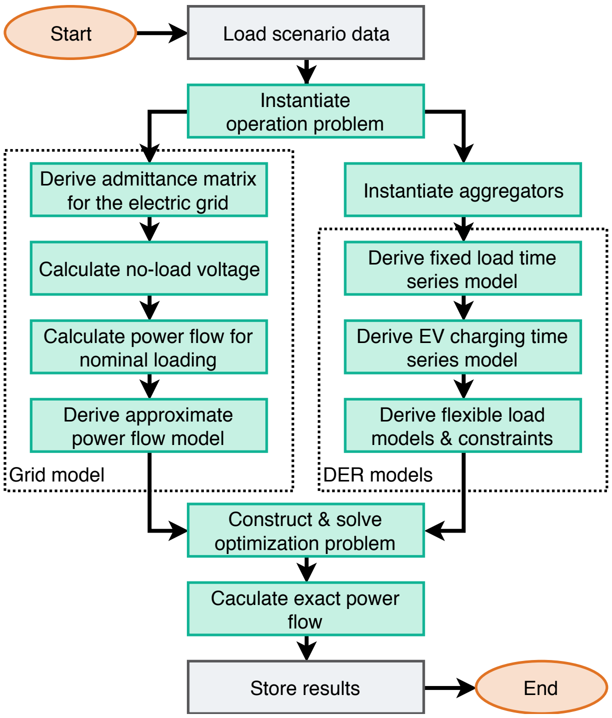

Fig. 3. Operation problem process flow

nodal admittance matrix excluding the root node and $\mathbf{Y}_{L0}$ is the nodal admittance matrix representing the connection from the root node towards all other nodes. The voltage solution for nominal loading $\hat{\mathbf{v}}_n$ is then computed by applying iteratively [16]:

$$\mathbf{v}_n^{(k+1)} = \mathbf{w} + \mathbf{Y}_{LL}^{-1}\left(\mathrm{diag}(\overline{\mathbf{v}}_n^{(k)})^{-1}\overline{\mathbf{s}}_n^{\mathrm{Y}} + \mathbf{H}^{\intercal}\mathrm{diag}(\mathbf{H}\overline{\mathbf{v}}_n^{(k)})^{-1}\overline{\mathbf{s}}_n^{\Delta}\right) \tag{1}$$

with $\hat{\mathbf{v}}_n = \lim_{k\to\infty}(\mathbf{v}_n^{(k)})$ and $\mathbf{v}_n^{(0)} = \mathbf{w}$. The vectors $\mathbf{s}_n^{\mathrm{Y}}$ and $\mathbf{s}_n^{\Delta}$ denote the nominal nodal loading of wye-connected and delta-connected loads.

A state-of-the-art convex approximation of the power flow is sought for the optimization to enable the use of computationally efficient convex optimization solvers. To this end, the approximate power flow model is given as [17]:

$$\begin{aligned} \mathbf{z} &= \mathbf{M}\mathbf{s} + \mathbf{c} \\ \mathbf{z}_{min} &\leq \mathbf{z} \leq \mathbf{z}_{max} \end{aligned} \tag{2}$$

where the sensitivity matrix $\mathbf{M}$ and the constant term $\mathbf{c}$ are constructed according to [17] as a function of admittance matrix $\mathbf{Y}$, incidence matrix $\mathbf{H}$ and nominal voltage solution $\hat{\mathbf{v}}_n$. The power flow properties are collected into $\mathbf{z}$, i.e, the nodal voltages and line flows, and the nodal loading is collected into $\mathbf{s} = \left[\mathrm{Re}(\mathbf{s}^{\mathrm{Y}})^{\intercal}, \mathrm{Im}(\mathbf{s}^{\mathrm{Y}})^{\intercal}, \mathrm{Re}(\mathbf{s}^{\Delta})^{\intercal}, \mathrm{Im}(\mathbf{s}^{\Delta})^{\intercal}\right]^{\intercal}$. The vectors $\mathbf{z}_{min}$, $\mathbf{z}_{max}$ are the power flow constraints.

B. DER models

1) Fixed load model: The fixed load model takes the form of a load profile time series model.

2) EV charging model: EV charging is considered a non-controllable asset and, hence, also takes the form of a load profile time series model. However, FLEDGE in principle allows representing vehicle to grid (V2G) by extending the EV charging class implementation.

3) Flexible load model: The flexible load model takes the form of a state space equation along with appropriate resource constraints. Written in augmented state space form as in [1], the flexible model is:

$$\begin{aligned} \mathbf{y} &= \mathbf{C}\mathbf{x}_0 + \mathbf{D}_u\mathbf{u} + \mathbf{D}_v\mathbf{v} \\ \mathbf{y}_{min} &\leq \mathbf{y} \leq \mathbf{y}_{max} \end{aligned} \tag{3}$$

Where $\mathbf{y}$, $\mathbf{x}_0$, $\mathbf{u}$ and $\mathbf{v}$ are the augmented output, initial state, input and disturbance vectors. The matrices $\mathbf{C}$, $\mathbf{D}_u$, $\mathbf{D}_v$ are the augmented output, input feedthrough and disturbance feedthrough matrices. The vectors $\mathbf{y}_{min}$, $\mathbf{y}_{max}$ are the output constraints of the flexible resource.

