7-22-2022

Modeling Environment for Testing a Distributed Energy Resource Management System (DERMS) Using GridAPPS-D Platform

Shiva Poudel
Pacific Northwest National Laboratory

Sean Keene
Portland State University

Roshan Kini
Pacific Northwest National Laboratory

Sarmad Hanif
Pacific Northwest National Laboratory

Robert B. Bass
Portland State University, robert.bass@pdx.edu

See next page for additional authors

Citation Details

S. Poudel, S. J. Keene, R. L. Kini, S. Hanif, R. B. Bass and J. T. Kolln, "Modeling Environment for Testing a Distributed Energy Resource Management System (DERMS) Using GridAPPS-D Platform," in IEEE Access, vol. 10, pp. 77383-77395, 2022, doi: 10.1109/ACCESS.2022.3192845.

Authors

Shiva Poudel, Sean Keene, Roshan Kini, Sarmad Hanif, Robert B. Bass, and Jaime Kolln

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 June 2022, accepted 10 July 2022, date of publication 22 July 2022, date of current version 27 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192845

TUTORIAL

Modeling Environment for Testing a Distributed Energy Resource Management System (DERMS) Using GridAPPS-D Platform

SHIVA POUDEL[1], (Member, IEEE), SEAN J. KEENE[2], (Senior Member, IEEE), ROSHAN L. KINI[1], (Member, IEEE), SARMAD HANIF[1], (Member, IEEE), ROBERT B. BASS[2], (Member, IEEE), AND JAIME T. KOLLN[1], (Member, IEEE)

[1]Pacific Northwest National Laboratory, Richland, WA 99354, USA
[2]Department of Electrical and Computer Engineering, Portland State University, Portland, OR 97201, USA

Corresponding author: Shiva Poudel (shiva.poudel@pnnl.gov)

This material is based on work supported by the U.S. Department of Energy's (DOE) Office of Electricity (OE), US DOE OE-0000922. Portland State University (PSU) is developing an Energy Service Interfaces (ESI) for the DOE. Pacific Northwest National Laboratory (PNNL) provided training, advice, and technical support to PSU in the use of the GridAPPS-D platform. PNNL is operated for DOE by the Battelle Memorial Institute under Contract DE-AC05-76RL01830.

ABSTRACT The electric power system is currently undergoing a major transition due to growing numbers of distributed energy resources (DERs) and increased distribution automation. If optimally managed and operated, DERs could provide flexibility and highly valuable grid services such as restoration, peak shaving, voltage regulation, and frequency support to maintain grid reliability. Different applications and enterprises, such as distributed energy resources management systems (DERMS), are being developed for coordinated and optimal operation of DERs. However, to attract sufficient DER participation and achieve the coordinated operation of DERs, systems and components must be interoperable and information exchange must be secure. Along this line, the Portland State University power engineering group is developing Energy Grid of Things (EGoT) DERMS prototype. The proposed application requires the coordinated dispatch of large numbers of DERs and testing such a system presents a challenge; it is not practical to test system prototypes using thousands of real DERs. Hence, the modeling environment (ME) is designed as a co-simulation tool to model interactions between a DERMS and a mass of simulated DERs. The ME is expected to address the scalability issue inherent to hardware-in-the-loop DERMS simulation; many assets are needed to observe effects on the grid from deployment and dispatch of DERs. To enable the development and testing of such advanced applications for power distribution system planning and operations, the U.S Department of Energy developed GridAPPS-D$^{\text{TM}}$, an open-source, standards-based platform at Pacific Northwest National Laboratory. This paper introduces the proposed ME for testing a DERMS application. The architecture of the ME is presented, and the GridAPPS-D features for a such simulation environment are discussed. Additionally, the procedure for developing the ME within the GridAPPS-D platform and use of different APIs for efficient and timely integration are discussed in detail.

INDEX TERMS Common information model, distributed energy resources, grid services, power system simulation.

NOMENCLATURE

ACRONYMS

ADMS	Advanced Distribution Management System
API	Application Programming Interface
BPS	Bulk Power System
CIM	Common Information Model
DERMS	Distributed Energy Resources Management System
DTM	Distributed Trust Model
EDM	Electrical Distribution Model
GSP	Grid Service Provider

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

HELICS	Hierarchical Engine for Large-scale Infrastructure Co-Simulation
IEC	International Electrotechnical Commission
ME	Modeling Environment
MRID	Master Record Identifiers
m-MRID	Measurement MRID of a given type for a particular component
RDF	Resource Description Framework
SPC	Service-Provisioning Customer
SPARQL	SPARQL Protocol and RDF Query Language
STOMP	Streaming Text Orientated Messaging Protocol
UUID	Universally Unique Identifier

DEFINITION OF TERMS

CIM	A standard developed by electric power industry and adopted by the IEC to exchange information about an electric network
DER	Grid-enabled generation, storage, and load assets located behind a customer meter
GridAPPS-D	An open-source, standards-based platform for application development
GridLAB-D	Distribution grid simulator
OpenDSS	Distribution grid simulator
CIMHub	Tool set for translating electric power distribution system models between various formats. Java component of GridAPPS-D
Blazegraph	A triple store and graph database
Query	Request for information, data results, and for action on data
SPARQL	A query language for databases
PNV/VA/Pos	Measurement types: phase-to-neutral voltage, complex power, and equipment status

I. INTRODUCTION

The electric power distribution system is rapidly transforming from a passive to an active system with increasing penetrations of Distributed Energy Resources (DERs), including responsive loads, and with the widespread deployment of sensing and communication infrastructure [1], [2]. The high penetration of DERs in modern distribution systems introduces intermittent power generation, stochastic system operating conditions, and bi-directional power flows that impact system responses to various types of disturbances. This poses challenges to the operation and reliability of the Bulk Power System (BPS). On the positive side, there are many potential benefits, such as reduced environmental impact, deferred capacity upgrades, optimized distribution operations, lower emissions, grid flexibility, optimized distribution operations, improved power system resiliency, and expanded DER-grid service capabilities [3]. It is of paramount importance for grid operators to fully understand the reliability impacts of the high penetration of DERs on the BPS and ensure secure and reliable grid planning and operations [4].

Over the past decade, extensive research has been conducted regarding DER management, coordinated operational schemes, and optimal energy dispatch [5]–[8]. DERs can support a more flexible and efficient grid, and several studies have evaluated technologies based on their abilities to provide energy, capacity, and ancillary services for both the distribution and bulk power systems [3], [9]–[12]. For example, it's been demonstrated that aggregated water heaters have the ability to provide demand response and frequency response services [13]–[15]. Similarly, battery energy storage systems are being deployed to provide several different services, such as energy scheduling, reserves, black-start, voltage management, and frequency response [16]–[20]. A Distributed Energy Resource Management System (DERMS) can be used to coordinate these assets and leverage DER to provide grid services and potentially bid into wholesale markets [21].

Various subsystems and model-based applications within an Advanced Distribution Management System (ADMS) can also be used to manage DERs and offer affordable, reliable, secure, and resilient electricity services [22]–[24]. Different applications and enterprises are expected to work in coordination to determine which distribution grid services are needed and provide the services as requested [25]. Currently, most ADMS software is proprietary and non-configurable thus making ADMS challenging to implement and integrate across different systems. These challenges impose high costs thus limiting the deployment of such systems to large utilities. There is a need to enable cost-effective development and deployment of advanced applications for distribution system planning and operations. One of the elements to address the aforementioned need is to define data abstractions for application development based on interoperability standards such as the Common Information Model (CIM) [26]. CIM is applied at the interface between applications to provide to provide a uniform name space and ontology (i.e. semantic interoperability) for coordinated operations.

Both DERMS and ADMS may need to coordinate tens of thousands of DERs to provide grid services that have a discernible impact within a balancing area [15]. To attract sufficient DER participation, DER owners need assurance that their privacy is respected and that they are in control of their DERs [27]. System components must be interoperable to promote technological innovation, and information must be secure to ensure trust and motivate participation [28], [29]. To abide by these constraints while providing effective grid services, the system architecture must be service-oriented, and information exchange must be secure, trustworthy and based on open communication protocols [30]. Validating that a DERMS or ADMS achieves these objectives requires thorough system testing using a simulation platform.

In this paper, we introduce the DERMS developed by Portland State University and describe how the Modeling Environment (ME) for simulating DERs is developed using GridAPPS-D, an open-source standards-based simulation

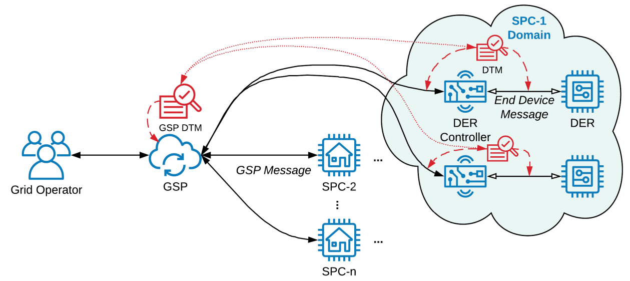


FIGURE 1. The PSU DERMS: a GSP provides grid services to a Grid Operator (GO) through the coordinated dispatch of DERs (blue elements). DTMs monitor information exchange between participants to provide measures of trustworthiness (red elements).

platform [31]. Furthermore, we provide detailed instruction on how different platform features are utilized in developing and integrating the ME. The rest of the paper is organized as follows. In Section II, a DERMS application developed by PSU is introduced and the need for open source platforms is discussed. Section III describes the architecture of GridAPPS-D and the ME for DERMS. Section IV details the procedures by which the GridAPPS-D platform was used to create the ME. Section V describes in detail the use of Application Program Interfaces (APIs) for integration with the platform. A sample use case with the IEEE 13-bus test feeder is detailed in Section VI. Section VII provides a discussion around the motivation of using the GridAPPS-D platform in developing the ME. Finally, Section VIII concludes the paper.

