Conversion and Validation of Distribution System Model from a QSTS-Based Tool to a Real-Time Dynamic Phasor Simulator

Manohar Chamana, Kumaraguru Prabakar, Bryan Palmintier and Murali M. Baggu
Power Systems Engineering Center
National Renewable Energy Laboratory
Golden, CO, USA
Manohar.Chamana@nrel.gov, Kumaraguru.Prabakar@nrel.gov, Bryan.Palmintier@nrel.gov, Murali.Baggu@nrel.gov

***Abstract*—A software process is developed to convert distribution network models from a quasi-static time-series tool (OpenDSS) to a real-time dynamic phasor simulator (ePHASORSIM). The description of this process in this paper would be helpful for researchers who intend to perform similar conversions. The converter could be utilized directly by users of real-time simulators who intend to perform software-in-the-loop or hardware-in-the-loop tests on large distribution test feeders for a range of use cases, including testing functions of advanced distribution management systems against a simulated distribution system. In the future, the developers intend to release the conversion tool as open source to enable use by others.**

***Keywords*-Power systems conversion tool; real-time simulator; distribution systems; advanced distribution management systems; enterprise integration**

I. Introduction

Interest has been growing in the electric power industry to perform real-time simulation tests to validate the effectiveness of prototype equipment and controllers prior to their deployment in the field. This enables testing in realistic routine and contingency scenarios without subjecting customers or actual grid equipment to experimental situations. Such real-time simulators go beyond traditional distribution simulation software, such as OpenDSS and Cyme, by enabling interaction with external hardware and software systems in real time.

Opal-RT's ePHASORSIM, a real-time simulator, can be utilized as a test platform for a wide range of cases, including simulating a large distribution feeder, or a collection of feeders, to evaluate various advanced distribution management system (ADMS) functionalities with a millisecond time step. ePHASORSIM can simulate tens of thousands of buses on one computer processing unit core with a time step of 10 milliseconds in real time. The real-time simulator calculates the root mean square and angle values of the voltage and current at every location/feeder node [1].

Real-time simulators have been previously used for a wide range of distribution system applications, including performing accurate closed-loop testing of protection and control hardware by utilizing input/output signals and measurements [2]. Extensive fault and operating events can be repeatedly simulated to test and validate the system response. In distribution systems, protection coordination can be tested with high penetrations of solar photovoltaic (PV) and protection/control devices either as software models or real power hardware-in-the-loop [3]. Another key use case is testing ADMS. Current ADMS-based vendors use model-based distribution automation and optimization, which requires network connectivity information acquisition along with state-estimation capability to calculate accurate power flows [4]. In this context, real-time phasor simulation can provide an emulation of an actual large-scale distribution system both to test an ADMS's ability to accurately estimate power flow and to evaluate its performance before implementation in the field.

However, one classic challenge has been building the network model in the real-time environment. In classic real-time applications with transient-scale (μs) time steps, real-time computation challenges typically force the use of highly reduced network representations, often limited to tens of nodes and assembled semi-manually; however, with real-time phasor simulation, it becomes possible to simulate much larger networks with thousands of nodes and without reducing such network complexity. At this scale, semi-manual processes are intractable, making it difficult to exercise the full capabilities of real-time phasor simulations. Although not available for real-time phasor simulations, extensive data sets are available for both test and real distribution systems for power flow and quasi-steady-state time-series (QSTS) analysis. A key challenge is that these data are incompatible formats that require considerable effort to convert for real-time phasor use. In [5], a similar process to convert a power systems network from the PSS®E model to the RSCAD model was discussed along with its challenges and limitations. In [6], a unified platform was

This work was supported by the U.S. Department of Energy, Office of Electricity Delivery & Energy Reliability, as part of the Grid Modernization Laboratory Consortium. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

2166-5478/17 $31.00 © 2017 IEEE
DOI 10.1109/GreenTech.2017.38

proposed in which the power system circuit model data are stored in a spreadsheet file with a defined forma, to be used to reconstruct the circuit in the destination software, such as ETAP, OpenDSS, Grid LabD, and DEW.

