

THESIS

AN OPEN SOURCE INTERFACE FOR DISTRIBUTION SYSTEM MODELING IN POWER
SYSTEM CO-SIMULATION APPLICATIONS AND TWO ALGORITHMS FOR
POPULATING FEEDER MODELS

Submitted by

Rahul Kadavil

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2017

Master's Committee:

Advisor: Siddharth Suryanarayanan

Howard J. Siegel
Thomas H. Bradley

Copyright by Rahul Kadavil 2017

All Rights Reserved

ABSTRACT

AN OPEN SOURCE INTERFACE FOR DISTRIBUTION SYSTEM MODELING IN POWER SYSTEM CO-SIMULATION APPLICATIONS AND TWO ALGORITHMS FOR POPULATING FEEDER MODELS

The aging electric infrastructure power system infrastructure is undergoing a transformative change mainly triggered by the large-scale integration of distributed resources such as distributed generation, hybrid loads, and home energy management systems at the end-use level. The future electric grid, also referred to as the Smart Grid, will make use of these distributed resources to intelligently manage the day to day power system operations with minimum human intervention. The proliferation of these advanced Smart Grid resources may lead to coordination problems to maintain the generation-demand balance at all times. To ensure their safe integration with the grid, extensive simulation studies need to be performed using distributed resources.

Simulation studies serve as an economically viable alternative to avoid expensive failures. They also serve as an invaluable platform to study energy consumption behavior, demand response, power system stability, and power system state estimation. Traditionally, power system analysis has been performed in isolated domains using simulation tools for the transmission and distribution systems. Moreover, modeling all the power system assets using a single power system tool is difficult and inconclusive. From the Smart Grid perspective, a common simulation platform for different power systems analysis tools is essential. A co-simulation framework enables the interaction of multiple power system tools, each modeling a single domain in detail, to run simultaneously and provide a holistic power system overview.

To enable the co-simulation framework, a data exchange platform between the transmission and distribution system simulators is proposed to model transmission and distribution assets on different simulation testbeds. A graphical user interface (GUI) is developed as a front-end tool for the data exchange platform and makes use of two developed algorithms that simplifies the task of:

1. modeling distribution assets consisting of diverse feeder datasets for the distribution simulator and balanced three-phase level assets for the transmission system simulator, and
2. populating the distribution system with loads having stochastic profiles for time-step simulations.

The load profiles used in the distribution system models are created using concepts from one-dimensional random walk theory to mimic the energy consumption behavior of residential class of consumers. The algorithms can simulate large scale distribution system assets linked to a transmission system for co-simulation applications.

The proposed algorithms are tested on the standard test system – Roy Billinton Test System (RBTS) to model detailed distribution assets linked to a selected transmission node. Two open source power system simulators—MATPOWER[®] and GridLAB-D[®] are used for the transmission and distribution simulation process. The algorithms accurately create detailed distribution topology populated with 4026 residential loads expanded from the transmission node, bus 2 in RBTS.

Thus, an automated modeling of power system transmission and distribution assets is proposed along with its application using a standard test system is provided.

ACKNOWLEDGMENTS

I would like to sincerely thank my advisor Dr. Siddharth Suryanarayanan for his continued guidance and support throughout my study at Colorado State University. He has always encouraged me with many opportunities to highlight my skills which I would never have explored on my own. I am truly grateful for his patience with helping me improve my technical writing skills. I would also like to thank Dr. Timothy Hansen for his valuable guidance and inputs provided while working on my thesis. I appreciate Dr. Howard J. Siegel, and Dr. Thomas H. Bradley for agreeing to be a part of my thesis committee. I am thankful to the Energy Institute for sponsoring the thesis work with the seed grant. Further, I would like to extend my sincere regards to National Renewable Energy Laboratory (NREL), Bonneville Power Administration (BPA), and US Department of Energy (DOE) to acknowledge their financial support under the contract UGA-0-41027-00 for my master thesis study. I would also like to thank my fellow power system researchers at the Advanced Power Engineering Laboratory (APEL)—Fathalla Eldali, Turki Alaqeel, Mayank Panwar, Sulaiman Almohaimeed, and Abhimanyu Das Choudhary for their advice and suggestions during the various phases of this thesis work.

Lastly, I would like to thank my parents, family, and friends for their love and support to pursue the M.Sc. degree in Electrical Engineering from Colorado State University.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS.....	iv
TABLE OF CONTENTS	v
LIST OF TABLES	viii
LIST OF FIGURES.....	ix
INTRODUCTION.....	1
1.1 A PRIMER ON THE SMART GRID INITIATIVE AND MOTIVATION FOR RESEARCH WORK.....	1
1.2 OBJECTIVE OF THE RESEARCH	2
1.3 ORIGINAL CONTRIBUTIONS.....	3
1.4 SOFTWARE TOOLS.....	4
1.5 ORGANIZATION OF THESIS	5
DEx.py - DATA EXCHANGE PLATFORM BETWEEN POWER SYSTEM SIMULATORS	6
2.1 END-USER AND POWER NETWORK MODELING USING DEx.py	6
2.2 THE DEx.py INTERFACE	7
2.3 DISTRIBUTION TOPOLOGY CREATION USING DEx.py	11
a) DATA PROCESSING.....	11
b) DISTRIBUTION NETWORK CONFIGURATION	16

c) SIMULATION RUNTIME FILES	26
2.4 CONCLUDING REMARKS	27
AN ALGORITHMIC APPROACH FOR CREATING DIVERSE STOCHASTIC FEEDER	
DATASETS FOR POWER SYSTEM CO-SIMULATIONS.....	
3.1 INTRODUCTION	29
3.2 A MATHEMATICAL MODEL FOR CREATING THE DISTRIBUTION SYSTEM	
TOPOLOGY	31
a) TOPOLOGY OF THE DISTRIBUTION NETWORK.....	31
b) PARAMETERS FOR THE DISTRIBUTION TOPOLOGY	33
c) DISTRIBUTION FEEDER GENERATION ALGORITHM.....	34
3.3 STOCHASTIC LOAD PROFILE GENERATION FOR RESIDENTIAL LOADS .	35
a) OVERVIEW	35
b) STOCHASTIC LOAD PROFILE GENERATION FOR H_{actual}^{new} RESIDENTIAL	
LOADS USING THE RANDOM WALK PROCESS	36
c) STOCHASTIC LOAD PROFILE GENERATION ALGORITHM.....	37
3.4 SIMULATION STUDY	38
a) OVERVIEW	38
b) THE CANDIDATE SYSTEM	38
c) SIMULATION PARAMETERS	38
d) SYSTEM LOAD PROFILE	41

e) RESULTS	42
3.5 CONCLUSIONS	44
CONCLUSION AND FUTURE WORK.....	45
4.1 CONCLUSION	45
4.2 FUTURE WORK	46
REFERENCES.....	47
APPENDIX A	50
APPENDIX B	51
APPENDIX C	53

LIST OF TABLES

Table 2.1: The RBTS load points modeled in PowerWorld® [13].....	16
Table 2.2: Distribution system line configuration set up in DEx.py	20
Table 2.3: Setup of feeders, load points, and parallel configurations for each RBTS bus	22
Table 2.4: Deviation from H_{ideal} created by p parallel connection round-offs at each load point	23
Table 2.5: Final house objects (H_{actual}^{new}) placed at each RBTS bus by re-arranging the number of parallel connections at each load point	26
Table 3.1: RBTS bus 2 distribution network topology from [13].....	41