II. EGoT–INTRODUCTION TO PSU DERMS

The Portland State University (PSU) Power Engineering Group is developing an Energy Grid of Things (EGoT) DERMS. Figure 1 shows an overview of the information exchange pathways between EGoT participants [32]. These participants are the Grid Operator (or systems governed by the Grid Operator), the Grid Service Provider (GSP), and Service-Providing Customers (SPCs) which own the DERs. This DERMS is used by a GSP to coordinate the dispatch of DERs. However, it is not practical to test system prototypes using thousands of real DERs. A simulation solution is required that provides a co-simulation environment, thereby allowing integration between the GSP DERMS and a grid simulation tool.

A. MODELING ENVIRONMENT

The PSU team is developing an EGoT ME[1] to provide simulation and testing for the EGoT DERMS. The team intends to test these systems within several electrical system models, and to implement communications protocols to manage information exchange between participants.

The Modeling Environment (ME) receives inputs from a significant number of DERs which may be simulated, physical systems providing real time operating data, or logged operating data. The inputs may come from one or many DERs and provide the DER's power draw from or injection to the grid over time. These inputs are sent to a running grid simulation which produces outputs for data analysis or feedback to the GSP, The GSP then determines whether to dispatch more or fewer of the DERs as appropriate, completing the operating loop. The system is designed to be compatible with any conceivable current or future DER models; also, it is meant to facilitate testing of a variety of potential aggregator systems.

B. OPEN SOURCE PLATFORMS FOR DERMS

As DERs such as solar PV, battery energy storage and flexible loads become more common, there is a strong need to provide tools for demonstrating their capabilities. A principal responsibility of utilities is to ensure safe and reliable power distribution; therefore, contemporary technologies, and the advanced algorithms used to integrate them, need to be evaluated extensively prior to adoption to ensure they will not disrupt the reliability of the power system.

Open-source platforms can help in the development of new methods and integration of legacy methods, while promoting transparency and collaboration. For example, new market mechanisms, e.g. retail markets, peer-to-peer markets, are being proposed. These mechanisms require adequate simulators and co-simulation platforms, and they must also provide sufficient speed to allow the transactive energy mechanisms to be developed in an iterative, incremental, and evolutionary manner [33], [34].

There have been numerous simulation tool developments directed using open-source code bases such as OpenDSS and GridLAB-D. OpenDSS has been used to conduct several smart grid research projects, including advanced automation, DER penetration, state estimation, and green circuits [35]. Similarly, GridLAB-D is an open-source system that provides the next generation of power system simulation integrated with high-performance computing capabilities to deliver in end-use modeling, coupled with distribution automation models, and software integration tools for users of many power system analysis tools [36]. Additionally, a number of testbeds have been established by national laboratories and universities for distribution system modeling and simulation [37]–[40]. These testbeds integrate a variety of power system simulators, devices, software platforms, and research applications such as [41], [42]

The advanced open-source tools are co-simulation environments that use different open-source simulators to emulate the complexity of the power grid by allowing seamless communication of transmission, distribution, and transactive data among elements of the grid [37], [43]–[45]. Recently, an open-source standards-based platform for advanced application development called GridAPPS-D was developed by the U.S Department of Energy at Pacific Northwest National Laboratory (PNNL) [31]. GridAPPS-D supports the development of advanced, data-driven distribution system applications that take advantage of the data-rich environment

[1] Modeling Environment: github.com/PortlandStatePowerLab/doe-egot-me

expected in modernized electric power distribution systems. It provides an open-source, standards-based modeling platform for achieving the modeling environment objectives.

III. MODELING ENVIRONMENT USING GridAPPS-D

In this section, we describe the architecture of GridAPPS-D platform and discuss how the EGoT ME can be developed within the GridAPPS-D environment.

A. GridAPPS-D ARCHITECTURE AND FEATURES

GridAPPS-D is an open source standard-based software platform that provides an environment for developing, testing, and executing data-driven advanced operations and planning applications. The work in [31] describes the logical functionality and conceptual architecture of the GridAPPS-D platform concerning the application developer and commercial tools. A key feature of this framework is standards-based data representation using a standard information model, thus providing application developers with a uniform approach to data.

Figure 2 depicts the architecture of the GridAPPS-D platform. The architecture has five critical functional elements: tools, input/output, development utilities, databus, and distribution simulator. The functional elements of GridAPPS-D enable the separation of application development from data infrastructure. There are two types of data flow classes: data flow to the power system model and control/configuration signals to the power system simulator. The input-output interface allows convenient access to the power system model and data and sends control signals to the distribution simulator using standard message schemas through the Databus (see the red block in Fig. 2). The Databus uses extensible standards such as CIM for developing standard data models. These models enable the semantic interoperability of the developer's data model and data exchange. The "tools" includes optimization toolboxes, state estimators, statistical processing, and the "development utilities" includes loggers, debuggers, access control, test managers, user interface toolkits, and other application support functions, etc. Both the "tools" and "development utilites" are designed to support efficient application development and enhance application functionality. Finally, the data models are fed to the Distribution simulator, which includes power flow simulators such as GridLAB-D and OpenDSS [35], [36]. This allows functionality to test the application under realistic conditions. Furthermore, the "Data Ingest" function provides the ability to exchange data with commercial tools such as an historian database, Energy Management System (EMS), Distribution Management System (DMS), etc.

GridAPPS-D provides APIs that allow for direct control of simulation components–including DERs in the model–and receipt of grid state measurement outputs. Use of these APIs facilitates development of a simulation environment that functions in real-time to test a DERMS or ADMS. The environment receives inputs from the DERMS in the form of control signals, simulated DER data, etc., and provides outputs to the DERMS in the form of feedback data or electrical grid states. This allows for in-the-loop simulations and testing of the DERMS.

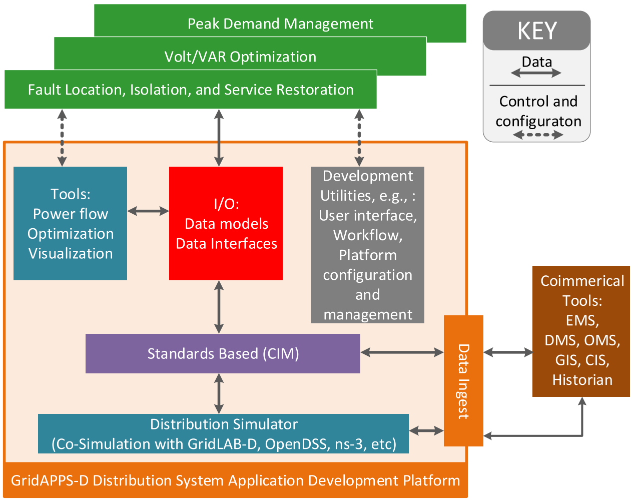

Peak Demand Management
Volt/VAR Optimization
Fault Location, Isolation, and Service Restoration
KEY
Data
Control and configuraton
Tools: Power flow Optimization Visualization
I/O: Data models Data Interfaces
Development Utilities, e.g., : User interface, Workflow, Platform configuration and management
Coimmerical Tools: EMS, DMS, OMS, GIS, CIS, Historian
Standards Based (CIM)
Data Ingest
Distribution Simulator (Co-Simulation with GridLAB-D, OpenDSS, ns-3, etc)
GridAPPS-D Distribution System Application Development Platform

FIGURE 2. GridAPPS-D: Logical functionality and conceptual architecture [31].

B. MODELING ENVIRONMENT ARCHITECTURE

GridAPPS-D is selected as the development platform for the EGoT ME, as it fits all requirements of the system. Battery equivalent models are added to a variety of grid simulation topologies and ingested into the model database. These models serve as representations of the electrical effects of DERs on the grid. The DER models simulate electrical characteristics, which are communicated to the simulation via the GridAPPS-D Simulation API. This produces grid state changes in the simulation. The Simulation API also provides measurement messages to the Python control script, which are converted into data logs or communicated to the GSP.

The ME is a grid simulation and DERMS test platform designed to allow representation of a variety of DERs. The ME consists of the Model Controller, an object-oriented application developed in Python and using the GridAPPS-D platform. Also included in the ME are files for input data, output logs, and system configuration. Grid models are contained within GridAPPS-D's Blazegraph database; during simulation, the Model Controller sends inputs and retrieves outputs from the simulation using the GridAPPS-D APIs. The architecture of the system is depicted in Fig. 3.

The Electrical Distribution Model Core (EDM Core) objects govern startup, configuration, and communication with the GridAPPS-D database and simulation using the GridAPPS-D Python library. An `EDMCore` class contains methods that read configuration parameters from a file, load a preconstructed grid model from the Blazegraph database, and generate lookup tables that associate various MRIDs within the CIM model with their plain-English names.

