CIGRE-300 **2020 CIGRE Canada Conference**
Toronto, Ontario, October 19-22, 2020

GLOW: A Novel Co-simulation Framework for Integrated Distribution Network Planning and Data Analysis using GridLAB-D

Yanzhu Ye, Bo Yang, Panitarn Chongfuangprinya
Hitachi America, Ltd.
Santa Clara, CA, USA

SUMMARY

In recent years, energy regulations, environmental policies, and technology advances have driven a significant rise in the penetration level of distributed energy resources (DERs) in the distribution system, including solar photovoltaic panels (PVs), energy storage system, electric vehicles (EVs), etc. Such transition brings various challenges to conventional distribution grid planning and operations, e.g., reverse power flow, over-voltage violations, etc. There are increasing needs for advanced tools for comprehensive study of the integration impacts of DERs on distribution networks under various control/modeling practices. These modeling tools not only require network simulation engine with modeling capabilities for complicated distribution grids and novel DERs, but also a flexible and interactive environment for development/testing of advanced analysis/control methodologies.

Currently popular commercial software, i.e. CYME and Synergi, were designed to meet the requirements of traditional distribution resource planning processes, which are not flexible enough to address the arising needs of novel DER modeling and joint simulation with 3rd party analytical tools. In this paper, we selected GridLAB-D as the simulation solver for our framework development and validation, which is an open-source tool developed by PNNL under U.S. Department of Energy [1][2]. A user-friendly co-simulation framework for integrated distribution network planning and analysis is proposed - GLOW (GridLAB-D Open Workspace). GLOW is more than a high-end interactive user interface based on GridLAB-D, which further provides an intuitive and user-friendly open environment for different stakeholders to perform system modeling and DER integration study. GLOW consists of three major layers: simulation layer, data management layer and application layer. The GridLAB-D simulation engine is located on the simulation layer, which interacts with the other two layers to exchange simulation data, to receive and respond to different simulation workflow control, etc. On the data management layer, a flexible data architecture is designed to handle a variety of data format, e.g. GridLAB-D model file, meter data, forecasting data, analysis output data, etc. And the data interactions are coordinated through a collection of well-designed Application Programming Interfaces (APIs). Python is chosen as the

Yanzhu.ye@hal.hitachi.com

programming language for this framework development, considering its intensive third-party libraries and the compatibility with most other languages and platforms. The architecture of GLOW is demonstrated through the implementation of Integration Capacity Analysis (ICA) on distribution network.

*GLOW is funded by CEC (Californian Energy Commission) under grant EPC-17-043.
GLOW is scheduled to be launched online for alpha-test on Sept 2020.

KEYWORDS

GridLAB-D, GLOW (GridLAB-D Open Workspace), Co-Simulation, Open-source, Distributed Energy Resources (DER), ICA (Integration Capacity Analysis)

Yanzhu.ye@hal.hitachi.com

1 INTRODUCTION

In recent years, the high penetration level of distributed energy resources (DERs) and more and more intelligent controllable devices in the distribution system brings various challenges to conventional distribution grid operations. There are increasing needs for advanced distribution system study tools to evaluate the potential impact of these novel DERs on network planning and operations. These modeling and simulation tools not only require a powerful network simulation engine with modeling capabilities for complicated distribution grids and various DER technologies, but also a flexible and open interactive environment for development/testing of advanced analysis/control methodologies.

There have been a variety of distribution network simulation softwares [3]. Most commercial software is proprietary, highly specialized, vertically integrated, which is not flexible enough to address the arising needs of DER modeling and joint simulation with 3rd party analytical tools. While those open-source tools, e.g. GridLAB-D, OpenDSS, etc., are more suitable and flexible for exploring and testing new analysis/control methodologies. In this paper, we selected GridLAB-D as the simulation engine for our co-simulation framework development and validation. GridLAB-D is developed by PNNL under U.S. Department of Energy [1] [2], which provides modeling and simulation for distribution assets, as well as energy markets, building technologies, and other demand-side technologies, etc., but lack of intuitive and user-friendly graphical interface for system modelling and simulation.