In this paper, we describe the considerations and process to automate the conversion from the QSTS power flow models to real-time phasor simulations. Because real-time phasor simulations are relatively new, few tools have been developed to automate such translations, and the existing tools tend to focus on the conversion from commercial distribution planning to real-time phasors (e.g., CyME to ePHASORSIM) [7]. In contrast, the described tool supports the conversion from the open-source OpenDSS, which provides access to a wide range of existing research system data available only in this format. CyME files are based on the comma-separated values (".csv") format in the Extensible Markup Language (XML) language; these are much simpler to convert than OpenDSS files, which are based on the common text data format. Additional data parsing from text-based files is required to extract model data from OpenDSS files. The feeder model information can be represented in numerous ways, which makes it more difficult to come up with a comprehensive converter than the existing CyME-to-ePHASORSIM converter.

The paper is arranged as follows: Section II describes the process of breaking up the topology of a typical distribution system into different files, Section III describes the different input spreadsheets required for ePHASORSIM, Section IV describes the conversion process, Section V presents a validation case study, and Section VI concludes.

II. Input Files From QSTS Tool

OpenDSS is an open-source simulation tool with the capability to perform QSTS simulations of large-scale distribution systems [8]. OpenDSS runs on feeder information imported from plain-text files that contain the topology of the distribution system in one file or many separate files. The OpenDSS database has a large repository of distribution feeder models in text format. Typically for large feeders, all the information is provided in separate OpenDSS files, as discussed below. The following individual files are available for large feeders that are utilized to build the input data spreadsheets to be interfaced with the ePHASORSIM simulator. The feeder model information described in this section is a high-level summary of an example feeder. More information on detailed feeder modeling is provided in the OpenDSS manual [7]. In case all the feeder information is provided in one ".dss" file, it is recommended to prepare individual files, as discussed below:

1) Line Codes

A file usually named "LineCodes.dss" defines the catalog of overhead (OH), underground (UG) cable, and service line configurations shared by various line segments in the three-phase line spreadsheet. Overhead and underground line configuration types are represented in symmetrical (positive, negative, and zero sequence) impedance format,[1] and the service/cable line configuration types are represented in the NxN matrix format, where N is the number of phases (typically three) in the example files. The information of units is utilized to obtain the ohms/units line impedance information of different configuration types, which are extracted from the "LineCodes.dss" file. The units could be ohms/km, ohms/kft, etc.

2) Lines

Files named "Lines.dss" contain the list of overhead and underground line segments in the network.

3) Services

Files named "Services.dss" contain the service cable information from the service transformers to the customer loads. Typically, "Services.dss" also contains the information for switches used in distribution systems for fault isolation and to provide flexibility in reconfiguring the networks.

4) Loads

In OpenDSS, both three-phase and single-phase loads can be modeled in a wide range of representations, with so-called "ZIP"—constant impedance (Z), current (I), and power (P)—representation. ZIP is the only representation currently supported by ePHASORSIM. This load information is typically available in "Loads.dss". The file contains the nominal kV, nominal kW, nominal KVAr, bus connection, and configuration.

5) Transformers

The service transformers at the customer end and voltage regulator information are typically found in "Transformers.dss". The number of windings, bus connections, nominal kV, nominal kVA of the primary and secondary sides, and information about the percentage of the resistive and reactance losses of the transformers are contained in this file.

6) Capacitors

Capacitors could be connected as either three-phase banks or as individual single-phase capacitors. The capacitor information is available in "Capacitors.dss". The file contains the nominal kV, nominal kVAr, bus connection, number of phases, and configuration information.

7) Distributed Generation/PV

Usually information about the distributed generators and PV are available in files called "Distributedgen.dss" and "ExistingPV.dss". The bus identification (ID) information, phase connections, nominal KV, nominal kW, and power factor are provided in these files.