LIST OF FIGURES

Figure 1.1: Time-line of research.....	3
Figure 2.1: DEx.py as a pre-simulation network and end-user model generator	7
Figure 2.2: Main user interface of DEx.py	8
Figure 2.3: Generating the run time simulation files for MATPOWER [®] and GridLAB-D [®]	9
Figure 2.4: Line parameter configuration for distribution modeling in DEx.py	10
Figure 2.5: Load configuration for distribution system modeling in DEx.py	10
Figure 2.6: General options for distribution modeling in DEx.py	11
Figure 2.7: The operational flowchart for DEx.py	12
Figure 2.8: Three bus test system case saved as IEEE common format data in PowerWorld [®]	13
Figure 2.9: DEx.py console output from parsing an IEEE common format file for three-bus test system	13
Figure 2.10: DEx.py console output from parsing an IEEE common format file for 30 bus test system [21].....	14
Figure 2.11: The RBTS network modeled in PowerWorld [®]	15
Figure 2.12: Feeder configuration format for DEx.py to parse RBTS bus 2 network model.....	19
Figure 2.13: Tree diagram of the distribution topology for RBTS bus 2 created by DEx.py.....	19
Figure 2.14: Before and after count of house objects or residential loads placed at each RBTS load bus using DFGA.....	27
Figure 2.15: Simulation output of the distribution topologies created for each RBTS bus in GridLAB-D [®]	28
Figure 3.1: The distribution topology linked from a transmission node i	32
Figure 3.2: The low voltage network topology for each load point (LP).....	33

Figure 3.3: Single line diagram of the RBTS candidate system from [13].....	40
Figure 3.4: System load, $sys(t)$, for bus 2 of RBTS with peak at 17MW	41
Figure 3.5: $\beta_i^{pre}(t)$ for 5 randomly selected residential loads out of 4026	42
Figure 3.6: $\beta_i(t)$ for 5 randomly selected residential loads out of 4026	42
Figure 3.7: Figure 3.7: 3D plot of the sum of 4026 residential loads generated using $\beta_i^{pre}(t)$ (shown as dotted plot) and $\beta_i(t)$ (shown as hatched plot) compared with the custom created system load $sys(t)$ (shown as starred plot)	43

CHAPTER 1

INTRODUCTION

1.1 A PRIMER ON THE SMART GRID INITIATIVE AND MOTIVATION FOR RESEARCH WORK

The electric grid is undergoing a paradigm shift towards a complex cyber-physical entity, also known as the Smart Grid, to automate and manage the growing needs for electricity demand in the 21st century [1, 2]. The Smart Grid refers to the class of advanced digital technologies that only will allow for bidirectional communication between grid operators and consumers but also be spatially aware for quick and efficient management of the changing electricity demand [3]. The transformation has the potential to enable consumers to have more choice and control to better manage their energy demand and also reliably maintain grid security while minimizing the operational cost [4]. Additionally, the Smart Grid will be resilient enough to automatically detect and isolate contingencies at the physical and cyber level before they exacerbate to a cataclysmic cascade [5].

Smart Grid technologies will be able to take advantage of the large number of distributed energy resources (DER) to supply energy when power is not available from traditional sources [6]. While technology on both sides of the meter is advancing, the interaction of large number of DERs at the grid interface to mitigate the energy demand-supply imbalance may prove to be challenging to coordinate due to the number of involved entities [7]. The integration of these new technologies along with their control algorithms requires complex control strategies and extensive testing with respect to power quality and grid stability. To this effect, simulation testbeds offer an economically viable alternative to test the large-scale deployment of DER in a simulated Smart Grid setting [8].

Before the advent of the Smart Grid, the tools used for power system analysis modeled the transmission and the distribution network in isolated domains. With the penetration of more Smart Grid technologies such as distributed photovoltaics (PV), and electric vehicles (EV) at the distribution level, their complex interaction with the bulk power network cannot be satisfactorily analyzed by modeling them in a single power system domain [9]. This necessitates the development of a co-simulation environment where multiple power system modeling tools, each modeled to great detail in their specific domain, can interact and run simultaneously to analyze the impact of the new Smart Grid distribution system on the transmission grid [10].

The research work aids to facilitate the reader with easy comprehension of ideas, procedures, and algorithms pertaining to the modeling and simulation of detailed distribution topologies linked to a transmission system for power system co-simulation applications. The tools presented in this work cater to the distribution network creation and population for the co-simulation environment. As a part of the network creation process, the distribution level feeders are populated with detailed end-use assets and load profiles. The distribution system created can be used as a simulation testbed to integrate different DER technologies and perform long term co-simulations over a wide time horizon to study their impacts and develop new reliability metrics that will evaluate system performance.

1.2 OBJECTIVE OF THE RESEARCH

The main objective of this research is to develop the tools to automate the swift creation of detailed power system network models in the transmission and distribution domain. The tool presented in this research work is called the Data Exchange Model or DEx.py (pronounced as dex dot py) that combines a top-down approach for distribution network model creation and bottom-up approach for populating the network with load profiles to aid in the co-simulation process. The

other important objectives also include the application of DEx.py on a standard IEEE test system that serves as the candidate system and the demonstration of the results. Figure 1.1 depicts the research time-line for the milestones achieved in this thesis work.

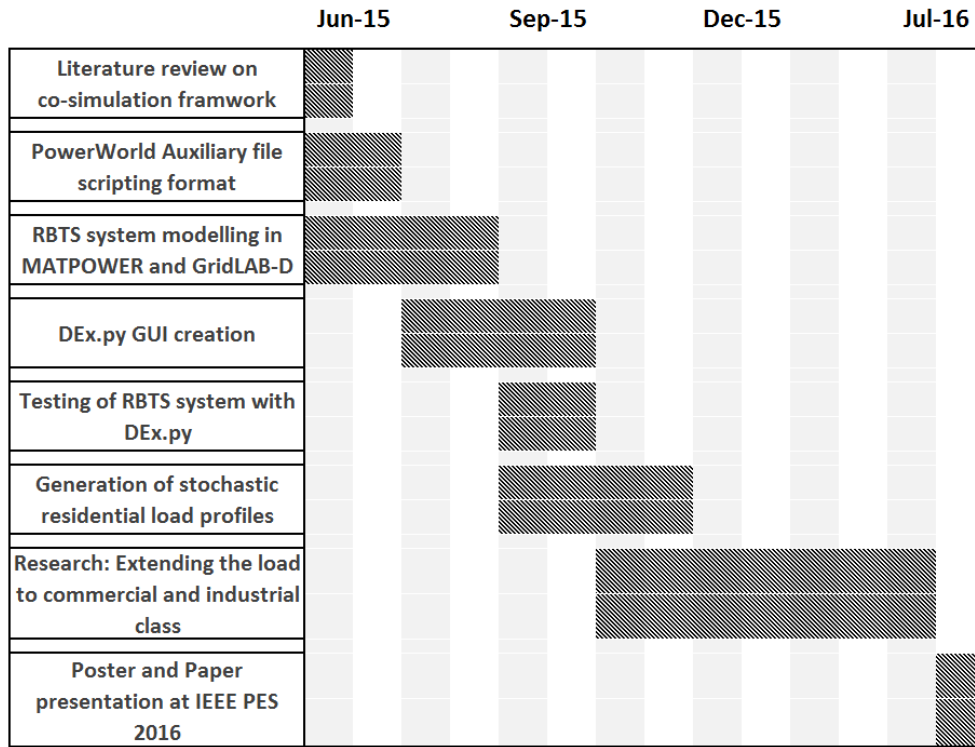


Figure 1.1: Time-line of research

1.3 ORIGINAL CONTRIBUTIONS

The research work performed in this thesis is original and contributes to multiple domains—power system modeling, power flow studies, power system operations, and random walk theory applications to develop tools to model the distribution system asset in the emerging Smart Grid. Two articles based on the research work have been published at the time of writing this thesis [11, 12]. The following is a summary of the contributions of the work performed in the thesis:

1. Comparison of existing tools for modeling distribution system assets that include stochastic load models over a wide time range
2. Development of a distribution network modeling algorithm that uses abstract mathematical approach for network creation
3. Development of a stochastic load models that can use real or synthetic load profiles for modeling the residential load class in the distribution system
4. Development of a graphical user interface for easy configuration of user inputs to perform the distribution asset population
5. Application of the proposed research work on a standard test system are outlined

1.4 SOFTWARE TOOLS

The Roy Billinton Test System (RBTS) serves as the candidate system [13]. To set up the co-simulation variables, the candidate test system modeled in PowerWorld[®], a commercial three phase load flow analysis platform serves as the physical network [14]. Using open source tools, the physical network is simultaneously modeled at the transmission level for use in MATPOWER[®] and at its load points at the distribution level to run in GridLAB-D[®], an agent-based distribution system simulator [15, 16]. The network information in PowerWorld[®] is extracted using its Auxiliary File Scripting tool to serve as an input to DEx.py [17]. Three phase network information in the form of IEEE common data format (CDF) is also a supported input to DEx.py [18]. The data exchange model script written in Python 3.0 then proceeds forward to create the runtime configuration scripts for the transmission and distribution simulator [19]. The instructions and the source codes to run these tools are made available in the public domain at the following webpage:

<http://www.engr.colostate.edu/sgra/>

1.5 ORGANIZATION OF THESIS

The work in this thesis is organized into four chapters. Chapter 2 provides an overview of the power system model creation tool developed in Python for setting up the co-simulation environment.

Chapter 3 explains the two-step algorithm developed for creating scalable distribution system in a bottom-up approach to match a connected load at a transmission bus and populating the distribution system with active power loads possessing time varying load profiles. The application of the developed algorithms on a selected bus in the candidate system – the Roy Billinton Test System (RBTS) is presented.

Chapter 4 provides the conclusion and recommendations for future work.

CHAPTER 2

DEx.py - DATA EXCHANGE PLATFORM BETWEEN POWER SYSTEM SIMULATORS¹

This chapter presents an overview of the data exchange tool and provides detailed features and methodologies followed for rapid customization of distribution network parameters linked from a transmission system.

2.1 END-USER AND POWER NETWORK MODELING USING DEx.py

In the emerging Smart Grid, many new technologies are being introduced at the distribution-level [20]. These new end-user assets, such as distributed PV solar, smart appliances, and smart heating, ventilation and air conditioning (HVAC) systems, may impact the transmission grid and wholesale market level unlike in the past. To properly determine the impact of these technologies and their controls on power system network and market operations, we need methods for populating distribution feeders with differing amounts of these technologies stochastically while still maintaining a realistic aggregate load curve at the point of common coupling (PCC) with the transmission level. To that end, we developed the data exchange model (DEx.py) in the programming language Python, shown in Figure 2.1. The developed tool can accept customized network configurations to create distribution topologies and custom load profile data to base the profiles for each of the connected loads.

¹This chapter is a verbatim reproduction of portions of a co-authored article published in the IEEE Electrification Magazine as cited in the reference [11]. The required permissions for re-use of the material have been obtained from the copyright holders and are included in the Appendix A. The numbering of the figures and tables has been modified to satisfy the formatting requirements of the thesis.

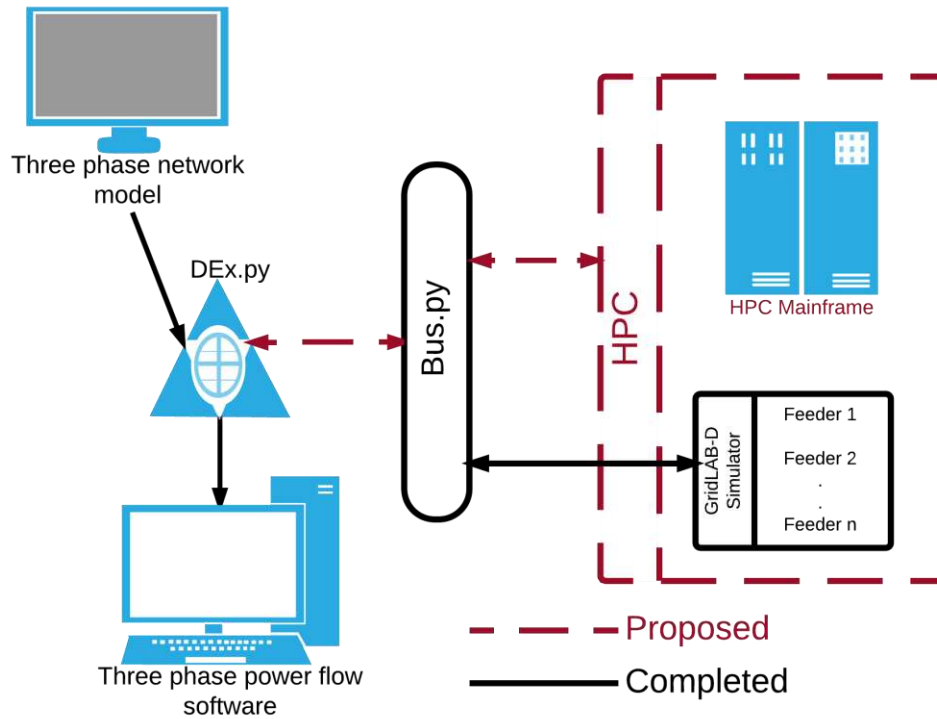


Figure 2.1: DEx.py as a pre-simulation network and end-user model generator

2.2 THE DEx.py INTERFACE

DEx.py provides a graphical user interface, as shown in Figures 2.2-2.6, developed in the Python environment for rapid creation of distribution network parameters linked from a transmission system. Figure 2.3 shows the format of the transmission and distribution network models created by DEx.py. The output file formats are limited to MATPOWER[®] and GridLAB-D[®] simulators due to their open-source nature. In the initial phase, DEx.py performs the following two functions:

1. distribution network creation, and
2. populating the network with end-use loads.

The distribution network thus created, forms an input to GridLAB-D[®], and is linked to a transmission bus present in a transmission system modeled in MATPOWER[®].