During startup, the MCInputInterface object reads the user configuration attributes from EDMCore and determines

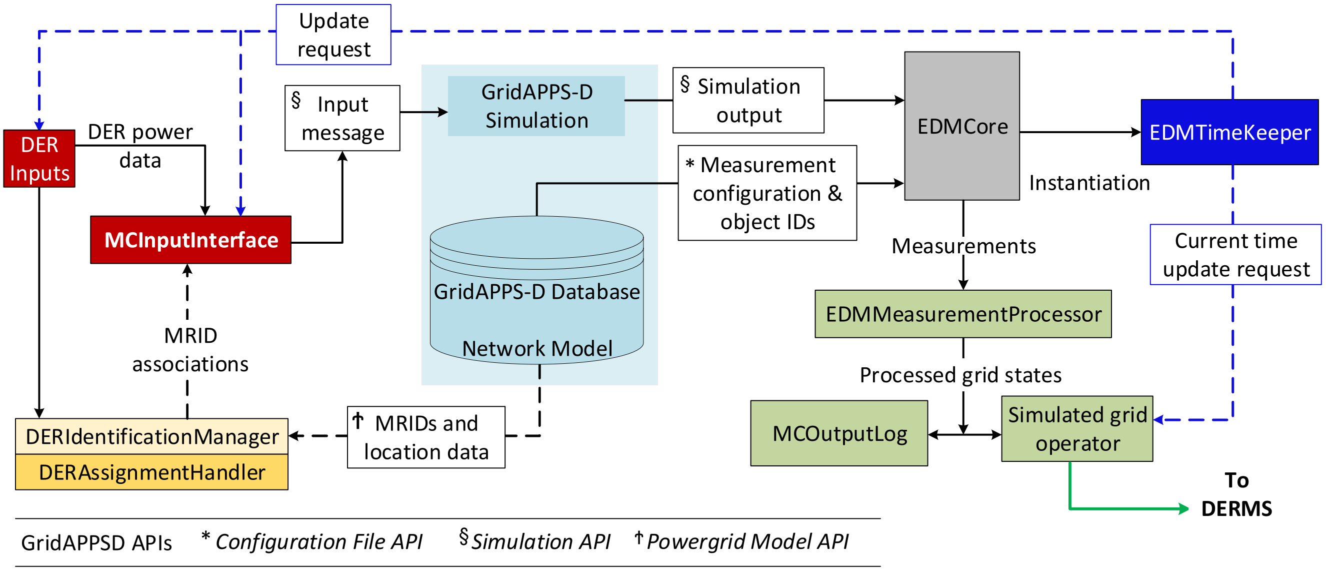


FIGURE 3. ME architecture and its interactions with the GridAPPS-D platform. Colored blocks represent objects or participants within the "Model Controller" Python script. DER inputs (red) can be received from external DER control modules or simulators, historical data logs, etc, and processed into input messages for the simulation. Association data (orange) between input identifiers and the control MRIDs of the DER representations within the grid model keeps track of which inputs provide for which DERs in the model, and allow for automatic assignment of DERs to the proper bus within the model. The light blue block represents the boundaries of the GridAPPS-D simulation and model database within the Docker container; the Model Controller communicates with these for input, output, and model querying purposes. Simulation grid state measurements are provided to the EDMCore class (grey), which also handles simulation configuration and startup. The outputs (green) are parsed by a Measurement Processor into something usable by both the output log class and a simulated grid operator, which is in communication with the DERMS under test. The Timekeeper (blue) ensures that all messages are sent to the proper objects once per simulation time step.

which "simulated DERs" will be used. The simulated DERs could be individual scripts emulating DERs, or a .csv input containing historical data from many DERs. Each of these DERs is automatically associated with a DER-equivalent battery inverter model within the electrical system model by the DERAssignmentHandler object; this association information is contained within the DERIdentificationManager object. This allows inputs from a particular simulated DER to be communicated to and reflected within the model at the proper bus locations.

The `EDMCore` then generates a set of callback objects associated with the simulation:

- `EDMTimeKeeper` contains data and methods that are used once every simulation time step, or every one second. For example: the global timestamp is incremented, and the Input Handlers query DER simulations for new data and send it to the running GridAPPS-D simulation via the Simulation API.
- `EDMMeasurementProcessor` contains data and methods relating to grid measurements, which are provided by the GridAPPS-D Simulation API once every three seconds. These measurements are parsed, reorganized, and sent to the Output Handlers to be processed into .csv logs or checked against thresholds to determine if the simulated Grid Operator (GO) should make a service request. A "service request" is a message formatted per the needs of the DERMS under test which requests a particular grid service from the DERMS with specific parameters at a specific time and duration.

These callback objects provide timing and automation for the script. Inputs to the simulation use the GridAPPS-D Python library functions to automatically generate and send control messages to GridAPPS-D via the Simulation API. The updated grid states are provided in a configurable message to the `EDMOnMeasure` encapsulation method. This message is parsed and sent to a logging class, as well as a Simulated Grid Operator class which provides requests and feedback between the ME and the DERMS under test. These simple, configurable and automatic methods of accessing grid simulation input and outputs allow the team to focus on development of new DER models and rapidly test them without reconfiguring the grid model used.

IV. METHODOLOGY

The Modeling Environment is designed to test the effects of DER dispatch on different grid topologies and using a varying number of DERs. This requires the ability to utilize a variety of grid models as well as a capability to add DERs to each model as required for the test. The GridAPPS-D platform facilitates this by providing a Blazegraph database to which grid models can be added or modified. The following section describes the process in which new feeder models are added or existing feeder models are updated with additional DERs for use by GridAPPS-D.

A. GridAPPS-D DATABASE

The GridAPPS-D platform stores the CIM Extensible Markup Language (XML) model in an open-source

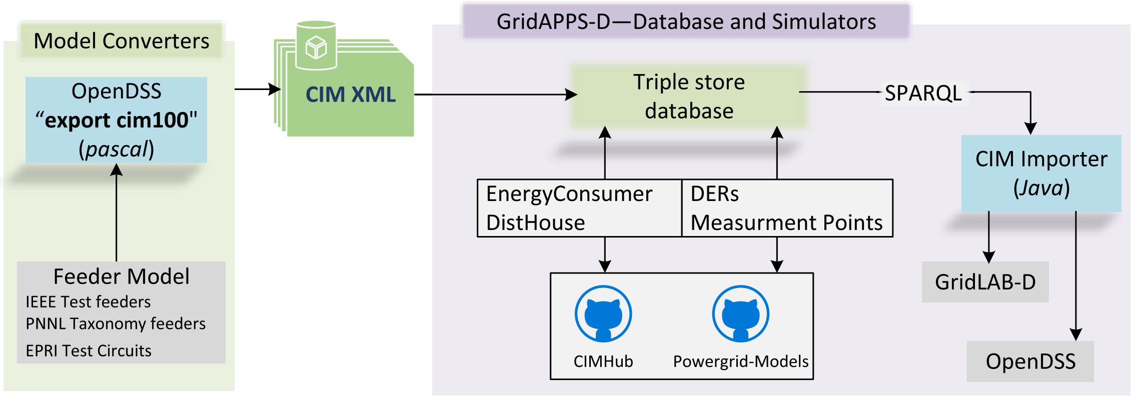


FIGURE 4. Feeder model conversion, ingestion, and database update. The CIMHub and Powergrid-Models repositories allows user to add measurement points, DERs, and house models into the database.

triple-store database called Blazegraph that can be retrieved via a query language such as SPARQL.

This section describes the procedures to ingest models in the CIM XML format from various sources into the database, generate OpenDSS/GridLAB-D files from the database, and do a round trip validation with the models. The CIMHub repository provides a tool set for translating electric power distribution system models between various formats using the International Electrotechnical Commission (IEC) standard 61970/61968 CIM.

B. NETWORK MODEL INGESTION IN DATABASE

Fig. 4 provides a workflow for feeder model conversion and model ingestion. OpenDSS provides a "`cim100`" command that converts an OpenDSS feeder model into CIM XML format. OpenDSS will create random Universally Unique Identifier (uuid) values when running the conversion process for the first time on a new circuit ("`uuids name.dat`"). To re-run the conversion process on the same circuit, previous uuids could be reused ("`export uuids name.dat`"). In case the OpenDSS model is not available for the original feeder, there are conversion scripts and configuration files provided for the users that convert commercial distribution feeder models in common file types to OpenDSS.[2] Once inside the platform, a JAVA program [46] converts the CIM model into the native format of GridLAB-D/OpenDSS, which are the platform simulators within GridAPPS-D. Note that GridLAB-D and OpenDSS are distribution simulators, and for a valid network model, both the simulators produce similar results [47]. Currently, only GridLAB-D is available as a simulator inside the platform; OpenDSS will be available in a future release.

Once the CIM XML feeder model is uploaded into the database, a user can run several SPARQL queries to get the model information and detailed component information for a feeder. The CIMHub module provides sample SPARQL queries[3] that can be pasted into the Blazegraph browser window ("QUERY tab"). Upon execution, a tabular response for a given query is displayed. A SPARQL query consists of triples that filter results from the database. A stored triple takes the form of subject-predict-object, which maps directly to an attribute or association in CIM [48]. A sample query and a response is shown in Fig. 5. The query retrieves the switch information from a feeder model that is specified by an ID. The details on a sample query are summarized in [48].

C. NETWORK MODEL UPDATE IN DATABASE

The Modeling Environment requires the database to update with additional DERs for the simulation process. In what follows, we describe in detail how new DERs and measurement points can be added into the feeder within the database.