In recent years there have been some related works for co-simulation framework development based on GridLAB-D [4]-[8]. In [4]-[6], the Matlab-GridLAB-D co-simulation framework are developed, however, these works focus on limited use cases and selected feeder models, e.g. [4] is for demand response study, , [5] is for microgrid study, and [6] only supports traditional power flow study on IEEE-13, 37 node system. Also, as a proprietary tool, Matlab has its limits on the flexibility and compatibility across different platform. [7] proposes a Python-GridLAB-D framework that focuses dynamic interaction with ongoing GridLAB-D simulation, however, it lacks a flexible infrastructure to develop new use cases. [8] presented a cloud-based GridLAB-D simulation framework, which provides the flexibility to integrate and develop new functions, however it lacks an advanced graphical user interface to provide fully interaction capabilities between users and various back-end features. To further address these challenges and enhance the co-simulation capability between GridLAB-D and other tools, a user-friendly co-simulation framework is proposed in the paper - GLOW (GridLAB-D Open Workspace).

GLOW provides an intuitive and user-friendly open environment for different stakeholders to investigate DER integration impacts on power grid. GLOW features with an advanced interactive graphical user interface, which will argument and replace the CLI (Command-Line Interface) of GridLAB-D for model creation, model modification, simulations execution and control, data processing, etc. Python is chosen as the programming language for this framework development, considering its intensive third-party libraries and compatibility with most other languages and platforms. The proposed GLOW system mainly consists of three major layers: simulation layer, data management layer and application layer. The GridLAB-D simulation engine is located on the simulation layer, which interacts with the other two layers to exchange simulation data, to receive and respond to different simulation workflow control, etc. On the data management layer, a flexible data architecture is designed to handle complex data format, e.g. meter data, utility feeder model, forecasting data, simulation input/output data, etc. The data and control interaction between different layers and 3rd-party tools are coordinated through a well-designed API gateway.

The paper is organized as follows: In Section II, the system architecture and implementation details of GLOW is presented. In Section III, a use case - Integration Capacity Analysis (ICA) is demonstrated based on this proposed co-simulation framework. Finally, section IV concludes the paper.

2 GLOW SYSTEM ARCHITECTURE AND IMPLEMENTATION

This section provides an overview of the multi-layer architecture of GLOW and implementation details of each layer, system data infrastructure, and web-based graphical user interface.

2.1 Layered System Architecture

As shown in Figure 1. , the GLOW framework consists of three major layers: simulation layer, data management layer and application layer. An API gateway is designed and implemented to coordinate data and control interaction among the three layers and 3rd-party systems.

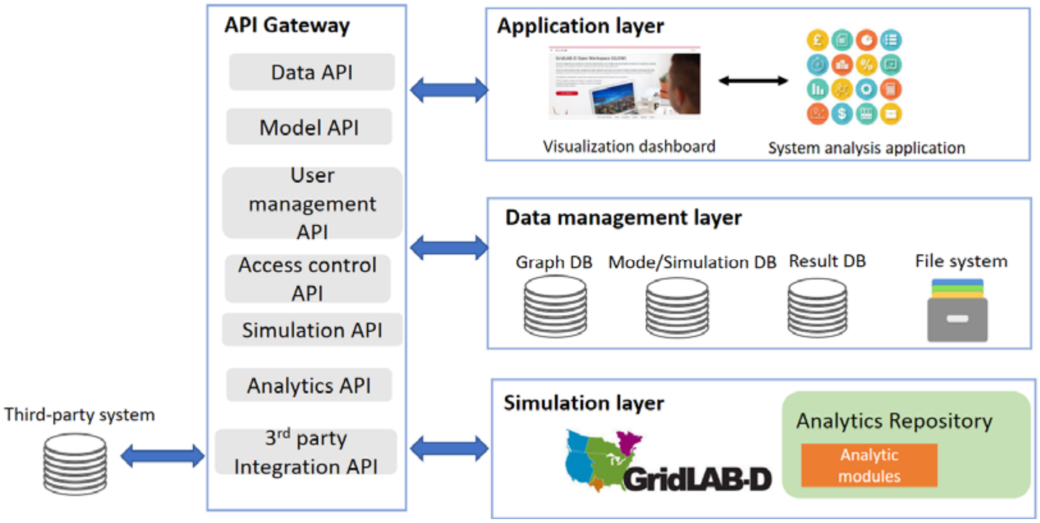


Figure 1. Layered architecture of GLOW framework

(1) Simulation layer

The GridLAB-D simulation engine is located on the simulation layer, which interacts with the other two layers to exchange simulation data (input/output), to receive and respond to different simulation workflow control, etc. As an open-source tool, GridLAB-D provides interfaces to directly access network components, modify simulation settings, control/monitor simulation process, etc. One is through REST (Representational State Transfer) API, the other is through GridLAB-D-Python library. These APIs or libraries are utilized to establish the two-way communication interaction between GridLAB-D engine and external control and analytical function. Besides GridLAB-D simulation engine, other third-party tools and customized analytical modules can also be deployed on simulation layer.