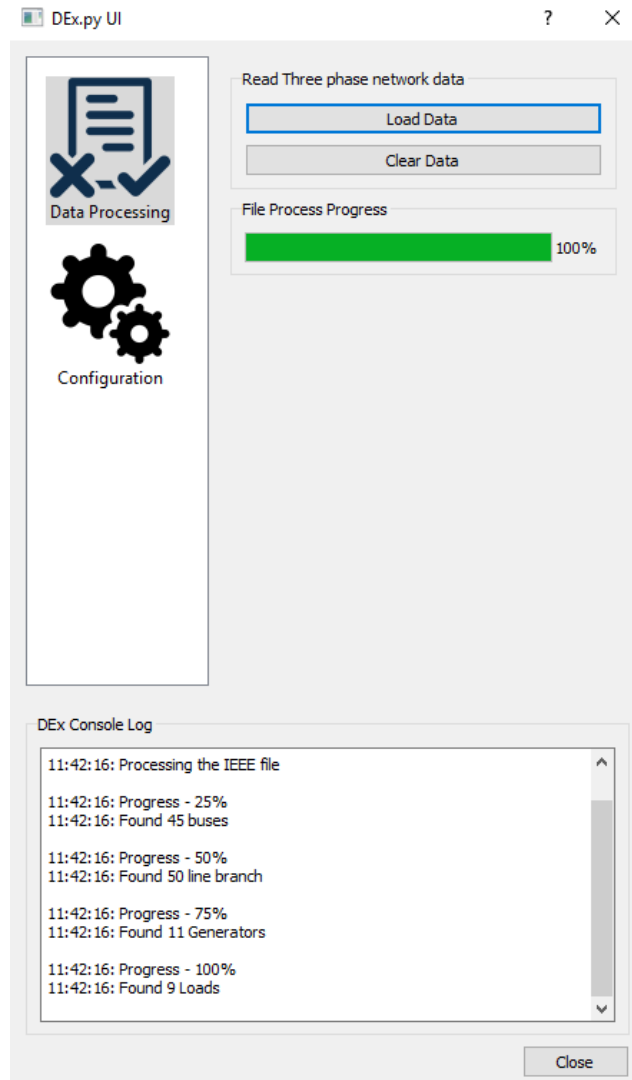


Figure 2.2: Main user interface of DEx.py

The distribution topology is created by setting the upper bounds for the number of primary feeders, sub-feeders per primary feeder, the load points, distribution transformers, and cables per

load point for each transmission bus. Figure 2.4-2.5 presents an overview of the detailed customization options available to the user for creating a distribution topology. DEx.py currently populates the distribution system with residential load class.

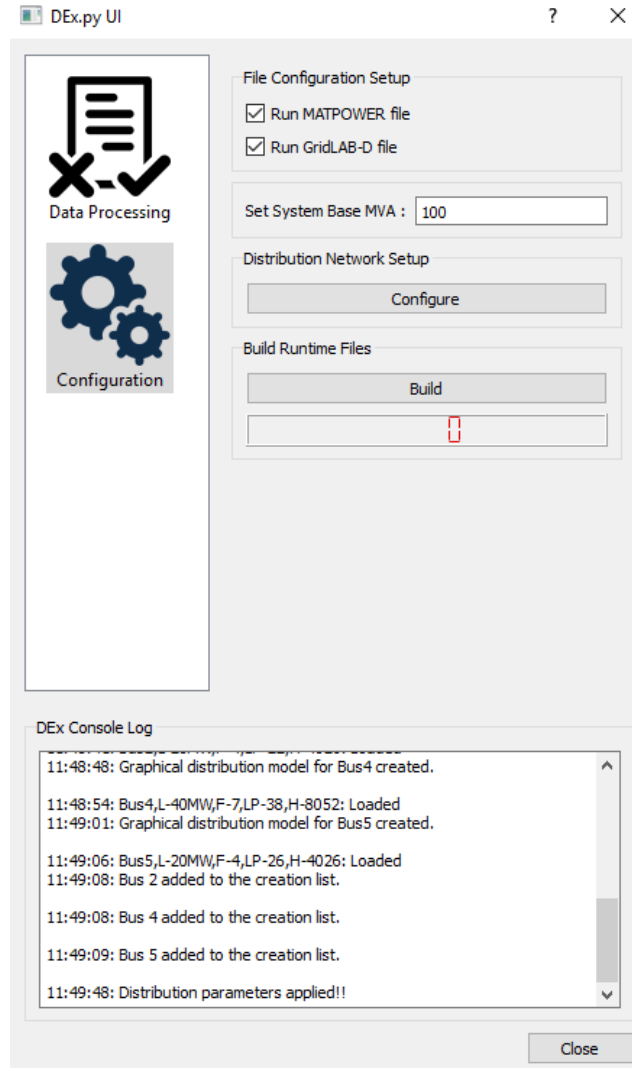


Figure 2.3: Generating the run time simulation files for MATPOWER® and GridLAB-D®