1) INSERTING DERs

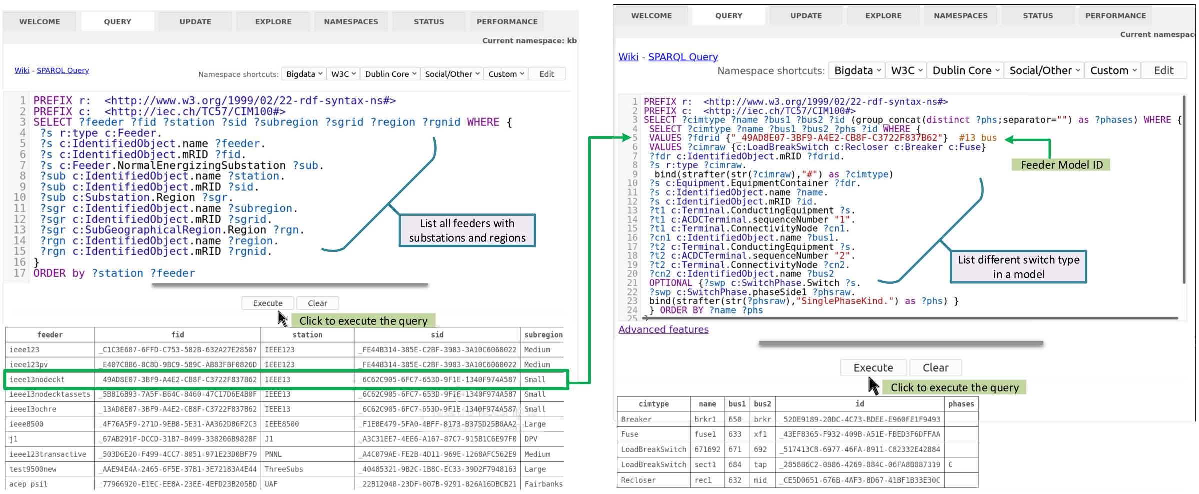
The Modeling Environment will use one topology–or one model–to conduct a variety of tests using differing numbers of DERs. Rather than developing a new model for each test, it is convenient to simply add the required number of DERs to a model on a per-test basis. DERs may be added into an existing network model in Blazegraph. For convenience, the DER locations are defined by a bus name, i.e., the name of CIM `ConnectivityNode`, rather than the MRID of a CIM `ConnectivityNode`. This assumes that `ConnectivityNode` names will be unique, which is not required in CIM, although it is required in most power flow programs. Fig. 6 shows the overall process for inserting DERs into an existing feeder in the database.

The first step is to create a comma-separated "DER definition" file. It contains three specific pieces of information and all are required.

- DER uuid file – This is a filename to save the MRIDs for inserted DER. This file will persist the MRID values if the user has to run the script again to change some DER attributes; any DER that is updated on the same

[2] https://github.com/GRIDAPPSD/CIMHub/tree/master/converters

[3] https://github.com/GRIDAPPSD/CIMHub/blob/master/queries.txt


WELCOME
QUERY
UPDATE
EXPLORE
NAMESPACES
STATUS
PERFORMANCE
Current namespace: kb
Wiki - SPARQL Query
Namespace shortcuts: Bigdata W3C Dublin Core Social/Other Custom Edit
List all feeders with substations and regions
Execute
Clear
Click to execute the query

```
PREFIX r:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX c:  <http://iec.ch/TC57/CIM100#>
SELECT ?feeder ?fid ?station ?sid ?subregion ?sgrid ?region ?rgnid WHERE {
 ?s r:type c:Feeder.
 ?s c:IdentifiedObject.name ?feeder.
 ?s c:IdentifiedObject.mRID ?fid.
 ?s c:Feeder.NormalEnergizingSubstation ?sub.
 ?sub c:IdentifiedObject.name ?station.
 ?sub c:IdentifiedObject.mRID ?sid.
 ?sub c:Substation.Region ?sgr.
 ?sgr c:IdentifiedObject.name ?subregion.
 ?sgr c:IdentifiedObject.mRID ?sgrid.
 ?sgr c:SubGeographicalRegion.Region ?rgn.
 ?rgn c:IdentifiedObject.name ?region.
 ?rgn c:IdentifiedObject.mRID ?rgnid.
}
ORDER by ?station ?feeder
```

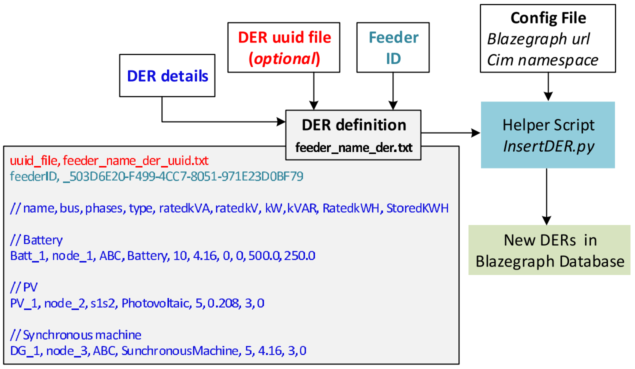
feeder	fid	station	sid	subregion
ieee123	_C1C3E687-6FFD-C753-582B-632A27E28507	IEEE123	_FE44B314-385E-C2BF-3983-3A10C6060022	Medium
ieee123pv	_E407CBB6-8C8D-9BC9-589C-AB83FBF0826D	IEEE123	_FE44B314-385E-C2BF-3983-3A10C6060022	Medium
ieee13nodeckt	_49AD8E07-3BF9-A4E2-CB8F-C3722F837B62	IEEE13	_6C62C905-6FC7-653D-9F1E-1340F974A587	Small
ieee13nodecktassets	_5B816B93-7A5F-B64C-8460-47C17D6E4B0F	IEEE13	_6C62C905-6FC7-653D-9F1E-1340F974A587	Small
ieee13ochre	_13AD8E07-3BF9-A4E2-CB8F-C3722F837B62	IEEE13	_6C62C905-6FC7-653D-9F1E-1340F974A587	Small
ieee8500	_4F76A5F9-271D-9EB8-5E31-AA362D86F2C3	IEEE8500	_F1E8E479-5FA0-4BFF-8173-B375D25B0AA2	Large
j1	_67AB291F-DCCD-31B7-B499-338206B9828F	J1	_A3C31EE7-4EE6-A167-87C7-915B1C6E97F0	DPV
ieee123transactive	_503D6E20-F499-4CC7-8051-971E23D0BF79	PNNL	_A4C079AE-FE2B-4D11-969E-1268AFC562E9	Medium
test9500new	_AAE94E4A-2465-6F5E-37B1-3E72183A4E44	ThreeSubs	_40485321-9B2C-1B8C-EC33-39D2F7948163	Large
acep_psil	_77966920-E1EC-EE8A-23EE-4EFD23B205BD	UAF	_22B12048-23DF-007B-9291-826A16DBCB21	Fairbanks

WELCOME
QUERY
UPDATE
EXPLORE
NAMESPACES
STATUS
PERFORMANCE
Current namespac
Wiki - SPARQL Query
Namespace shortcuts: Bigdata W3C Dublin Core Social/Other Custom Edit
Feeder Model ID
List different switch type in a model
Advanced features
Execute
Clear
Click to execute the query

```
PREFIX r:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX c:  <http://iec.ch/TC57/CIM100#>
SELECT ?cimtype ?name ?bus1 ?bus2 ?id (group_concat(distinct ?phs;separator="") as ?phases) WHERE {
 SELECT ?cimtype ?name ?bus1 ?bus2 ?phs ?id WHERE {
 VALUES ?fdrid {"_49AD8E07-3BF9-A4E2-CB8F-C3722F837B62"}  #13 bus
 VALUES ?cimraw {c:LoadBreakSwitch c:Recloser c:Breaker c:Fuse}
 ?fdr c:IdentifiedObject.mRID ?fdrid.
 ?s r:type ?cimraw.
  bind(strafter(str(?cimraw),"#") as ?cimtype)
 ?s c:Equipment.EquipmentContainer ?fdr.
 ?s c:IdentifiedObject.name ?name.
 ?s c:IdentifiedObject.mRID ?id.
 ?t1 c:Terminal.ConductingEquipment ?s.
 ?t1 c:ACDCTerminal.sequenceNumber "1".
 ?t1 c:Terminal.ConnectivityNode ?cn1.
 ?cn1 c:IdentifiedObject.name ?bus1.
 ?t2 c:Terminal.ConductingEquipment ?s.
 ?t2 c:ACDCTerminal.sequenceNumber "2".
 ?t2 c:Terminal.ConnectivityNode ?cn2.
 ?cn2 c:IdentifiedObject.name ?bus2
 OPTIONAL {?swp c:SwitchPhase.Switch ?s.
 ?swp c:SwitchPhase.phaseSide1 ?phsraw.
 bind(strafter(str(?phsraw),"SinglePhaseKind.") as ?phs) }
 } ORDER BY ?name ?phs
}
```

cimtype	name	bus1	bus2	id	phases
Breaker	brkr1	650	brkr	_52DE9189-20DC-4C73-BDEE-E960FE1F9493	
Fuse	fuse1	633	xf1	_43EF8365-F932-409B-A51E-FBED3F6DFFAA	
LoadBreakSwitch	671692	671	692	_517413CB-6977-46FA-8911-C82332E42884	
LoadBreakSwitch	sect1	684	tap	_2858B6C2-0886-4269-884C-06FA8B887319	C
Recloser	rec1	632	mid	_CE5D0651-676B-4AF3-8D67-41BF1B33E30C	

FIGURE 5. Query to get all feeders available in the database and a sample query executed for a IEEE 13-bus feeder model.



```
uuid_file, feeder_name_der_uuid.txt
feederID, _503D6E20-F499-4CC7-8051-971E23D0BF79

// name, bus, phases, type, ratedkVA, ratedkV, kW, kVAR, RatedkWH, StoredkWH

// Battery
Batt_1, node_1, ABC, Battery, 10, 4.16, 0, 0, 500.0, 250.0

// PV
PV_1, node_2, s1s2, Photovoltaic, 5, 0.208, 3, 0

// Synchronous machine
DG_1, node_3, ABC, SunchronousMachine, 5, 4.16, 3, 0
```

FIGURE 6. Inserting DERs into existing network model.

bus will use the same MRID values. If this file doesn't exist, random MRID values will be generated, and then saved for subsequent re-use.