(2) Data management layer

The database management system on this layer is in charge of storing and managing different data types, e.g. system model data, advanced metering data, load profiles, simulation configurations, user information, system analysis results etc. A variety of data format will be collected, converted, processed and then stored onto database. The detailed data processing pipeline and data communication infrastructure are described in detail in Section 2.2.

(3) API Gateway

An API gateway is designed and implemented to coordinate data and control interaction among the three layers and 3rd-party systems. The API gateway controls and manages access to a collection of back-end service, e.g. data query, data processing, model management, user access control, simulation analysis control, etc. One implementation example in GLOW is to use Django REST framework, which is a powerful and flexible toolkit for building Web APIs. In Django, different functionalities are designed and implemented by using the RESTful technology to support the exposition and request of services.

The API gateway serves as the entry point for all API service requests in GLOW, which will make the simulation layer, application layer and data management layers loosely coupled, increase system scalability. Also, through these well-defined APIs, the application of system analytical functions is insulated from how these functions are implemented in detail on the back-end independent of the programming language.

(4) Application layer

Through these well-designed APIs managed on GLOW API gateways, various distribution system analysis application can be developed as individual module and deployed on this open data platform. These applications modules can be organized around specific business goals, user groups, DER types, or individual analytical functions, etc. These application modules can be as simple as time-series power flow simulation, or as complicated as system-wide integration capacity analysis, AI-based system control, etc. Each individual functional module can be wrapped and communicated through well-constructed APIs, so that a complicated application can be easily built up as a suite of small, individual functional modules.

Besides these analytical application modules, GLOW features with a user-friendly graphic interface to dynamically interact with GridLAB-D simulator, access and manage these analytical functions, visualize system analysis output, etc., which is presented in Section. 2.3.

2.2 Data communication infrastructure

In GridLAB-D, the feeder model, simulation set-up and simulation schedules are described using .glm file. This .glm file contains the detailed representation of power grid topology and its asset's properties. The .glm file is the principle input file format for GridLAB-D to run simulation. GridLAB-D also take other file type as input to support simulation, e.g. csv, xml, tmy weather file, etc. The .glm file is relatively easy for human-reading and editing, however, not directly compatible with any major software tools. Currently there is no effective tool for editing and parsing .glm file.

Figure 2 shows the detailed data communication infrastructure in GLOW, including data conversion pipeline, common data structure, etc. In this data infrastructure, the .glm file format will be converted to a common data structure – JSON, which is an open standard file format, and is completely language-independent. It is easy for humans to read and write. And it is also easy for machines to parse and generate. There exists a diverse range of applications, third-party libraries which is compatible with JSON data structure. These features make JSON an ideal data-interchange language.

The flexible model presentation in JSON format coupled with different processing module can further fit different application use cases [9], e.g. the conversion to network graph data structure for advanced graph analysis, etc.