8) Source/Substation

Information about the source/substation bus and voltage is extracted from "Substation.dss". Typically the information could be extracted from the substation transformer.

[1] OpenDSS line codes can also be represented geometrically, but the conversion from geometric formats to the real-time phasor is left for future development.

“Substation.dss” also contains information about the step-down transformer to feed the distribution network, including tap changers for feeder-head voltage regulation. Additional switches/circuit-breaker information can be available in “Substation.dss”.

III. Input Data Spreadsheets for the PHASOR-Domain Real-Time Hardware-in-the-Loop Simulator

ePHASORSIM’s phasor domain solver typically performs at a time step of few milliseconds, providing voltage and current information as phasor magnitudes and angles. The ePHASORSIM solver block is typically integrated into a larger Simulink/RT-LAB model. Any status change to the feeder components are fed externally in the Simulink environment to the ePHASORSIM solver block. The feeder topology is interfaced with ePHASORSIM in an Excel template saved as either “.XLS” or “.XLSX”. For distribution feeders, the system is based on unbalanced three-phase models. In the Excel workbook, all the components, their required parameters, and initial values are defined. The file includes pages with specific names that cannot be modified or changed. A short description of each page in the Excel spreadsheet is given below:

1) General

Contains the basic information of the power system. In this page, the base frequency and MVA are entered.

2) Pins

Outgoing information from the Simulink/RT-Lab blocks and incoming information to the Simulink/RT-Lab blocks are represented in the “Pins” sheet. There are two types of outgoing and incoming pins.

a) Outgoing: to define measurement probes or status monitoring in the power system (e.g., voltage of a bus or in-service status of a shunt device).

b) Incoming: to send operational commands to the simulator (e.g., applying faults) or fed in voltage, current, active, and reactive power signals from Simulink models or hardware-in-loop.

3) Bus

The “Bus” sheet contains all the bus ID information along with voltages and angle information for the initialization of the simulation run.

4) Vsource Three-Phase

Information about the substation bus and voltage with its series impedance is provided in the “Vsource Three-Phase” sheet.

5) Current Injector

The buses along with IDs of the current sources, such as distributed generators and PV, are described in the “Current Injector” sheet. Models for such devices are not currently directly available in the ePHASORSIM component library; instead, distributed generators need to be built as models in the Simulink environment, which injects currents based on the voltage measurements at the coupling node and power set points. The resulting current injection is then fed into the model at the nodes configured in the “Pins” sheet.

6) Load Three-Phase

The three-phase ZIP load is a combination of constant impedance, constant current, and constant power loads. The user needs to define the portion of each load part, from 1 to 100 for each load.

7) Shunt Three-Phase

The “Shunt Three-Phase” sheet is utilized to model shunt devices, such as capacitor banks or reactors.

8) Line Three-Phase

The three-phase pi-section model line information is provided in the “Line Three-Phase” sheet. Overhead, underground, and service lines are described in this sheet.

9) Transformer Three-Phase

Transformer information in the feeder is provided in the “Transformer Three-Phase” sheet. A voltage regulator can be implemented at the transformer by changing the tap position (limited to the range from -16 to 16) via an external signal. As with distributed generators, there is no native support for regulator or other controllers, so such systems must be included in the surrounding Simulink model.