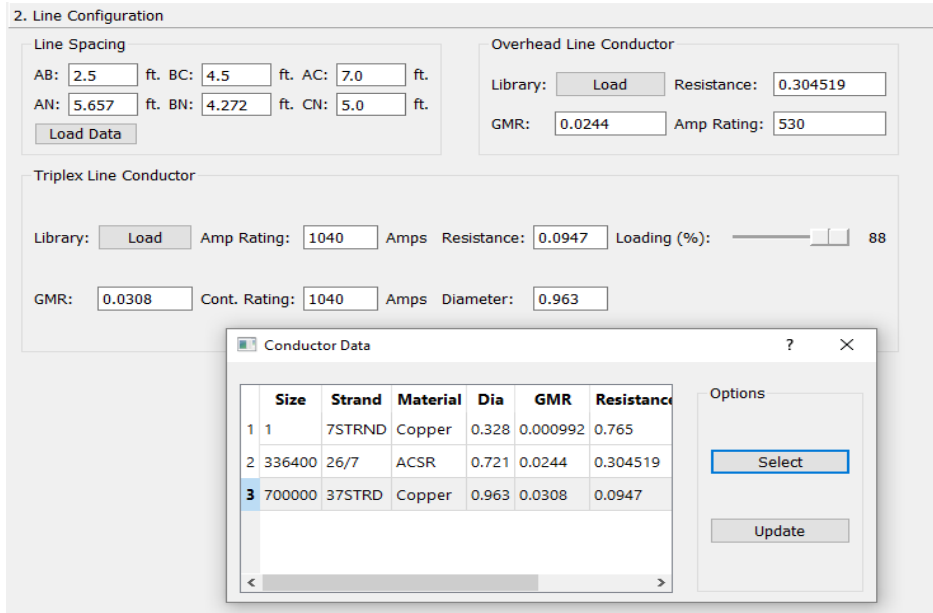


Figure 2.4: Line parameter configuration for distribution modeling in DEx.py

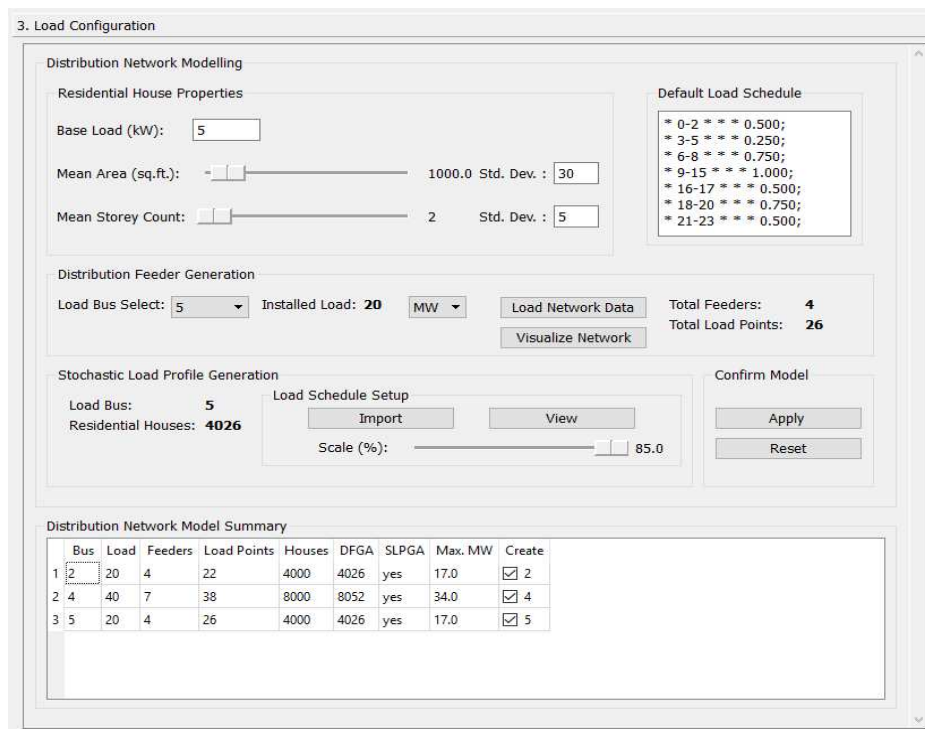


Figure 2.5: Load configuration for distribution system modeling in DEx.py

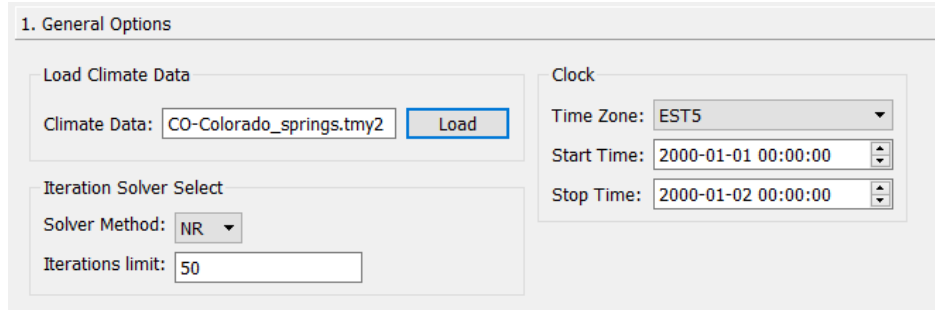


Figure 2.6: General options for distribution modeling in DEX.py

2.3 DISTRIBUTION TOPOLOGY CREATION USING DEX.py

As an example, case scenario, each load bus in RBTS is modeled by DEX.py as a distribution system in GridLAB-D[®] [13, 16]. RBTS has a peak connected load of 185MW. The end use loads connected in the distribution system aggregate to the installed capacity at each transmission bus in the RBTS. Figure 2.7 provides the operational flowchart of the data exchange tool. An overview of the major steps involved in using DEX.py is as follows:

a) DATA PROCESSING

In this stage, the balanced three phase network data is synthesized to store information about the generation, buses, transmission lines, and load points. The PowerWorld[®] Auxiliary Script File is used to extract the power system data as an input to DEX.py [17]. Alternatively, DEX.py can read case information data from power system simulators that supports the IEEE common data format [18]. As shown in Figure 2.5, DEX.py automatically populates information about the load points such as the transmission node where the load point is present and the power of the connected load in megawatts (MW) or kilowatts (kW). Load point data extracted from the power system case are used by DEX.py to model the distribution assets.

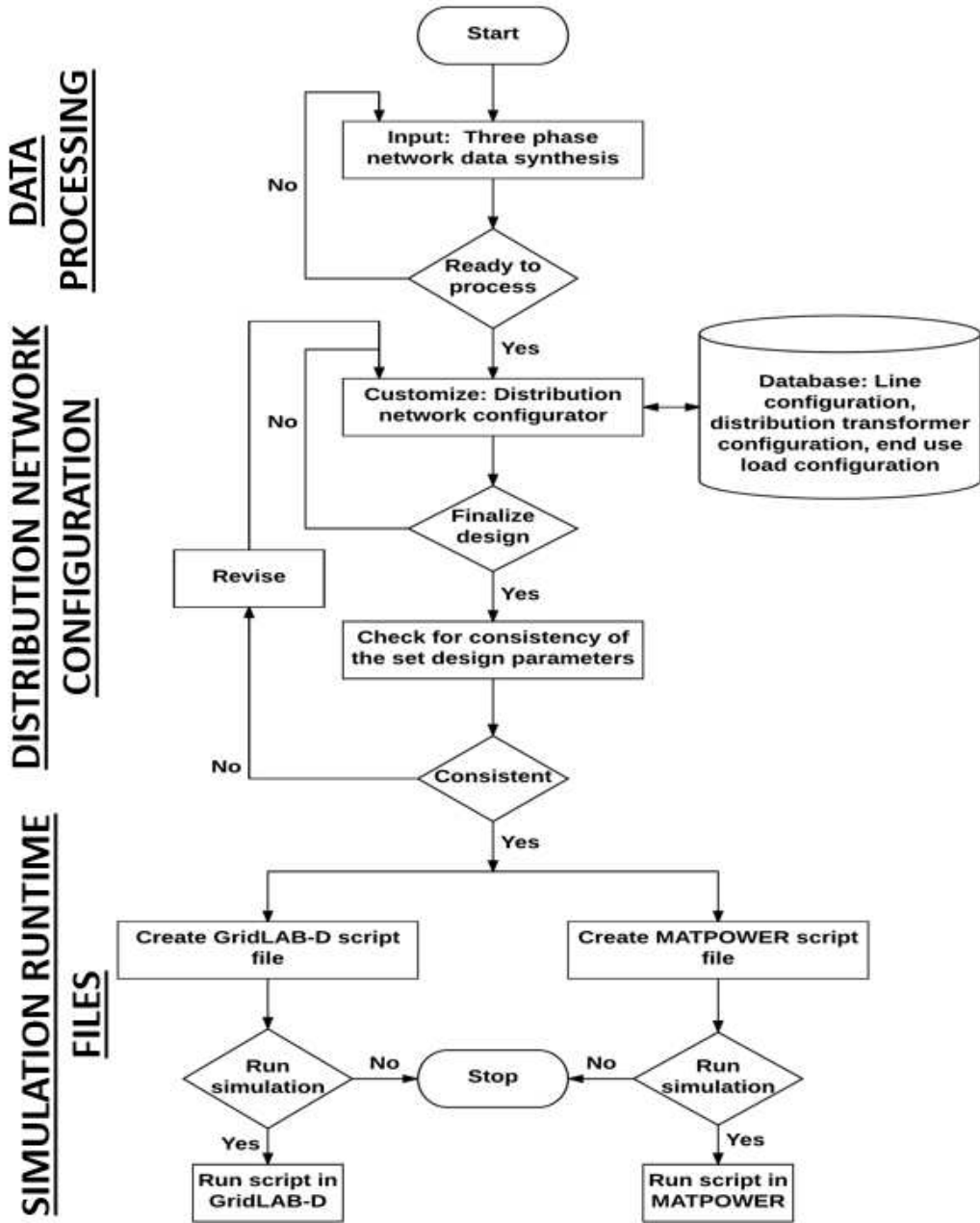


Figure 2.7: The operational flowchart for DEx.py

Figure 2.7 shows a three-bus test case system modeled in a commercial three phase simulator PowerWorld[®] and saved in the IEEE common format. DEx.py parses the three bus test