- Feeder ID – This is the MRID of the feeder to search for the buses with DER. This feeder should already be present in the Blazegraph database.
- DER details – This defines attributes for each new or updated DER. The comma-separated fields are the name of the DER, bus name of DER connection, phases (ABC/s1s2), type, rated kVA & rated kV of the inverter, active power output in kW, reactive power output in kVAr, rated energy in kWh, and stored energy in kWh. The last two fields are only specified for battery inverter models and may be omitted for Photovoltaic and SynchronousMachine.

Once DERs are defined, they are inserted into the database using the CIMHub Python utility program.[4] The required arguments to the scripts are the DER definition file and a configuration file with a Blazegraph url and CIM namespace.

A sample configuration file is shown below:

```
{
"blazegraph_url": "http://localhost:8889
/bigdata/namespace/kb/sparql,"
"cim_ns": "<http://iec.ch/TC57/CIM100#"
}
```

2) INSERTING MEASUREMENTS

After the Blazegraph database has ingested the detailed feeder model and additional DERs, measurement points are added into the database for each of the added DERs. This must be done before starting the simulation. Inserting measurements allows the user to query for particular simulation output data. Four different types of measurements are currently supported: phase-to-neutral voltage (PNV), complex power (VA), discrete measurements such as switch state, capacitor status, regulator tap (Pos), and state of charge (SoC). The utility functions in CIMHub are used to insert measurements into the database. Fig. 7 shows the overall process for inserting measurements for a feeder in the Blazegraph database.

The first step in the measurement insertion process utilizes the script `ListMeasureables.py` to create the measurement lists in several .txt files. The required arguments to the scripts are the configuration file with Blazegraph url and CIM namespace, feeder name, feeder ID, and the output path where the measurement lists will be written. Next, the script `InsertMeasurements.py` uses the previously-written measurement lists to update the model in the database with measurement points. The required arguments to the scripts are a configuration file with a Blazegraph url and CIM namespace, the file name of the measurement lists (txt file) created in the first step, and the name of the JSON file containing measurement MRID (m-MRID) values. An m-MRID is an unique alphanumeric string that is defined for each measurement type (i.e., PNV, VA, Pos, and SoC) for various components such as lines, loads, transformers, DERs, switches,

[4] https://github.com/GRIDAPPSD/CIMHub/tree/master/utils

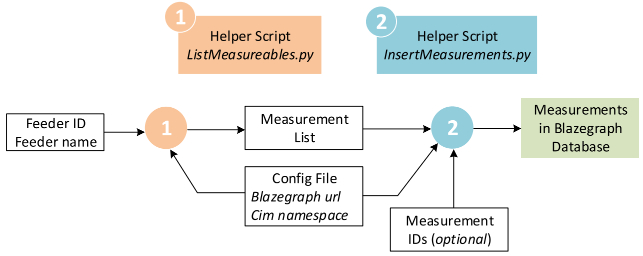


FIGURE 7. Inserting measurements into Blazegraph database for a feeder model.

capacitors and synchronous machines. Note that if the file name provided for the m-MRID values does not correspond to an existing file, a new file will be created and populated with random m-MRIDs. The file then can be saved for subsequent re-use.

Once added to the model, these measurements become available during simulation runtime via the simulation API (described below in section V.C). These measurements can be fed back to the DERMS via a ''simulated grid operator'' function to verify that the DER dispatch was executed properly. They can also be provided as logs to a test engineer for data analysis and functional testing purposes.

V. APIs FOR MODELING ENVIRONMENT AND GridAPPS-D INTEGRATION

APIs provide an easy interface by enabling access to data and services by making logical references to the data model. This reduces the cost and complexity of integration of applications and interaction between multiple systems. In this section, we describe how GridAPPS-D's APIs are used to integrate the simulation and provide communications between it and external systems such as the Modeling Environment. GridAPPS-D platform defines and implements a set of APIs to develop and test advanced applications and services. The platform provides a Python library that wraps these APIs for efficient and timely integration. Refer to the GitHub repository for installing the library in a local environment.[5]

The most commonly used functions and the required arguments are:

- `send(topic, message)`
- `get_response(topic, message, timeout=5)`
- `subscribe(topic, callback)`

where,

→ `topic`: A channel on which to communicate with the API

→ `message`: A valid python dictionary or JSON-formatted string

→ `callback`: A function that is called automatically at predetermined intervals

→ `timeout`: Wait time for the response from an API call

Next, we describe several APIs that are used by the ME to interact with the GridAPPS-D platform. A sample request and response for each API is provided in [49]. The Modeling Environment uses these APIs to communicate with the GridAPPS-D database and simulation for input and output purposes, as will be described below.

A. POWERGRID MODEL API

The Powergrid Model API allows the user to query the power grid model data store. It is mostly used to understand and extract data from the underlying feeder model. Note that any model-based algorithm requires an accurate model for making decisions. This API can be used to retrieve the static model that includes graph connectivity; component specific information such as names, IDs, ratings, rated power, rated voltage, admittances, susceptances; and measurement points created for specific components.

The Model Controller uses the Powergrid Model API to retrieve MRIDs for each DER on each bus in the model. This allows input DERs to be assigned to model DERs on the proper bus. Without this API, each input DER would need to be hardcoded with the control MRID of a DER on the proper bus.However, the inputs do not have this information since it's internal to the model database and the inputs are expected to be external simulations or programs.

B. CONFIGURATION FILE API

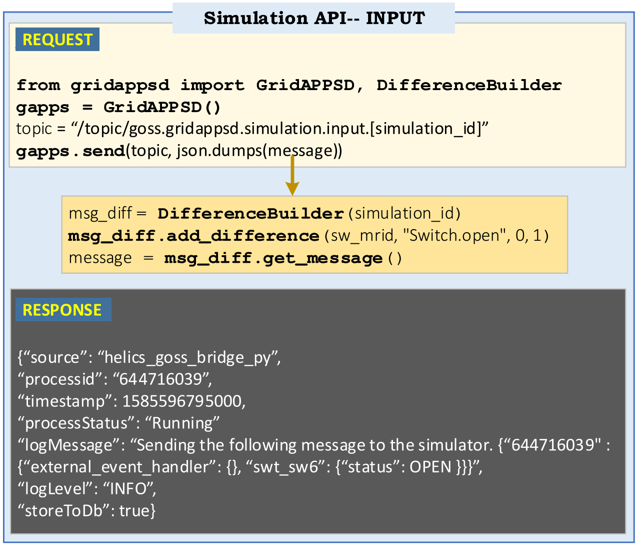
The Configuration File API provides all the configuration files necessary to run a simulation using the simulator (OpenDSS/GridLAB-D) and generates other files such as GridLAB-D Symbols, CIM Dictionary, DSS All, DSS Base, Ybus Export, and Vnom Export, depending on the ''configurationType'' [49]. The Model Controller uses this to request the model dictionary file of a feeder necessary to associate measurements with inputs. This is fed through the system to the measurement processor and, eventually, the output logs (See Fig. 3). This information is necessary to verify that changes on a particular input are being reflected within the system.

C. SIMULATION API

The Simulation API is used to subscribe to the simulation output and send inputs to the running simulation. It can also be used to start, pause, and resume the simulation.

The Model Controller extensively uses the Simulation API for both input and output purposes. The MC Input Interface retrieves updated DER input data once per time step; each input DER is assigned an associated control MRID within the model, and this association data is maintained within the DER Identification Manager. The Input Interface uses the updated DER data to generate difference messages in the proper format using the GridAPPS-D Python library (see Fig. 8.) These difference messages are then sent to the model via the Simulation API; the simulator updates the DERs within the model with the new states (such as new real power consumption levels, for example) and these affect change on the grid

[5] https://github.com/GRIDAPPSD/gridappsd-python



Simulation API-- INPUT

REQUEST

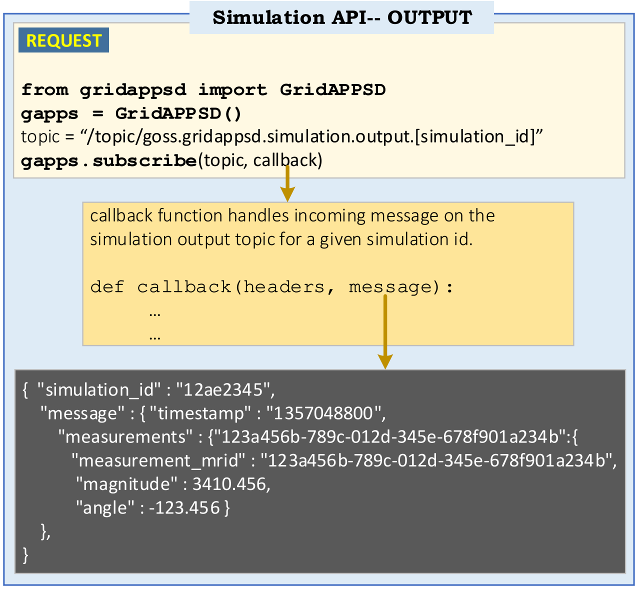
```
from gridappsd import GridAPPSD, DifferenceBuilder
gapps = GridAPPSD()
topic = "/topic/goss.gridappsd.simulation.input.[simulation_id]"
gapps.send(topic, json.dumps(message))
```

```
msg_diff = DifferenceBuilder(simulation_id)
msg_diff.add_difference(sw_mrid, "Switch.open", 0, 1)
message = msg_diff.get_message()
```

RESPONSE

```
{"source": "helics_goss_bridge_py",
"processid": "644716039",
"timestamp": 1585596795000,
"processStatus": "Running"
"logMessage": "Sending the following message to the simulator. {"644716039" :
{"external_event_handler": {}, "swt_sw6": {"status": OPEN }}}",
"logLevel": "INFO",
"storeToDb": true}
```

FIGURE 8. A sample request and response in using Simulation Input API.