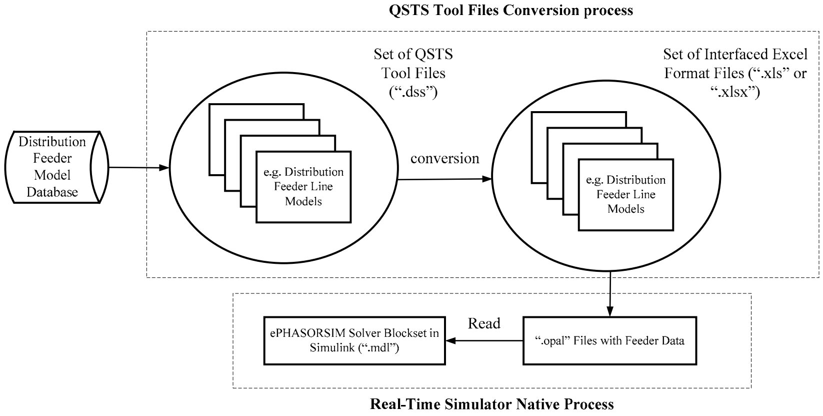
QSTS Tool Files Conversion process
Set of QSTS Tool Files (“.dss”)
Set of Interfaced Excel Format Files (“.xls” or “.xlsx”)
Distribution Feeder Model Database
e.g. Distribution Feeder Line Models
conversion
e.g. Distribution Feeder Line Models
ePHASORSIM Solver Blockset in Simulink (“.mdl”)
Read
“.opal” Files with Feeder Data
Real-Time Simulator Native Process

Figure 1. O en-source model file conversion to real-time simulator interfaced files.

10) Switch

Switches such as breakers and disconnects are described in the "Switch" sheet. A switch is open if the status is 0 and closed if the status is 1. Similar to the transformer tap control command, the on/off commands are given externally.

11) Bus Faults Three-Phase

The "Bus Faults Three-Phase" sheet is created to simulate different types of balanced and unbalanced faults in three-phase systems. Different types of faults can me entered as "ab" for a two-phase short circuit between phases "a" and "b", "ag" for a phase-to-ground fault, "abc" for a three-phase fault, etc. A fault case can be applied at a bus via an external signal by changing its status from 0 to 1.

A high-level description of how the OpenDSS files are utilized to create the Excel spreadsheets for ePHASORSIM is shown in Fig. 1.

IV. Conversion Process

Python 2.7 [9] was utilized to create a software tool for file conversion and data parsing into the ePHASORSIM interfaced Excel spreadsheets from the OpenDSS ".dss" files. As described later, this tool was demonstrated by converting the feeder data for the example feeder from the Electric Power Research Institute, K1 [10].

Much of data from the ".dss" files were extracted using regular expressions in Python. Regular expressions are a sequence of characters that define a search pattern, mainly for use in pattern matching with strings, such as find-and-replace operations. The Pandas library [11] was used for both the Excel spreadsheet (".xlsx") and text (".csv") file reading and writing to manage data structures and perform data analysis. Initial conditions for the voltage magnitude and angles were obtained from the "Bus" details ".csv" file created during the power flow analysis with OpenDSS. The specific conversion steps are as follows:

1) Using Python's regular expressions, all the line code information data was parsed into a dictionary format. Later on this dictionary was utilized to extract all the line configuration information for individual overhead, underground, and service lines of the distribution feeder. For example, in the K1 feeder the overhead line ID "OH_B4784" between buses "bus1=B4787.1.2.3" and "bus2=B4784.1.2.3" of length 32.308799 m calls the line configuration information for "linecode=OH-3X_477AAC_4/0AAACN" from the earlier created dictionary. The line is a three-phase bus connection. Similarly, for service lines the line configuration information is extracted from the dictionary to build the "Line-Three-Phase" sheet.

2) Using Python's regular expressions, if any switch data exist in "Services.dss" and "Substation.dss", they are extracted and imported as a single-phase bus to bus data along with the status of the switch: 0 for open, and 1 for closed. Any status-changing commands need to be given externally through the "Incoming" pins.

3) Typically, the distribution feeder consists of single-phase loads. In OpenDSS, it is enough to enter only single-phase load information; whereas in ePHASORSIM, the three-phase loads spreadsheet are entered as three-phase loads with zeros for nonexistent loads on phases. All the load information is extracted from the "Loads.dss" file and utilized to create the "Load Three-Phase" spreadsheet. There is an option to control the loads by importing the values into the "Pins" through external commands.