C. Optimization problem

FLEDGE considers the separation of concerns between the DSO and the multiple aggregators according to [1]. On the one hand, the aggregators are responsible for the operation of DERs, which includes the optimal dispatch of flexible resources subject to their operation constraints. On the other hand, the DSO is responsible for the reliable operation of the distribution grid. This separation can be realized with a distributed market scheme with the distributed locational marginal price (DLMP) as a control signal. In this scheme, each aggregator first solves its optimal operation problem:

$$\begin{aligned} \min_{\mathbf{u}_f} \quad & \boldsymbol{\lambda}_f^{\intercal}\mathbf{u}_f\Delta t & \forall f \in F_{agg} \quad & \text{(4a)} \\ \text{s.t.} \quad & \mathbf{y}_f = \mathbf{C}_f\mathbf{x}_{0,f} + \mathbf{D}_{u,f}\mathbf{u}_f + \mathbf{D}_{v,f}\mathbf{v}_f & \forall f \in F_{agg} \quad & \text{(4b)} \\ & \mathbf{y}_{min,f} \leq \mathbf{y}_f \leq \mathbf{y}_{max,f} & \forall f \in F_{agg} \quad & \text{(4c)} \end{aligned}$$

Where $\boldsymbol{\lambda}_f$ is the DLMP and F_{agg} is the set of flexible loads of this aggregator.

The aggregator submits the optimal load schedules $\mathbf{u}_f$ to the DSO. Then, the DSO calculates the subgradient $\mathbf{s}$ according to [1] as a function of the load schedules $\mathbf{u}$, the approximate power flow model $\mathbf{M}$, $\mathbf{c}$ and constraints $\mathbf{z}_{min}$, $\mathbf{z}_{max}$. If there exists any overutilization of the distribution grid, i.e., $\mathbf{s} > 0$, with the load schedules $\mathbf{u}$, the DLMP $\boldsymbol{\lambda}$ is updated as a function of the subgradient $\mathbf{s}$. The DSO iteratively requests the aggregators to recalculate their load schedules until the overutilization is removed.

D. Power flow solver

To validate the solution of the optimization problem which uses the approximated power flow model, FLEDGE invokes an exact power flow solver in the final solution step. For this purpose, the simulation solver class utilizes the fixed point equation which was introduced in eq. (1). This method is computationally advantageous for its ability to provide sufficient conditions of the existence of the power flow solution [16]. Eventually, the exact power flow solution is stored in the database along with the DERs' dispatch schedules.

IV. Example: Singapore Hougang

The prototype version of FLEDGE is used to study an distribution grid operation scenario for a test case in the Hougang district of Singapore. The distribution grid test case

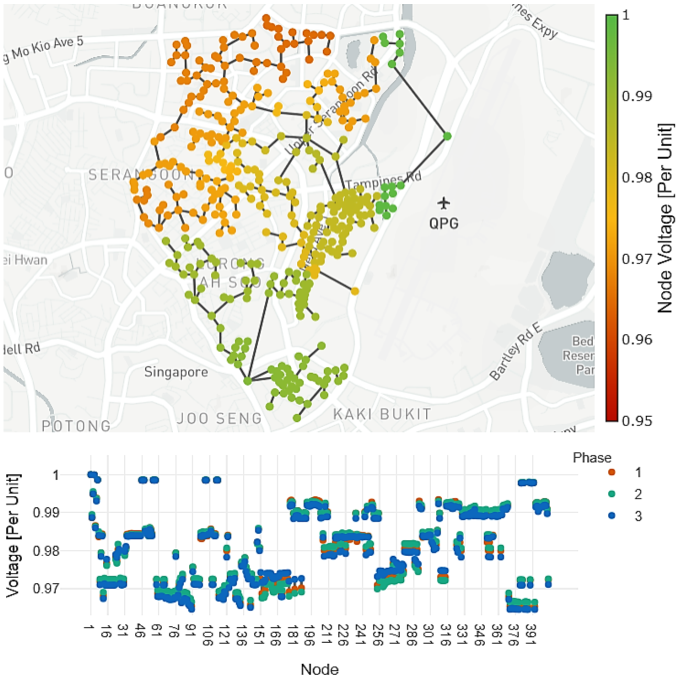

Fig. 4. Node voltage visualization

for the Hougang district in Singapore is derived through the work on synthetic distribution grid modelling by [13]. Fixed load time series data is derived from household surveys and smart meter readings of commercial buildings and industrial consumers. A time series model for private EV charging is derived from a probabilistic agent-based mobility simulation of EVs in Singapore [18]. Electric public bus charging demand is obtained from an energy model for electric buses [19] based on bus arrival and departure data. In this test case, heating ventilation and air-conditioning (HVAC) systems are considered as flexible loads.