Simulation API-- OUTPUT

REQUEST

```
from gridappsd import GridAPPSD
gapps = GridAPPSD()
topic = "/topic/goss.gridappsd.simulation.output.[simulation_id]"
gapps.subscribe(topic, callback)
```

callback function handles incoming message on the simulation output topic for a given simulation id.

```
def callback(headers, message):
    ...
    ...
```

```
{ "simulation_id" : "12ae2345",
   "message" : { "timestamp" : "1357048800",
      "measurements" : {"123a456b-789c-012d-345e-678f901a234b":{
        "measurement_mrid" : "123a456b-789c-012d-345e-678f901a234b",
         "magnitude" : 3410.456,
         "angle" : -123.456 }
   },
}
```

FIGURE 9. A sample request and response in using Simulation Output API.

model. Note that the API needs a topic with simulation id that allows it to pass the message into the running simulation. Inside the platform, the CIM difference message is translated by the HELICS bridge, then the message is finally sent to the simulator (GridLAB-D/OpenDSS).

Observation of these changes requires measurement of the model parameters over time. Once the simulation of the grid at a given condition is complete, the power flow solution and equipment state can be retrieved by subscribing to the simulation output using this API. Thus, system measurements are provided to the Model Controller via the Simulation API as they become available. This occurs every three seconds and is performed by a callback function (Fig. 9). Every time measurements become available, this callback function is called with all of the grid state measurements provided as an argument. These measurements are in dictionary form, with each key representing the m-MRID and each value containing measurement data. A Measurement Processor class in the Modeling environment parses those dictionaries, renames the m-MRIDs to a plain English name for each measurement point, appends data from the Identification Manager to the values so the user knows which input each measurement is associated with, and provides the measurement data to a log writer as well as a simulated grid operator function (See Fig. 3.)

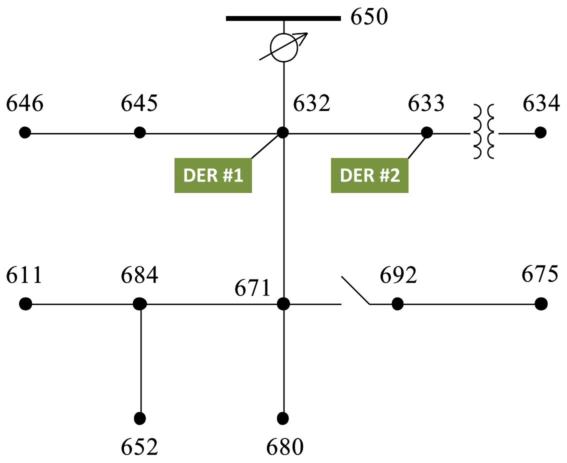


FIGURE 10. Modified IEEE 13-bus test feeder.

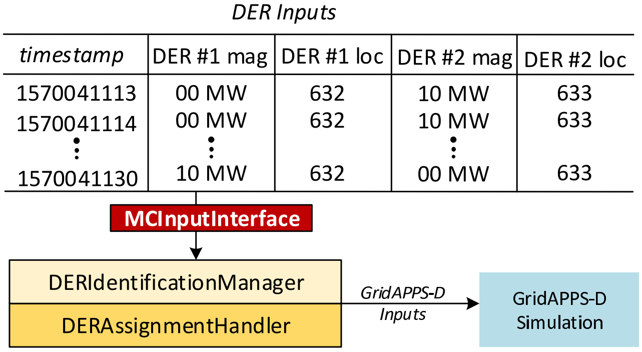
VI. USE CASE

In this section, we provide a sample use case for the proposed ME using a modified IEEE 13-bus test feeder model. The model was modified by the addition of two controllable DER-equivalent battery inverter models (See Fig. 10), following the process outlined in section Section IV-B. One DER was added to bus 632, and one was added to bus 633. In this use case, the inputs to these DERs derive from an input log, and simulation output measurements are formatted and written to an output log. For simplicity, the simulated grid operator is not considered in this section as its functionality is not relevant to GridAPPS-D and it's interactions with the Modeling Environment.

A. INPUTS TO THE GridAPPS-D SIMULATION

A sample taken from a user-written input file is shown in Fig. 11. This file contains DER inputs for two DERs: DER1 and DER2. During the simulation startup process, an input class parses this file and associates DER1 with the control MRID of the battery inverter model on bus 632, and DER2 with the MRID of the model on bus 633.

During the simulation, the input file is read at each time step by the DER input class. If a line of data corresponds with the current simulation time, that line is read into the system.



DER Inputs

timestamp	DER #1 mag	DER #1 loc	DER #2 mag	DER #2 loc
1570041113	00 MW	632	10 MW	633
1570041114	00 MW	632	10 MW	633
⋮	⋮		⋮	
1570041130	10 MW	632	00 MW	633

FIGURE 11. Input file for GridAPPS-D simulation.

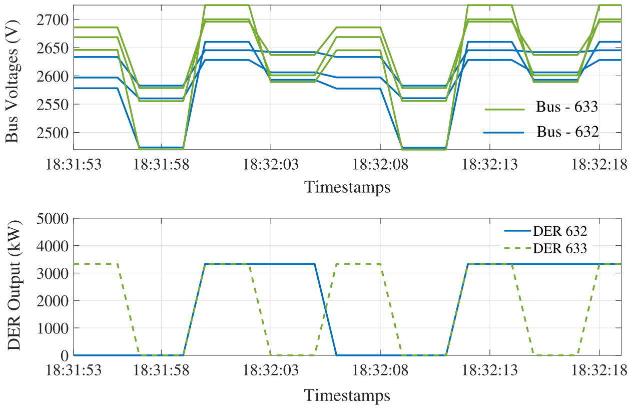


FIGURE 12. Grid states (bus voltages) as a result of DER inputs from the GridAPPS-D simulation.

For instance, at time 1570041114, DER1 power is 0 MW and DER2 power is 10 MW. These values are provided to an input manager class, which automatically provides the 0 MW value to the battery inverter control MRID associated with DER1, and 10 MW to the MRID associated with DER2, by parsing the values and packaging them into properly formatted messages for the Simulation API. The Modeling Environment is designed to use multiple input APIs simultaneously. As long as the inputs provided to the system include a unique identifier, a location, and values for each associated DER in the model, a DER input class can be written to provide the required information to the ME input manager. For example, data from a running DER emulator could be read in by an input API once per simulation time step to provide a real-time representation of a DER. This does not exclude the ME from simultaneously reading and using logged data as described above.

At each simulation time step, the input manager class checks each input API for new inputs. If new inputs are found, it packages them into the appropriate difference messages and sends those to the simulation (See Figure 8 for a sample difference message).

B. OUTPUTS FROM THE GridAPPS-D SIMULATION

Every three seconds, the GridAPPS-D simulation provides a set of grid state measurements to a measurement processing class. These measurements are labelled by m-MRID and include voltage, current, power, and switch state for all items in the model, including nodes and loads. This also includes measurements for the battery inverter models to which DER1 and DER2 were assigned. These measurements are provided in a "raw" form: they contain magnitudes, angles, and/or switch states, but are identified only by m-MRID, which is by definition a randomized string of letters and numbers. In order to make the m-MRID human-readable, the measurement processor sends queries to the model database. These queries include measurement names, the bus location of each measurement, and the type of measurement. These values are added to the grid state measurements in the measurement processor. Finally, the measurements are timestamped and sent to a logging class to be written to a .csv file for later analysis. Sample timestamped measurements for DER inputs and bus voltages are plotted in Fig. 12.

VII. DISCUSSION

With the increasing penetration of DERs such as rooftop photovoltaic solar panels, behind-the-meter batteries, and responsive loads, it is common for utilities to enable the provision of flexibility and grid services. However, there exist barriers to achieving such advanced capabilities: a) the cost and complexity of integrating the various data systems and b) the data integration and interfaces are typically proprietary. In such environments, the architectural impacts are significant and the successful interconnection of any electric power system component is measured by the interoperability–a capability that drives integration to be economical, simple, and reliable [27]. It indicates the ability of two or more networks, systems, devices, applications, or components to access, exchange, integrate and cooperatively use data in a coordinated manner without significant user intervention [50]. To achieve interoperability, it is fundamental that all the parties have a shared understanding of the information semantics—this can be achieved by leveraging standards-compliant data models, communications, and application architectures [51].