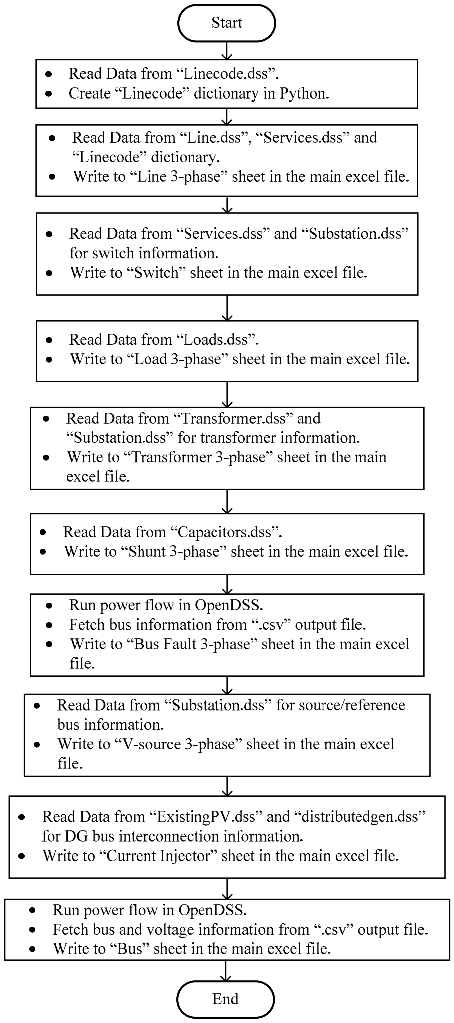
Start
Read Data from "Linecode.dss".
Create "Linecode" dictionary in Python.
Read Data from "Line.dss", "Services.dss" and "Linecode" dictionary.
Write to "Line 3-phase" sheet in the main excel file.
Read Data from "Services.dss" and "Substation.dss" for switch information.
Write to "Switch" sheet in the main excel file.
Read Data from "Loads.dss".
Write to "Load 3-phase" sheet in the main excel file.
Read Data from "Transformer.dss" and "Substation.dss" for transformer information.
Write to "Transformer 3-phase" sheet in the main excel file.
Read Data from "Capacitors.dss".
Write to "Shunt 3-phase" sheet in the main excel file.
Run power flow in OpenDSS.
Fetch bus information from ".csv" output file.
Write to "Bus Fault 3-phase" sheet in the main excel file.
Read Data from "Substation.dss" for source/reference bus information.
Write to "V-source 3-phase" sheet in the main excel file.
Read Data from "ExistingPV.dss" and "distributedgen.dss" for DG bus interconnection information.
Write to "Current Injector" sheet in the main excel file.
Run power flow in OpenDSS.
Fetch bus and voltage information from ".csv" output file.
Write to "Bus" sheet in the main excel file.
End

Figure 2. Flowchart of the data-parsing process to extract the distribution feeder model information.

4) The substation transformer is typically three phase, whereas service transformers are largely single phase. Even though single-phase transformers exist in the feeder, all transformers are modeled as three-phase systems. In the current version of ePHASORSIM (11.0.8.3), all the transformer information is extracted from the "Transformers.dss" and "Substation.dss" files and utilized to create the "Transformer Three-Phase" spreadsheet.

5) Similarly, both single-phase capacitors and three-phase capacitors are modeled as three-phase capacitor banks with zeros for nonexistent capacitor bank units on phases. All the capacitor information is extracted from the "Capacitors.dss" file and utilized to create the "Shunt Three-Phase" spreadsheet. The status of the capacitor on/off status is entered as 1/0.

6) Upon performing the load flow analysis on the distribution feeder by using OpendDSS, the output results of the voltages can be saved in ".csv" format. The ".csv" files contain all the bus names. The bus ID information read using the Pandas library [11] are utilized to create the "Bus Faults Three-Phase" sheet.

7) The ".csv" files contain the bus details along with the voltage magnitude and angles. The bus ID, magnitude, and angle information are utilized to create the "Bus" sheet.