Figure 4 highlights the node voltage visualization for the Hougang test case. In upper portion, the grid map depicts the geographic location and average per unit voltage of the grid nodes. Below, the phase per unit voltages for each node are depicted. The selected view captures the conditions at 12 a.m., where are high voltage drop is observed for areas with residential developments due to private EVs charging over night at residential carparks, whereas areas with a high share of commercial buildings do not experience a significant voltage drop.

V. Conclusion

The Flexible Distribution Grid Demonstrator (FLEDGE) is being developed by TUMCREATE as an extensible software tool for optimal planning and operation of distribution grids. FLEDGE incorporates classical distribution grid simulation capabilities with numerical optimization techniques which are suggested for the integration of flexible resources such as flexible loads and BESSs. This paper documents the requirements and software architecture envisioned for FLEDGE and lays the foundation for the further development as an open source tool.

Acknowledgment

This work was financially supported by the Singapore National Research Foundation under its Campus for Research Excellence And Technological Enterprise (CREATE) programme.

References

[1] S. Troitzsch, S. Hanif, and T. Hamacher, “Distributed Robust Reserve Scheduling in Congested Distribution Systems,” in *IEEE PES Gen. Meet.*, Portland, OR, USA, 2018.

[2] S. Hanif, K. Zhang, C. Hackl, M. Barati, H. B. Gooi, and T. Hamacher, “Decomposition and Equilibrium Achieving Distribution Locational Marginal Prices using Trust-Region Method,” *IEEE Trans. Smart Grid*, 2018.

[3] B. Shen, G. Ghatikar, Z. Lei, J. Li, G. Wikler, and P. Martin, “The role of regulatory reforms, market changes, and technology development to make demand response a viable resource in meeting energy challenges,” *Appl. Energy*, vol. 130, pp. 814–823, 2014.

[4] R. C. Dugan and D. Montenegro, *The Open Distribution System Simulator (OpenDSS) - Reference Guide*. EPRI Electric Power Reserach Institute, 2018.

[5] D. P. Chassin, K. Schneider, and C. Gerkensmeyer, “GridLAB-D: An open-source power systems modeling and simulation environment,” in *IEEE PES Transm. Distrib. Conf. Expo.*, 2008.

[6] D. Recalde, A. Trpovski, S. Troitzsch, K. Zhang, S. Hanif, T. Hamacher, and C. Way, “A Review of Operation Methods and Simulation Requirements for Future Smart Distribution Grids,” in *IEEE PES Innov. Smart Grid Technol. Conf. Asia*, 2018.

[7] W. H. Kersting, *Distribution System Modeling and Analysis*. CRC Press, 2001.

[8] M. Rylander, C. Trueblood, W. Sunderman, S. Coley, J. Smith, and T. Key, “Smart Inverter Modeling Results, Variability Analysis, and Hosting Capacity Beyond Thresholds,” Electric Power Research Institute, Tech. Rep., 2013.

[9] J. C. Fuller, K. P. Schneider, and D. Chassin, “Analysis of residential demand response and double-auction markets,” in *IEEE PES Gen.Meet.*, July 2011.

[10] J. Lofberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” in *IEEE Int. Conf. Robot. Autom.* IEEE, 2004, pp. 284–289.

[11] W. E. Hart, C. D. Laird, J.-P. Watson, D. L. Woodruff, G. A. Hackebeil, B. L. Nicholson, and J. D. Siirola, *Pyomo - Optimization modeling in Python*. Springer, 2017.

[12] D. Z. Ray, E. M. S. Carlos, and J. T. Robert, “Matpower: Steady-state operations, planning, and analysis tools for power systems research and education,” *IEEE Trans. Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[13] A. Trpovski, D. Recalde, and T. Hamacher, “Synthetic Distribution Grid Generation Using Power System Planning: Case Study of Singapore,” in *Int. Univ. Power Eng. Conf.*, 2018.

[14] K. Zhang, S. Hanif, and D. Recalde, “Clustering-based decentralized optimization approaches for dc optimal power flow,” in *IEEE PES Innov. SmartGrid Technol. Conf. Asia*, Dec 2017.

[15] M. Bazrafshan and N. Gatsis, “Comprehensive modeling of three-phase distribution systems via the bus admittance matrix,” *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2015–2029, 2018.

[16] A. Bernstein, C. Wang, E. Dall’Anese, J.-Y. Le Boudec, and C. Zhao, “Load-flow in multiphase distribution networks: Existence, uniqueness, non-singularity, and linear models,” *IEEE Trans. Power Syst.*, 2018.

[17] S. Hanif, M. Barati, A. Kargarian, H. Gooi, and T. Hamacher, “Multiphase Distribution Locational Marginal Prices: Approximation and Decomposition,” in *IEEE PES Gen. Meet.*, 2018.

[18] M. Huber, A. Trippe, P. Kuhn, and T. Hamacher, “Effects of large scale EV and PV integration on power supply systems in the context of Singapore,” *IEEE PES Innov. Smart Grid Technol. Conf. Eur.*, 2012.

[19] M. Gallet, T. Massier, and T. Hamacher, “Estimation of the energy demand of electric buses based on real-world data for large scale public transport networks,” *Appl. Energy*, vol. 230, pp. 344–356, 2018.

View publication stats