The GridAPPS-D platform is based on reference architecture and it applies existing standards to create standardized data models, APIs, and data exchange mechanisms to support data needs for the advanced applications [52]. Fundamentally, it helps transitioning distribution management architectures from a proprietary, vertically integrated approach to one that is open, layered, and standards-based (See Fig. 2). This layered architecture has the following advantages:

- Enables separation of functionality (application development) from data infrastructure.
- Supports the use of industry standards such as the Common Information Model.
- Shifts the focus of software development from protecting closed architectures, to a more collaborative open architecture.

As a reference implementation, architecture, and design, the results of the proposed ME are replicable, and the implementation can be upgraded/configured easily by individuals or entities beyond the original designer. It is an open-source, object-oriented system (designed with easily modifiable modules) that is configurable, scalable, and suitable for a variety of tests for several grid scales, DER counts/types, and grid services.

VIII. CONCLUSION

With millions of DERs coming online each year at the grid-edge, researchers are envisioning the need to control and monitor these DERs in real-time in order to leverage their potential as flexible assets. PSU is developing an EGoT DERMS that can coordinate the dispatch of DERs to provide grid services. A simulation solution is required for simulating the large number of DERs within a grid model to allow integration between DERMS and a grid simulation tool. Towards this goal, PSU has developed the ME by leveraging the GridAPPS-D platform. This paper introduced the ME developed for testing a DERMS application. The architecture of the ME is presented in detail and the interactions of the ME with the GridAPPS-D platform are discussed. Additionally, the procedure for developing the ME within the platform and the use of different APIs are provided in detail. The ME developed within the platform is able to model interactions between a DERMS and a large number of DERs. PNNL has provided training, advice, and technical support for the use of the GridAPPS-D platform. The successful deployment of the proposed ME within the GridAPPS-D platform provides a proof-of-concept for the adoption of such a simulation-based environment to test advanced applications in an ADMS environment. This document serves as an example and guide for adopting the GridAPPS-D platform as a simulation environment for managing the ever-increasing number of DERs that can be used individually or in aggregate to provide value to the grid, individual customers, or both.

ACKNOWLEDGMENT

The authors would like to acknowledge Jonathan L. Barr from the Pacific Northwest National Laboratory for his valuable feedback and suggestions. Portland State University (PSU) is developing an Energy Service Interfaces (ESI) for the U.S. Department of Energy. The Pacific Northwest National Laboratory (PNNL) provided training, advice, and technical support to PSU in the use of the GridAPPS-D platform. *(Shiva Poudel and Sean J. Keene contributed equally to this work.)*

REFERENCES

[1] U.S. Department of Energy. Voices of Experience: Insights. (Feb. 2015). *Into Advanced Distribution Management Systems*. [Online]. Available: https://www.smartgrid.gov/ADMS.html

[2] R. B. Melton, K. P. Schneider, T. E. McDermott, and S. V. Vadari, "GridAPPS-D conceptual design V1.0," Pacific Northwest Nat. Lab., Richland, WA, USA, Tech. Rep. PNNL-26340, 2017.

[3] M. Obi, T. Slay, and R. Bass, "Distributed energy resource aggregation using customer-owned equipment: A review of literature and standards," *Energy Rep.*, vol. 6, pp. 2358–2369, Nov. 2020.

[4] N. Kang, R. Singh, J. T. Reilly, and N. Segal, "Impact of distributed energy resources on the bulk electric system combined modeling of transmission and distribution systems and benchmark case studies," Argonne Nat. Lab., Argonne, IL, USA, Tech. Rep. ANL/ESD-17/26 140856, 2017.

[5] I. K. Song, W. W. Jung, J. Y. Kim, S. Y. Yun, J. H. Choi, and S. J. Ahn, "Operation schemes of smart distribution networks with distributed energy resources for loss reduction and service restoration," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 367–374, Mar. 2013.

[6] F. Adinolfi, G. M. Burt, P. Crolla, F. D'Agostino, M. Saviozzi, and F. Silvestro, "Distributed energy resources management in a low-voltage test facility," *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2593–2603, Apr. 2015.

[7] N. Amjady, A. Attarha, S. Dehghan, and A. J. Conejo, "Adaptive robust expansion planning for a distribution network with DERs," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1698–1715, Mar. 2018.

[8] S. Xia, S. Bu, C. Wan, X. Lu, K. W. Chan, and B. Zhou, "A fully distributed hierarchical control framework for coordinated operation of DERs in active distribution power networks," *IEEE Trans. Power Syst.*, vol. 34, no. 6, pp. 5184–5197, Nov. 2019.

[9] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 134–143, Sep. 2010.

[10] E. Barrett, C. Eustis, and R. B. Bass, "A dual-heat-pump residential heating system for shaping electric utility load," *IEEE Power Energy Technol. Syst. J.*, vol. 5, no. 2, pp. 56–64, Jun. 2018.

[11] N. B. Arias, S. Hashemi, P. B. Andersen, C. Treholt, and R. Romero, "Distribution system services provided by electric vehicles: Recent status, challenges, and future prospects," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 12, pp. 4277–4296, Dec. 2019.

[12] V. A. Evangelopoulos, I. I. Avramidis, and P. S. Georgilakis, "Flexibility services management under uncertainties for power distribution systems: Stochastic scheduling and predictive real-time dispatch," *IEEE Access*, vol. 8, pp. 38855–38871, 2020.

[13] T. Clarke, T. Slay, C. Eustis, and R. B. Bass, "Aggregation of residential water heaters for peak shifting and frequency response services," *IEEE Open Access J. Power Energy*, vol. 7, pp. 22–30, 2020.

[14] E. Mayhorn, S. Parker, F. Chassin, and R. Pratt, "Evaluation of the demand response performance of large capacity electric water heaters," Pacific Northwest Nat. Lab., Richland, WA, USA, Tech. Rep. PNNL-23527, 2015.

[15] K. Marnell, C. Eustis, and R. B. Bass, "Resource study of large-scale electric water heater aggregation," *IEEE Open Access J. Power Energy*, vol. 7, pp. 82–90, 2020.

[16] Y. J. A. Zhang, C. Zhao, W. Tang, and S. H. Low, "Profit-maximizing planning and control of battery energy storage systems for primary frequency control," *IEEE Trans. Smart Grid*, vol. 9, no. 2, pp. 712–723, Mar. 2018.

[17] A. Vafamehr, R. Moslemi, and R. Sharma, "Aggregation of BTM battery storages to provide ancillary services in wholesale electricity markets," in *Proc. IEEE 7th Int. Conf. Smart Energy Grid Eng. (SEGE)*, Aug. 2019, pp. 162–166.

[18] D. Wang, K. Ma, P. Wang, J. Lian, and D. J. Hammerstrom, "Frequency-domain flexibility characterization of heterogeneous end-use loads for grid services," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Feb. 2020, pp. 1–5.

[19] J. Alam, P. Balducci, K. Whitener, and S. Cox, "Energy storage control capability expansion: Achieving better technoeconomic benefits at Portland general electric's Salem smart power center," *IEEE Power Energy Mag.*, vol. 18, no. 2, pp. 69–80, Mar. 2020.

[20] F. Conte, S. Massucco, G.-P. Schiapparelli, and F. Silvestro, "Day-ahead and intra-day planning of integrated BESS-PV systems providing frequency regulation," *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1797–1806, Jul. 2020.

[21] K. B. Ardani, E. J. O'Shaughnessy, and P. D. Schwabe, "Coordinating distributed energy resources for grid services: A case study of Pacific Gas and electric," Nat. Renew. Energy Lab., Golden, CO, USA, Tech. Rep. NREL/TP-7A40-72108, 2018.

[22] A. Dubey, A. Bose, M. Liu, and L. N. Ochoa, "Paving the way for advanced distribution management systems applications: Making the most of models and data," *IEEE Power Energy Mag.*, vol. 18, no. 1, pp. 63–75, Jan. 2020.

[23] S. Poudel, P. Sharma, A. Dubey, and K. P. Schneider, "Advanced FLISR with intentional islanding operations in an ADMS environment using GridAPPS-D," *IEEE Access*, vol. 8, pp. 113766–113778, 2020.

[24] Y. P. Agalgaonkar, M. C. Marinovici, S. V. Vadari, K. P. Schneider, and R. B. Melton, "ADMS state of the industry and gap analysis," Pacific Northwest Nat. Lab., Richland, WA, USA, Tech. Rep. PNNL-26361, 2016.

[25] J. Wang, M. Blonsky, F. Ding, S. C. Drew, H. Padullaparti, S. Ghosh, I. Mendoza, S. Tiwari, J. E. Martinez, J. J. D. Dahdah, F. A. M. Bazzani, M. Baggu, M. Symko-Davies, C. Bilby, and B. Hannegan, "Performance evaluation of distributed energy resource management via advanced hardware-in-the-loop simulation," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Feb. 2020, pp. 1–5.

[26] M. Uslar, M. Specht, S. Rohjans, J. Trefke, and J. M. González, *The Common Information Model CIM: IEC 61968/61970 and 62325—A Practical Introduction to the CIM*. Cham, Switzerland: Springer, 2012.

[27] S. Widergren, R. Melton, A. Khandekar, B. Nordman, and M. Knight, "The plug-and-play electricity era: Interoperability to integrate anything, anywhere, anytime," *IEEE Power Energy Mag.*, vol. 17, no. 5, pp. 47–58, Sep. 2019.

[28] S. E. Widergren, D. Hardin, R. Ambrosio, R. Drummond, E. Gunther, G. Gilchrist, and D. Cohen, "Interoperability context-setting framework," Pacific Northwest Nat. Lab., Richland, WA, USA, Tech. Rep. PNNL-SA-53687, 2007.