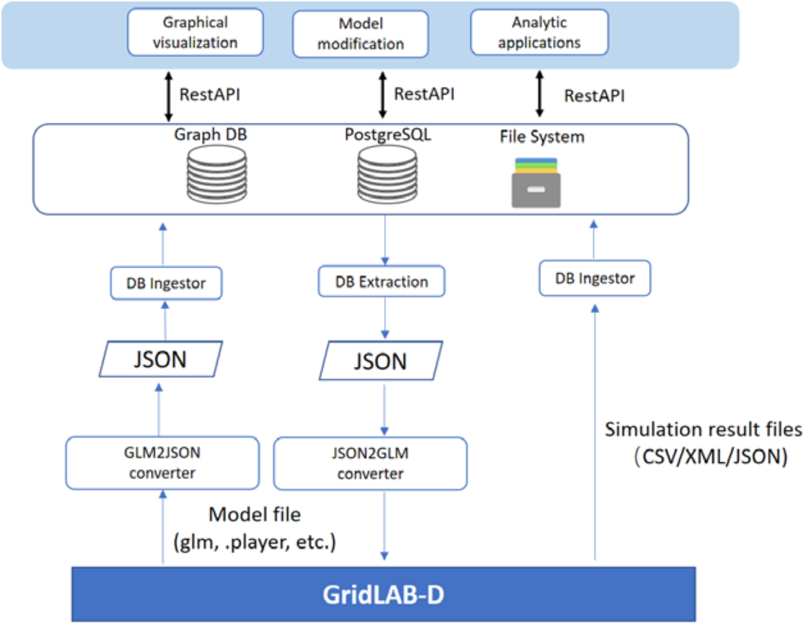


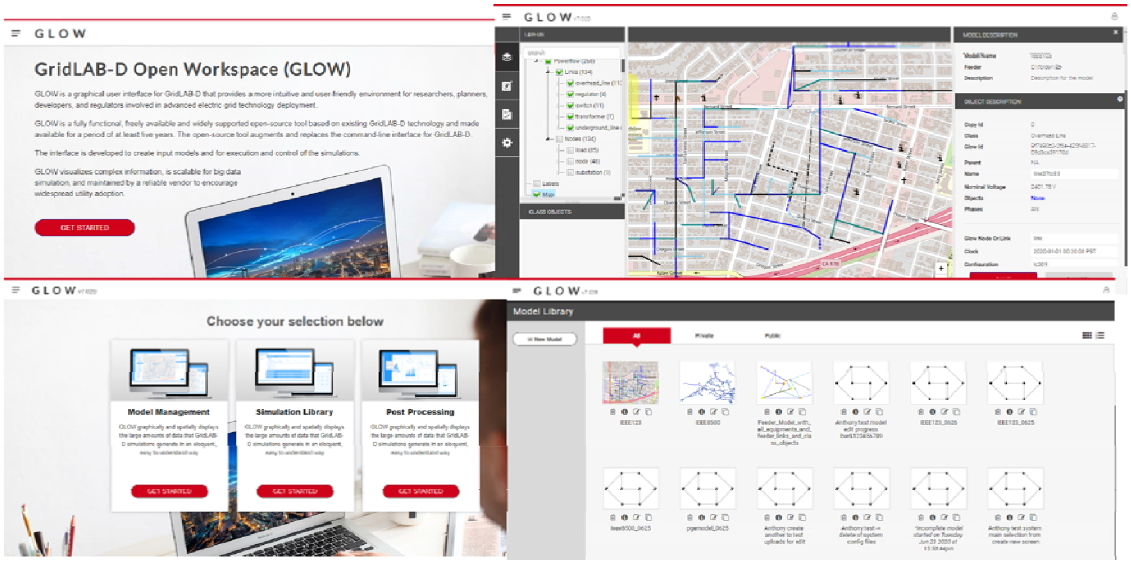
Figure 2. Diagram of data communication infrastructure

2.3 Graphical User Interfaces

Figure 3 demonstrates the graphical web portal of GLOW, which consist of several major graphical interfaces, e.g. model library, simulation management, Viewer, Post-processing, etc. The key functions provided through this web-portal, as shown in Figure 4, includes user authentication, feeder model import wizard, model explorer, simulation management, simulation set-up wizard, etc.

In the model graphical viewer interface, the feeder model will be loaded and represented in the map-view. If the imported models are provided with GIS information, the GUI will directly provide a map-view of the grid. If the models do not contain GIS information, an algorithm will be used to automatically generate artificial geolocation information for each node to layout the grid network. Furthermore, in the viewer interface, the feeder model can be directly modified and updated.

In GLOW framework, some common distribution analysis application modules have been developed and deployed as workflow templates, e.g. time-series power flow simulation, ICA, etc. Then through the simulation set-up wizard, these pre-deployed GLOW distribution analysis applications can be quickly set up, launched and reviewed. The analysis result will be automatically uploaded onto database and further loaded on-demand for viewing or post-processing. In Section 3, one distribution system analysis application – ICA is implemented and demonstrated on this proposed simulation framework.