A summary flowchart of the data-parsing process from the OpenDSS-based ".dss" files to the input data spreadsheets to be interfaced with the ePHASORSIM simulator is shown in Fig. 2. The time to convert the OpenDSS models to the ePHASORSIM models depends on the size of the network. Typically for a large system with approximately 1,000 buses it takes approximately 30 seconds to convert the distribution network model.

V. Validation Example

To validate the successful transfer of the feeder topology from OpenDSS files to the ePHASORSIM Excel spreadsheet, load flow analyses on both tools were performed, and the percentage error of the voltage magnitude and absolute error of the voltage angles at a few nodes were observed.

OpenDSS has an extensive repository of feeder models, from which feeder K1 was selected to perform the power flow validation. It is located in the southeastern United States with a 1-MW customer-owned PV system connected in the feeder service territory. The feeder contains approximately 1,300 buses. The K1 feeder serves approximately 500 commercial and residential class customers at a 13-kV level. The total sum of the maximum loads on the feeder is approximately 6 MW, which are supplied via a 28-mile long primary line. A load tap changer is provided at the substation along with a single capacitor bank located approximately mid-feeder [9]. A circuit diagram for this feeder is shown in Fig. 3. The figure also shows the location of the 12 buses selected for tabular comparison.

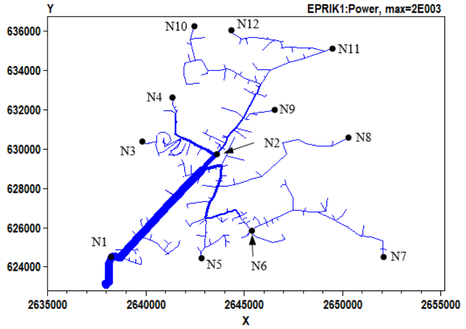

Figure 3. K1 feeder circuit diagram.

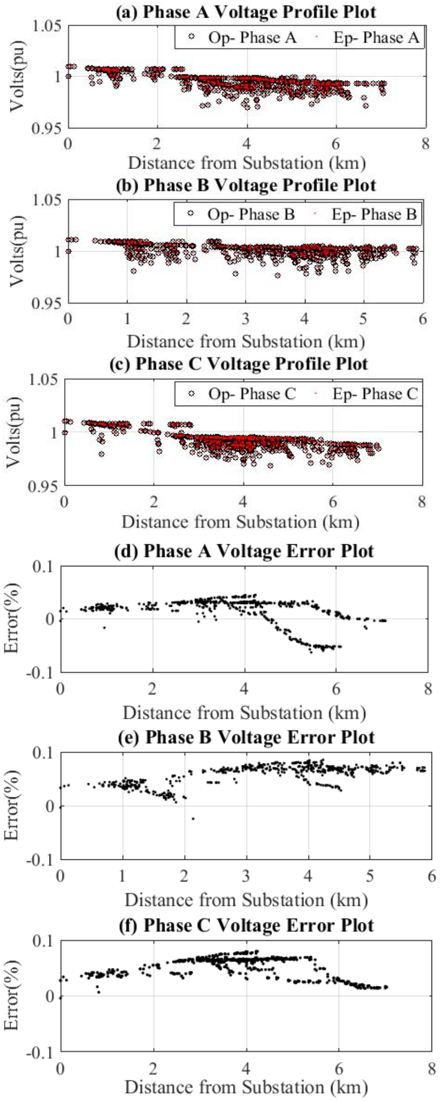

Figure 4. OpenDSS and ePHASORSIM voltage magnitude results comparison: (a) Phase A voltage profiles overlap, (b) Phase B voltage profiles overlap, (c) Phase C voltage profiles overlap, (d) Phase A % voltage error, (e) Phase B % voltage error, and (f) Phase C % voltage error.

The first step was to run the base-case snapshot (single time period) load-flow analysis in OpenDSS and store the resulting output file.

Next, the topology conversion from the OpenDSS files to the ePHASORSIM Excel spreadsheets (described in Section IV) was performed using a code written in Python. As mentioned previously, the real-time phasor simulator reads feeder information from the data spreadsheets.