case data to filter out the generators (connected at bus 1 and 3) and load points (connected at bus 2), as shown in Figure 2.9, for further perusal.

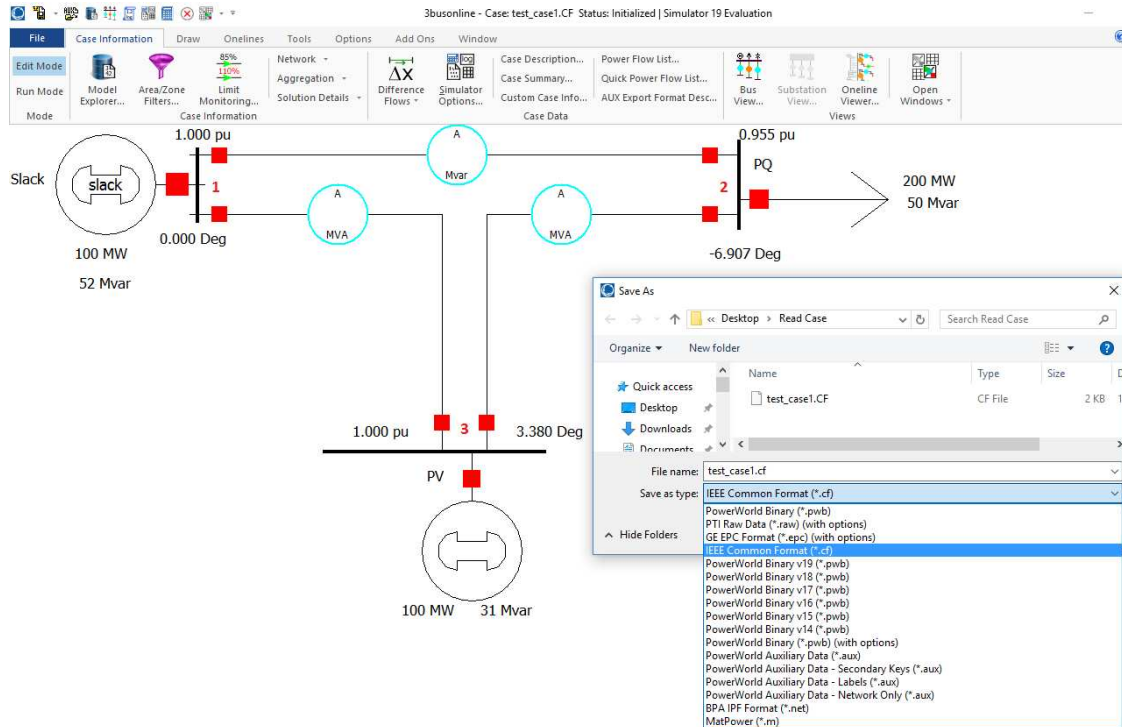


Figure 2.8: Three bus test system case saved as IEEE common format data in PowerWorld®

```
Python 3.4.4 (v3.4.4:737efcadf5a6, Dec 20 2015, 20:20:57) [MS
D64] on win32
Type "copyright", "credits" or "license()" for more informati
>>>
-----Generator Buses-----
Bus ID: 1, Name: Slack
Bus ID: 3, Name: PV
-----Load Buses-----
Bus ID: 2, Name: PQ
>>> |
```

Figure 2.9: DEx.py console output from parsing an IEEE common format file for three-bus test system


```

Python 3.4.4 Shell
File Edit Shell Debug Options Window Help
-----Generator Buses-----
Bus ID: 1, Name: GlenLyn132
Bus ID: 2, Name: Claytor1321
Bus ID: 3, Name: Kumis1321
Bus ID: 4, Name: Hancock1321
Bus ID: 5, Name: Fieldale1321
Bus ID: 6, Name: Roanoke1321
Bus ID: 7, Name: Blaine1321
Bus ID: 8, Name: Reusens1321
Bus ID: 9, Name: Roanoke1.01
Bus ID: 10, Name: Roanoke331
Bus ID: 11, Name: Roanoke111
Bus ID: 12, Name: Hancock331
Bus ID: 13, Name: Hancock111
Bus ID: 27, Name: Cloverdle331
Bus ID: 28, Name: Cloverdle13211
-----Load Buses-----
Bus ID: 2, Name: Claytor1321
Bus ID: 3, Name: Kumis1321
Bus ID: 4, Name: Hancock1321
Bus ID: 5, Name: Fieldale1321
Bus ID: 7, Name: Blaine1321
Bus ID: 8, Name: Reusens1321
Bus ID: 10, Name: Roanoke331
Bus ID: 12, Name: Hancock331
Bus ID: 14, Name: Bus1433
Bus ID: 15, Name: Bus1533
Bus ID: 16, Name: Bus1633
Bus ID: 17, Name: Bus1733
Bus ID: 18, Name: Bus1833
Bus ID: 19, Name: Bus1933
Bus ID: 20, Name: Bus2033
Bus ID: 21, Name: Bus2133
Bus ID: 23, Name: Bus2333
Bus ID: 24, Name: Bus2433
Bus ID: 26, Name: Bus2633
Bus ID: 29, Name: Bus2933
Bus ID: 30, Name: Bus3033
>>>
GUI: ON (TK)

```

Figure 2.10: DEx.py console output from parsing an IEEE common format file for 30 bus test system [21]

Figure 2.10 shows an additional example of the power system data parsed by DEx.py for an IEEE 30 bus test case made available in the public domain [21]. Figure 2.11 shows the three-phase network for the RBTS system modeled in PowerWorld[®], that serves as the candidate system for DEx.py. The details of the loads present at RBTS buses in PowerWorld[®] is shown in Table 2.1.