GridLAB-D Open Workspace (GLOW)

GLOW is a graphical user interface for GridLAB-D that provides a more intuitive and user-friendly environment for researchers, planners, developers, and regulators involved in advanced electric grid technology deployment.

GLOW is a fully functional, freely available and widely supported open-source tool based on existing GridLAB-D technology and made available for a period of at least five years. The open-source tool augments and replaces the command-line interface for GridLAB-D.

The interface is developed to create input models and for execution and control of the simulations.

GLOW visualizes complex information, is scalable for big data simulation, and maintained by a reliable vendor to encourage wide-spread utility adoption.

Choose your selection below

Model Management

GLOW graphically and spatially displays the large amounts of data that GridLAB-D simulations generate in an eloquent, easy to understand way

Simulation Library

GLOW graphically and spatially displays the large amounts of data that GridLAB-D simulations generate in an eloquent, easy to understand way

Post Processing

GLOW graphically and spatially displays the large amounts of data that GridLAB-D simulations generate in an eloquent, easy to understand way

GLOW
GET STARTED
Model Library
New Model
All
Private
Public
IEEE123
IEEE8500
Feeder_Model_with_all_equipments_and_feeder_links_and_class_objects
Anthony test model edit progress bar&123456789
IEEE123_0625
IEEE123_0625
Ieee8500_0625
pgemodel_0625
Anthony create another to test uploads for edit
Anthony test:-> delete of system config files
"Incomplete model started" on Tuesday Jun 23 2020 at 3:50:44pm
Anthony test system main selection from create new screen
CLASS OBJECTS

Figure 3. Illustration of graphical web portal of GLOW

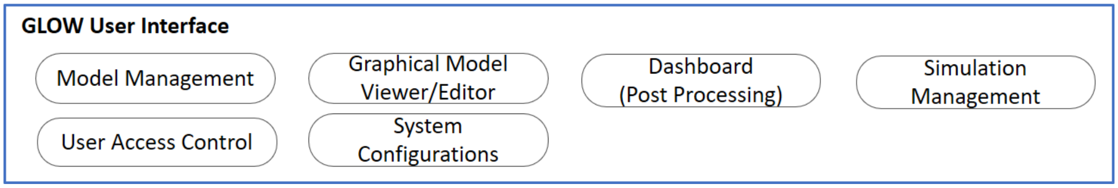


Figure 4. Key functions of GLOW graphical web portal

As an open-source simulation platform, GLOW will be host on AWS for public access at initial testing phase. GLOW is scheduled to be launched online for alpha-test on Sept. 2020.

3 USE CASE – INTEGRATION CAPACITY ANALYSIS (ICA)

In this section, one distribution system analysis application - ICA was implemented on this proposed framework. The "Integration Capacity Analysis" or ICA, is mostly California's term, for what is more commonly known as hosting capacity analysis. This analysis is to quantify the capability of distribution system to accommodate additional DER down to the line section or node level within a variety of power system limitations, e.g. thermal limit, power quality and safety standards, etc.

3.1 ICA Implementation

Typically, there are two methodologies for ICA calculation provided by the CPUC (California Public Utilities Commission) DRP (Demand Response Planning) guidelines [10], streamlined approach and iterative simulation. The streamlined approach is a simplified or abstracted evaluation that uses a set of equations. This approach requires less processing resources and highly inaccurate. The iterative simulation approach requires powerful computing resources through iterative power flow simulation with different placement/up-sizing of DER in the system model. This approach provides higher accuracy due to direct modeling of resources and system simulation.

Our implementation chooses the iterative simulation method for ICA study. The workflow of GLOW ICA is shown in Figure 5. The ICA is a complex modeling exercise that requires various detailed information, e.g. the detailed distribution grid modelling, the type and performance of load on the grid (e.g. load profiles showing maximum and minimum load conditions), the existing DER integration and load control on the grid, etc. The ICA implementation fully evaluates the flexibility of this co-simulation framework through repeatedly interaction between GridLAB-D simulator and external control and analytical functions.