Then, ePHASORSIM was run for comparison. Because it is a dynamic phasor simulator, single-snapshot power flow is not supported; instead, the simulation was run for a few seconds for the base case with constant conditions. The

TABLE I. K1 FEEDER LOAD FLOW RESULTS COMPARISON BETWEEN TWO SOFTWARE TOOLS

		Voltage					
		OpenDSS		ePHASORSIM		Percentage Error	Absolute Error
K1 Bus Name (Assigned ID) Bus Phases	**Ø**	**Magnitude (pu)**	**Phase Angle (Degrees)**	**Magnitude (pu)**	**Phase Angle (Degrees)**	**Magnitude (%)**	**Phase Angle (Degrees)**
10573638 (N1) 3-Ø	a	1.008	-1.8	1.008	-1.770	0.02%	0.030
	b	1.010	-121.6	1.010	-121.607	0.04%	-0.007
	c	1.009	118.3	1.009	118.355	0.06%	0.055
34086008 (N2) 3-Ø	a	0.999	-2.9	0.999	-2.928	0.03%	-0.028
	b	1.004	-122.2	1.004	-122.235	0.06%	-0.035
	c	0.995	117.4	0.996	117.418	0.06%	0.018
20344372 (N3) 1-Ø	b	0.998	-122.3	0.998	-122.251	0.03%	0.049
10540314 (N4) 3-Ø	a	0.988	-3.0	0.989	-2.949	0.04%	0.051
	b	0.995	-122.2	0.995	-122.247	0.08%	-0.047
	c	0.986	117.4	0.987	117.432	0.08%	0.032
10549015 (N5) 1-Ø	b	1.002	-122.4	1.003	-122.36	0.07%	0.036
34085341 (N6) 3-Ø	a	0.996	-3.1	0.996	-3.063	0.03%	0.037
	b	1.002	-122.4	1.003	-122.394	0.07%	0.006
	c	0.994	117.3	0.995	117.325	0.07%	0.025
10549294 (N7) 3-Ø	a	0.996	-3.1	0.997	-3.062	0.03%	0.038
	b	1.002	-122.4	1.003	-122.393	0.07%	0.007
	c	0.994	117.3	0.995	117.325	0.07%	0.025
10532582 (N8) 1-Ø	c	0.993	117.3	0.993	117.326	0.02%	0.026
20494026 (N9) 1-Ø	c	0.994	117.3	0.994	117.347	0.06%	0.047
10559391 (N10) 1-Ø	a	0.993	-3.2	0.992	-3.258	-0.05%	-0.058
1055852 (N11) 3-Ø	a	0.999	-3.2	1.000	-3.148	0.03%	0.052
	b	1.004	-122.1	1.005	-122.124	0.08%	-0.024
	c	0.991	117.2	0.992	117.179	0.06%	-0.021
20122385 (N12) 1-Ø	c	0.988	117.1	0.988	117.112	0.01%	0.012

phase voltage magnitudes across the feeder nodes are plotted and shown in Fig. 4. The phase voltage magnitudes from the OpenDSS power flow results (Op- Phase A, Op- Phase B, and Op- Phase C) are close and overlap with the voltage magnitudes from the ePHASORSIM simulation (Ep- Phase A, Ep- Phase B, and Ep- Phase C), as shown in Figs. 4 (a), (b) and (c). The voltages are plotted with respect to the nodal distances from the substation node. The errors in phase voltage magnitudes between the two software tools are shown in Figs. 4 (d), (e), and (f). The errors are within the 0.1% standard, which validates the power flow results from the ePHASORSIM simulator.