[29] M. R. Knight, J. T. Kolln, S. E. Widergren, D. Narang, A. Khandekar, and B. Nordman, "Interoperability maturity model," Pacific Northwest Nat. Lab., Richland, WA, USA, Tech. Rep. PNNL-29683, 2020.

[30] N. S. Fernando, J. M. Acken, and R. B. Bass, "Developing a distributed trust model for distributed energy resources," in *Proc. IEEE Conf. Technol. Sustainability (SusTech)*, Apr. 2021, pp. 1–6.

[31] R. B. Melton, K. P. Schneider, E. Lightner, T. E. Mcdermott, P. Sharma, Y. Zhang, F. Ding, S. Vadari, R. Podmore, A. Dubey, R. W. Wies, and E. G. Stephan, "Leveraging standards to create an open platform for the development of advanced distribution applications," *IEEE Access*, vol. 6, pp. 37361–37370, 2018.

[32] T. Slay and R. B. Bass, "An energy service interface for distributed energy resources," in *Proc. IEEE Conf. Technol. Sustainability (SusTech)*, Apr. 2021, pp. 1–8.

[33] S. Ciraci, J. Daily, J. Fuller, A. Fisher, L. Marinovici, and K. Agarwal, "FNCS: A framework for power system and communication networks co-simulation," in *Proc. Symp. Theory Modeling Simulation-DEVS Integrative*, 2014, pp. 1–8.

[34] P. Top, R. Mast, S. Smith, J. Fuller, A. Fisher, J. Daily, B. Palmentier, D. Krishamurthy, H. Jain, and T. Elgindy, "Hiearchical engine for large scale infrastructure simulation," Lawrence Livermore Nat. Lab., Livermore, CA, USA, Tech. Rep. HELICS; 005239MLTPL00, 2017.

[35] R. C. Dugan and T. E. McDermott, "An open source platform for collaborating on smart grid research," in *Proc. IEEE Power Energy Soc. Gen. Meeting*, Jul. 2011, pp. 1–7.

[36] D. P. Chassin, J. C. Fuller, and N. Djilali, "GridLAB-D: An agent-based simulation framework for smart grids," *J. Appl. Math.*, vol. 2014, pp. 1–12, Jun. 2014.

[37] H. T. Nguyen, S. Battula, R. R. Takkala, Z. Wang, and L. Tesfatsion, "An integrated transmission and distribution test system for evaluation of transactive energy designs," *Appl. Energy*, vol. 240, pp. 666–679, Apr. 2019.

[38] E. Dunne, "Grid modernization funding initiative announced," *Control Eng.*, vol. 62, no. 2, pp. 21–22, 2016.

[39] A. Ellis, "DOE grid modernization initiative and Sandia R&D," Sandia Nat. Lab., Albuquerque, NM, USA, Tech. Rep. SAND2019-0595C, 2019.

[40] J. Xie, J. C. Bedoya, C.-C. Liu, A. Hahn, K. J. Kaur, and R. Singh, "New educational modules using a cyber-distribution system testbed," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5759–5769, Sep. 2018.

[41] M. Ostadijafari, A. Dubey, and N. Yu, "Linearized price-responsive HVAC controller for optimal scheduling of smart building loads," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3131–3145, Jul. 2020.

[42] H. M. Reeve, S. E. Widergren, R. G. Pratt, B. Bhattarai, S. Hanif, S. R. Bender, T. D. Hardy, and M. A. Pelton, "Distribution system operator with transactive (DSO+T) study: Main report," Pacific Northwest National Lab., Richland, WA, USA, Tech. Rep. PNNL-32170-1, 2022, vol. 1.

[43] (2017). *Transactive Energy Simulation Platform (TESP)*. [Online]. Available: https://tesp.readthedocs.io/en/latest/

[44] B. Palmintier, D. Krishnamurthy, P. Top, S. Smith, J. Daily, and J. Fuller, "Design of the HELICS high-performance transmission-distribution-communication-market co-simulation framework," in *Proc. Workshop Modeling Simulation Cyber-Phys. Energy Syst.*, 2017, pp. 1–6.

[45] Q. Huang, T. E. McDermott, Y. Tang, A. Makhmalbaf, D. J. Hammerstrom, A. R. Fisher, L. D. Marinovici, and T. Hardy, "Simulation-based valuation of transactive energy systems," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 4138–4147, Sep. 2019.

[46] *CIMHub: A Tool Set for Translating Electric Power Distribution System Models Between Various Formats*. Accessed: Jun. 1, 2021. [Online]. Available: https://github.com/GRIDAPPSD/CIMHub

[47] S. Poudel, G. D. Black, E. G. Stephan, and A. P. Reiman, "Admittance matrix validation for power distribution system models using a networked equipment model framework," *IEEE Access*, vol. 10, pp. 9108–9123, 2022.

[48] T. E. McDermott, E. G. Stephan, and T. D. Gibson, "Alternative database designs for the distribution common information model," in *Proc. IEEE/PES Transmiss. Distrib. Conf. Expo. (T&D)*, Apr. 2018, pp. 1–9.

[49] *GridAPPS-D Platform, API, and App Documentation*. Accessed: Jun. 1, 2021. [Online]. Available: https://gridappsd-training.readthedocs.io/en/latest/

[50] GridWise Architecture Council. (Aug. 2010). *Decision-Maker's Interoperability Checklist*. Version 1.5. [Online]. Available: http://www.gridwiseac.org/pdfs/gwac_decisionmakerchecklist_v1_5.pdf

[51] S. Sucic, B. Bony, and L. Guise, "Standards-compliant event-driven SOA for semantic-enabled smart grid automation: Evaluating IEC 61850 and DPWS integration," in *Proc. IEEE Int. Conf. Ind. Technol.*, Mar. 2012, pp. 403–408.

[52] R. B. Melton, K. P. Schneider, and S. Vadari, "GridAPPS-D a distribution management platform to develop applications for rural electric utilities," in *Proc. IEEE Rural Electr. Power Conf. (REPC)*, Apr. 2019, pp. 13–17.

SHIVA POUDEL (Member, IEEE) received the B.E. degree from the Department of Electrical Engineering, Pulchowk Campus, Kathmandu, Nepal, in 2013, the M.S. degree from the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA, in 2016, and the Ph.D. degree in electrical engineering from Washington State University, Pullman, WA, USA, in 2020. He is currently a Power Systems Research Engineer at the Pacific Northwest National Laboratory. His current research interests include distribution system modeling and analysis, resilience assessment, and transactive energy systems.

SEAN J. KEENE (Senior Member, IEEE) received the B.A. degree in media studies from the University of North Carolina Greensboro, in 2009, and the B.S. degree in electrical engineering from Portland State University, Portland, OR, USA, in 2020, where he is currently pursuing the M.S. degree in electrical and computer engineering.

Since 2019, he has been employed as an Undergraduate Research Assistant and is currently a Graduate Research Assistant at the Power Engineering Group, Portland State University. His research interests include smart grid technologies and integration, including development of advanced synchrophasor frequency response systems, and DER aggregation methods and modeling.

ROSHAN L. KINI (Member, IEEE) received the B.E. and M.E. degrees in electronics and telecommunication engineering from Mumbai University, Mumbai, India, in 2010 and 2014, respectively, and the Ph.D. degree in engineering from The University of Toledo, Toledo, OH, USA, in 2019. He is currently working as a Research Engineer with the Pacific Northwest National Laboratory, WA, USA. His current research interests include the management of DER, PV variability mitigation techniques, transactive control techniques, the applications of wide-bandgap semiconductor devices in power electronics, high-frequency power conversion techniques, and maximum power point tracking techniques. He serves as the Publication Chair and the Session Chair for the IEEE National Aerospace and Electronics Conference (NAECON).

SARMAD HANIF (Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Engineering and Technology Lahore, Pakistan, in 2009, and the M.Sc. and Ph.D. degrees from the Technical University of Munich, Germany, in 2013 and 2018, respectively. He is currently a Power Systems Research Engineer at the Pacific Northwest National Laboratory (PNNL). At PNNL, he has been involved in research activities related to transactive energy systems, energy storage systems, and marine renewable energies. His research interests include power systems optimization, analysis, and economics.

ROBERT B. BASS (Member, IEEE) received the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, VA, USA, in 2004.

Since 2011, he has been an Associate Professor with the Department of Electrical and Computer Engineering, Portland State University, Portland, OR, USA. His research addresses engineering challenges to the electric power system that arise from large-scale societal issues, such as natural disasters, climate change, and cyber-physical security threats. His research interests include service-oriented distributed energy resource aggregation for providing grid services, distributed trust systems for quantifying energy transaction trustworthiness, high-power electric vehicle charging impacts on distribution systems, and cyber-physical system threat detection.

JAIME T. KOLLN (Member, IEEE) is currently a Power Systems Engineer at the Pacific Northwest National Laboratory working in grid architecture, primarily on DER integration and grid edge interoperability. He is also a Technical Oversight Representative, which includes development of technical specifications for on-site and off-site procurements including energy storage deployments. Prior to joining PNNL, he was the Sole Proprietor and a System Designer of a small business focused on building controls. He research interests include distributed energy technology, power system communications, protection, controls, energy storage, and other DER integration. He is active in many standards development organizations, alliances, and committees. He is the Chair of the Modular Energy Storage Architecture (MESA) Alliance Testing and Certification Working Group.