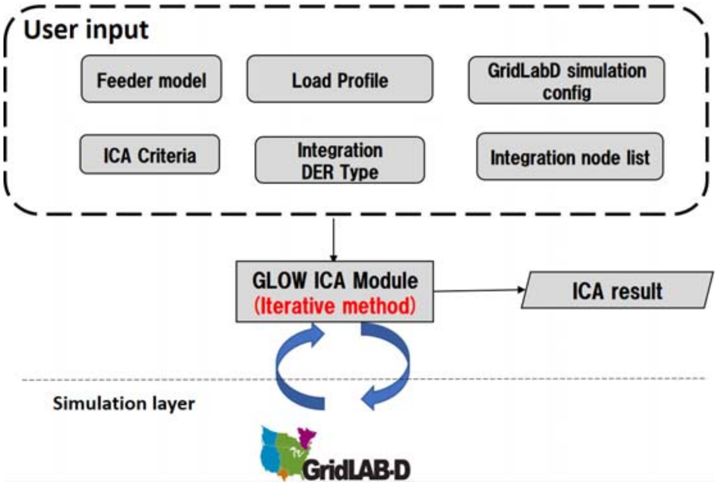


Figure 5. GLOW ICA Workflow

3.2 ICA Workflow Set-up and Results

The GLOW graphical web portal has implemented a friendly simulation set-up wizard, which faciliates the user to set up and launch a new simulation use case quickly, as shown in Figure 6. The ICA analytical module has been developed and deployed on the GLOW framework, users can directly set up a new ICA study case, launch analysis on selected feeder model, monitor and control analysis progress dynamically, and finally the analysis result can be loaded onto GLOW post-processing dashboard for data processing, data visualization and report generation, etc.

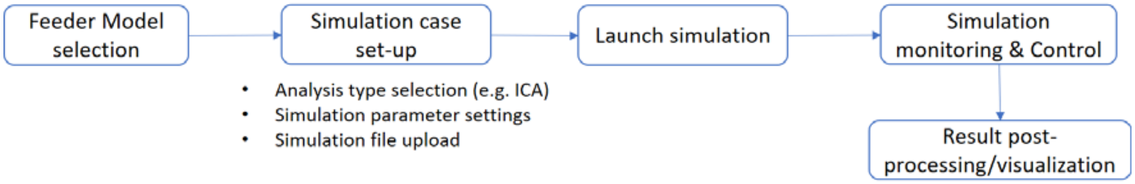

Feeder Model selection
Simulation case set-up
Launch simulation
Simulation monitoring & Control
Result post-processing/visualization
Analysis type selection (e.g. ICA)
Simulation parameter settings
Simulation file upload

Figure 6. Demonstration of simulation set-up wizard in GLOW

The IEEE 123 node feeder model [11] is selected to run the ICA. The power system limitation investigated in this study are listed in Table I.

Table I. Power System Criteria for ICA study

Name	Description	Threshold
Steady state voltage	Voltage at all the nodes should be within certain range	0.95 ~ 1.05 p.u.
Voltage fluctuation	Percentage change in voltage at all nodes when DER is turned ON from OFF condition	< 3%
Thermal rating limit	Line current cannot be over thermal limit per branch.	1000 A

For the generation type DER, the maximum integration capacity for different limiting categories can be easily generated by ICA module, as shown in Figure 7. (a)-(d).