The voltage magnitudes and angles at the 12 buses were observed using virtual output scopes, and the voltage magnitude percentage error and voltage angle absolute error between the ePHASORSIM simulations and OpenDSS load-flow analysis results were calculated and tabulated

As shown in Table I, the errors at the 12 selected buses were very small, with most errors at or below 0.1% and an average absolute angle difference of 0.0314°. Of the selected nodes, N4 was found to have the highest voltage magnitude mismatch error: 0.085%; whereas N10 was found to have the highest voltage angle mismatch error: -0.058°. These low errors show that the feeder topology data from OpenDSS were successfully transferred to ePHASORSIM.

VI. CONCLUSIONS

This paper presents a software conversion and validation process from a QSTS-based distribution software tool (OpenDSS) to a phasor-domain real-time simulator (ePHASORSIM). The tool described in the paper will be released as open source, and this discussion should also prove useful for researchers who intend to perform similar conversions. It will take a long time to manually create any large distribution circuit in the real-time simulator containing hundreds of lines, cables, loads, etc., but by automating the conversion process it would take approximately 30 seconds to create a distribution circuit model in the real-time simulator, hence saving manual labor and time.

After completing the conversion, the resulting ePHASORSIM input Excel spreadsheets were used to run a base case power flow. The errors for 12 selected buses were very small, with most at or below 0.1% (max 0.08%) and an average absolute angle difference of 0.0314° (absolute max 0.058°). This demonstrates the successful transfer of the feeder topology from OpenDSS to ePHASORSIM.

Looking ahead, this capability makes it possible to easily import existing models from the large set of existing OpenDSS models into ePHASORSIM for three-phase dynamic analysis and hardware-in-the-loop testing.

Acknowledgment

This work was supported by the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308 with the National Renewable Energy Laboratory. Funding provided by the DOE Office of Electricity Delivery & Energy Reliability. The authors would also like to acknowledge Electric Power Research Institute for use of its distribution feeder circuit models and analysis tool to accomplish the conversion and validation.

References

[1] V. Jalili-Marandi, J. Belanger, and F. J. Ayres, "Model-in-the-loop real-time simulation in phasor domain," in *Proc. 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, Istanbul, 2014, pp. 2280–2284.

[2] M. S. Almas, R. Leelaruji, and L. Vanfretti, "Over-current relay model implementation for real time simulation and hardware-in-the-loop (HIL) validation," in *Proc. IECON 2012—38th Annual Conference on IEEE Industrial Electronics Society*, Montreal, QC, 2012, pp. 4789–4796.

[3] F. Katiraei, K. Atef, K. Salem, T. Chang, B. Koosha, and A. Momeni, "Power hardware-in-loop testing of advanced inverter functionalities," In *Proc. 2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, Portland, OR, 2016, pp. 1381–1386.

[4] G. Cochenour, R. Ochoa, and V. Rajsekar, "Distribution network model readiness for advanced distribution management systems," In *Proc. 2014 IEEE PES T&D Conference and Exposition*, Chicago, IL, USA, 2014, pp. 1–5.

[5] H. Ravindra, M. O. Faruque, M. Steurer, M. Andrus, and M. K. H. Pulok, "Conversion of PSS®E models into RSCAD models: Lessons learned," In *Proc. IECON 2014—40th Annual Conference of the IEEE Industrial Electronics Society*, Dallas, TX, 2014, pp. 3743–3749.

[6] A. Khoshkbar-Sadigh, M. Heydari, M. Tedde, K. Smedley, R. Arghandeh, and A. von Meier, "A unified platform enabling power system circuit model data transfer among different software," presented at the 2015 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, 2015, pp. 1–5.

[7] ePHASORsim User Guide, Version 1.2 [Online], Available: http://www.opal-rt.com/resource-center/.

[8] OpenDSS Program [Online], Available: https://sourceforge.net/projects/electricdss/.

[9] Python Software Foundation. Python Language Reference, Version 2.7. Available at http://www.python.org.

[10] "Distributed PV monitoring and feeder analysis," Web Page, EPRI.com. Available: http://dpv.epri.com/feeder_k.html.

[11] Wes McKinney, Data Structures for Statistical Computing in Python: Proceedings of the 9th Python in Science Conference. 2010, 51–56.