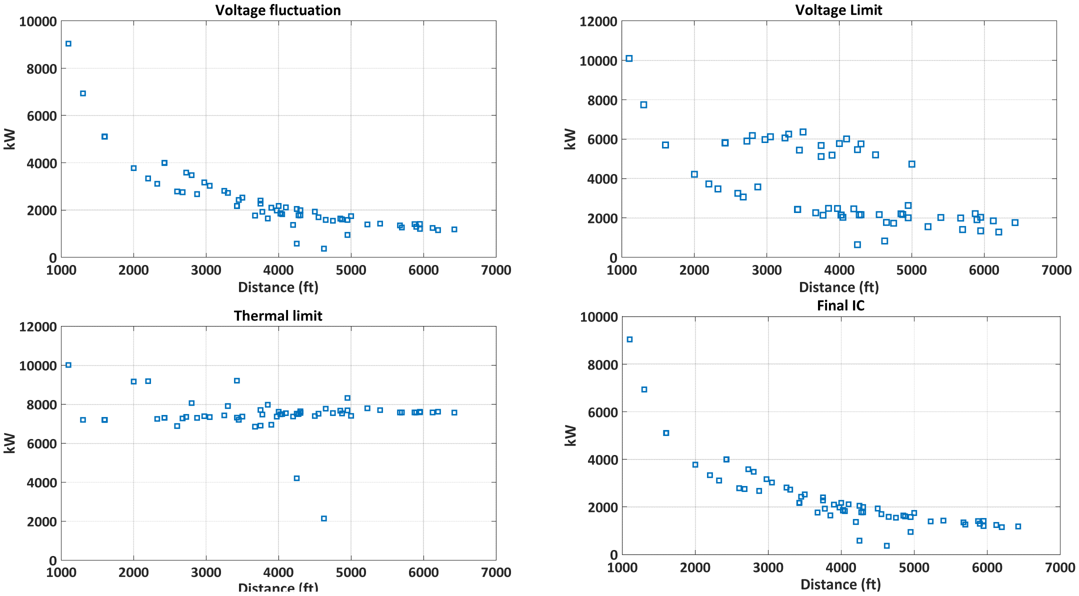


Figure 7. ICA result for generation-type DER integration

4 CONCLUSION

This paper proposed a novel co-simulation framework for integrated distribution system planning and analysis based on GridLAB-D and Python, which is GridLAB-D Open Workspace (GLOW). The proposed GLOW framework has multi-layer structure, including simulation layer, data management layer and application layer. On the simulation layer, GridLAB-D operates as distribution system simulation engine. The data and control interaction among the three layers and 3rd-party tools are coordinated through a well-designed API gateway. The API gateway controls and manages access to a collection of back-end service, e.g. data query, data processing, model management, simulation analysis management, etc. Another key feature of this proposed simulation framework is the advanced web-based graphical user interface, which effectively facilitate the design, testing and experiment of various distribution analysis for different stakeholders.

To demonstrate the implementation approach and system flexibility of the co-simulation framework, one distribution system analysis application – ICA is developed and deployed on this framework. The implementation of the friendly simulation set-up wizard effectively guides users to set up and launch a new simulation use case quickly. In the future, more distribution network analysis applications are expected to be developed and integrated onto this platform, especially leveraging various machine learning technologies, etc.

BIBLIOGRAPHY

[1] GridLAB-D^{TM}, https://www.gridlabd.org/

[2] P.D. Chassin, C.J. Fuller and N. Djilali, "GridLAB-D: An agent-based simulation framework for smart grids," Journal of Applied Mathematics, vol. 2014, pp. 1–12, 2014.

[3] D. W. Gao, E. Muljadi, T. Tian and M. Miller, "Software Comparison for Renewable Energy Deployment in a Distribution Network" Technical Report, 2017

[4] D. Wang *et al*., "A test bed for self-regulating distribution systems: Modeling integrated renewable energy and demand response in the GridLAB-D/MATLAB environment," *2012 IEEE PES Innovative Smart Grid Technologies (ISGT)*, Washington, DC, 2012, pp. 1-7

[5] M. A. Al Faruque and F. Ahourai, "GridMat: Matlab toolbox for GridLAB-D to analyze grid impact and validate residential microgrid level energy management algorithms," *ISGT 2014*, Washington, DC, 2014, pp. 1-5

[6] A. Nasiakou, M. Alamaniotis and L. H. Tsoukalas, "MatGridGUI — A toolbox for GridLAB-D simulation platform," *2016 7th International Conference on Information, Intelligence, Systems & Applications (IISA)*, Chalkidiki, 2016, pp. 1-5

[7] T. M. Hansen, B. Palmintier, S. Suryanarayanan, A. A. Maciejewski and H. J. Siegel, "Bus.py: A GridLAB-D communication interface for Smart distribution Grid simulations," *2015 IEEE Power & Energy Society General Meeting*, Denver, CO, 2015

[8] K. Anderson, J. Du, A. Narayan and A. E. Gamal, "GridSpice: A Distributed Simulation Platform for the Smart Grid," in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2354-2363, Nov. 2014

[9] K. Guddanti, Y.Ye, P. Chongfuangprinya, et al "Better Data Structures for Co-simulation of Distribution System with GridLAB-D and Python", *Accepted by 2020 IEEE Power & Energy Society General Meeting*

[10] ICA (Integration Capacity Analysis), https://www.cpuc.ca.gov/General.aspx?id=5071

[11] IEEE 123 feeder model, https://site.ieee.org/pes-testfeeders/resources/