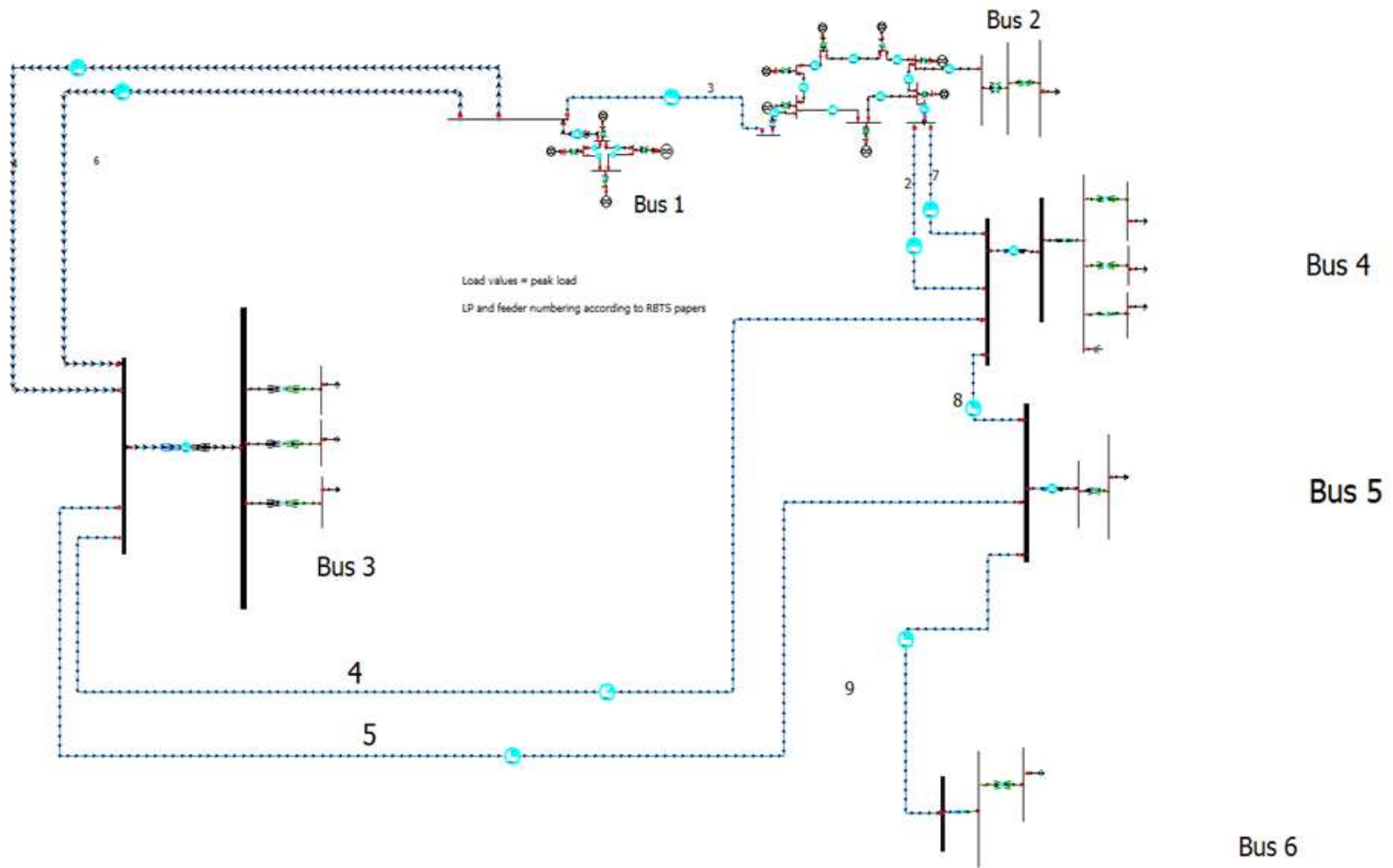


Figure 2.11: The RBTS network modeled in PowerWorld[®]

Table 2.1: The RBTS load points modeled in PowerWorld® [13]

Bus	Transmission Node	Load Points	Connected Load (MW)
2	38	22	20
3	344	44	30
	346		40
	347		15
4	106	38	10
	107		10
	108		20
5	180	26	20
6	263	31	20
Total MW			185

b) DISTRIBUTION NETWORK CONFIGURATION

The parameters associated with the selection of the distribution system assets—sizing of the cables, distribution transformers, load points are configured here. DEx.py uses the Distribution Feeder Generation Algorithm (DFGA) to create the distribution topology—real or synthetic—linked to each transmission node in a three-phase power system. Detailed implementation for DFGA is explained in Chapter 4.

For creating a distribution topology, the user can set the upper bounds on the number of feeders (F_N), its associated set of load points (LP_K), and sub-load points ($Sub - LP_j$) in a text file. Each line in the text file will be read as a feeder configuration for a distribution network at a particular transmission bus(TN_i). The general format for defining a feeder topology with its set of load points and sub-load points at a bus is as follows:

$$TN_i = \begin{bmatrix} [F_1: [LP_1: [Sub - LP_1, Sub - LP_2, \dots, Sub - LP_{j1}], LP_2, LP_3, \dots, LP_{k1}]], \\ [F_2: [LP_1: [Sub - LP_1, Sub - LP_2, \dots, Sub - LP_{j2}], LP_2, LP_3, \dots, LP_{k2}]], \\ \vdots \\ [F_n: [LP_1: [Sub - LP_1, Sub - LP_2, \dots, Sub - LP_{jn}], LP_2, LP_3, \dots, LP_{kn}]], \\ \vdots \\ [F_N: [LP_1: [Sub - LP_1, Sub - LP_2, \dots, Sub - LP_j], LP_2, LP_3, \dots, LP_K]] \end{bmatrix}$$

The above feeder configuration format represents ‘N’ number of primary feeders per transmission bus. Each feeder can have up to k_a set of load points where $a = 1, 2, 3, \dots, K$ and K is the total number of load points. Sub-load points, $Sub - LP_j$ can be placed at any load point, LP_K per feeder, F_N . To elucidate the feeder configuration format further, let us consider the first element in the feeder list:

$$[F_1: [LP_1, LP_2, LP_3, \dots, LP_{k_1}]]$$

The first element defines the first feeder, F_1 for a transmission bus, TN_i having 1 to k_1 load points which is basically the length of the connected sub-elements on that feeder. Each sub-element or load point inside the feeder object defines the number of load points attached to that node of the feeder. For example, consider the upper bounds of the load points for a single feeder configuration (F_1) at a bus saved in a text file as shown:

$$[[2, 2, 2, 1], [1, 1]]$$

The list shown above defines that the primary feeder, F_1 at a bus consists of two sub-feeders. The first sub-feeder has 4 nodes and the second sub-feeder has 2 nodes. The first sub-feeder has 2 load point connected at the first 3 nodes and 1 load point connected at the last node. The second sub-feeder has 1 load point connected at each of its 2 nodes. The total load points (LP_{total}) present in F_1 is,

$$LP_{total} = 2 + 2 + 2 + 1 + 1 + 1 = 9 \quad (2.1)$$

DEx.py reads the feeder configuration from the text file to customize the distribution system topology. Using the information presented in Table 2.1, the feeder configuration for each bus in RBTS is defined in the text file as follows:

i. Bus 2 with 4 feeders and 22 load points:

$[[2,2,2,1], [1,1], [1,2,2,1], [2,2,1,2]]$

ii. Bus 3 with 8 feeders and 44 load points:

$[[1,1,1,2,2], [1,1,1], [1,1,2,1,2], [1,1,2,1,2], [1,1,1,2,2], [2,2,1,2], [1,1,1], [1,1,1]]$

iii. Bus 4 with 7 feeders and 38 load points:

$[[1,1,1,2,2], [1,1,1], [1,1,2,1,2], [1,2,2,1,2], [1,1,1], [1,1,1], [1,1,2,1,2]]$

iv. Bus 5 with 4 feeders and 26 load points:

$[[2,2,2,1], [1,1,1,1,2], [1,2,1,2], [2,2,1,2]]$

v. Bus 6 with 4 feeders and 31 load points coupled with 8 sub-load points:

$[[1,1,1,1,1,1], [1,1,1,1,1,1], [1,1,1,1], [1,1,1,1,1,1], [[1], [1,1,1,1]], 1,1,1, [[1,1,1,1,1,1], 1,1,1]]$

A representation of the data input to DEx.py to read the feeder and load points for a transmission bus is shown in Figure 2.12. The tree diagram of the distribution topology for RBTS bus 2 consisting of feeders, sub-feeders, and its load points created using DFGA is shown in Figure 2.13. The tree diagram is generated in the DEx.py GUI and serves as a feedback to test with different user defined feeder configurations. The DFGA method built into DEx.py optimally arranges the load points to be created at each feeder for the user defined feeder configuration. It ensures that the total load points will always match the total load connected at a particular transmission bus. The constraints set into the DFGA method is further detailed in Chapter 4.

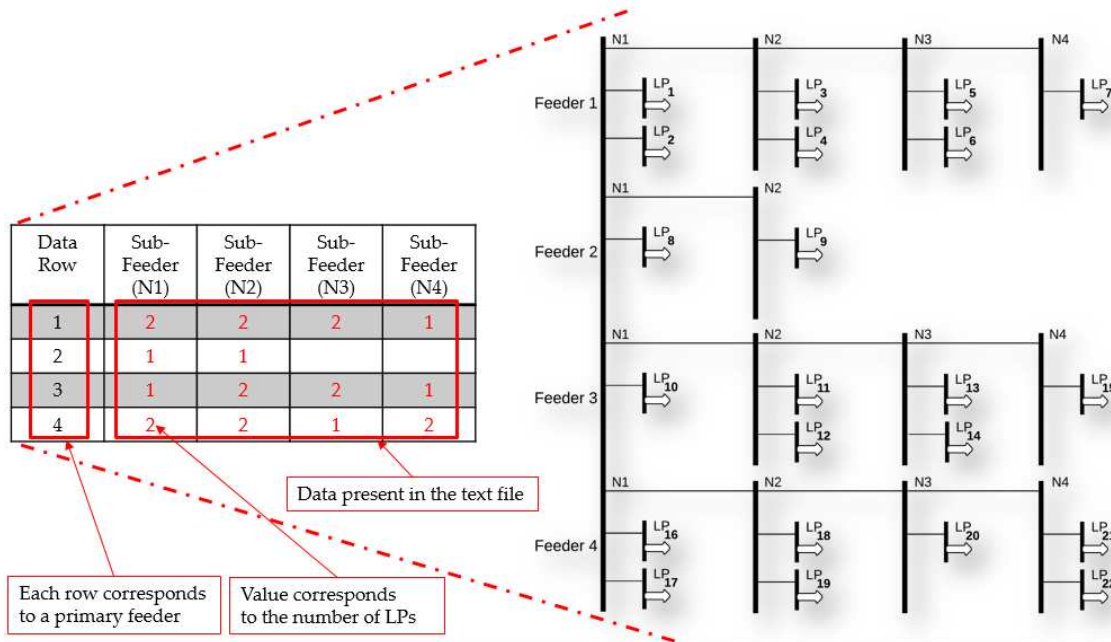


Figure 2.12: Feeder configuration format for DEX.py to parse RBTS bus 2 network model

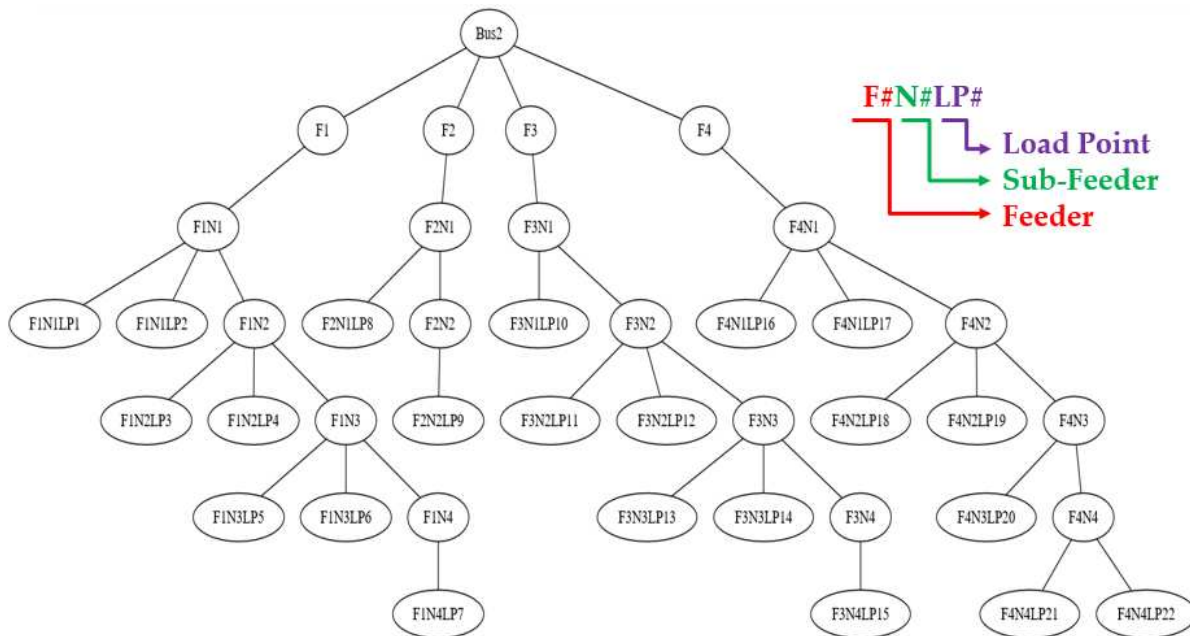


Figure 2.13: Tree diagram of the distribution topology for RBTS bus 2 created by DEX.py

After defining the feeder configuration for the distribution topology, DFGA proceeds with populating the distribution system with residential class of loads. The peak demand for each residential house object follows a schedulable ZIP load with its base power (P_{limit}) set at 5kW. ZIP load refers to the coefficients of a load model comprised of constant impedance, constant current, and constant power loads [22]. The peak power consumption for the residential load or a house object can be set in the DEx.py GUI's configuration window as shown in Figure 2.5. At present, the HVAC cooling or additional house object randomization parameters introduced by GridLAB-D[®] are kept off. Thus, the load at each RTBS bus is represented in terms of house objects connected at that bus. From Table 2.1, total connected load at RBTS bus 3 is 85MW and thus,

$$H_{ideal} = \frac{85MW}{(P_{limit}=5kW)} = 17,000. \quad (2.2)$$

So, 17000 residential house objects will be connected at RBTS bus 3 distribution system. The line configuration for the distribution system is set up in the DEx.py GUI line configuration window as shown in Figure 2.4. The details of the line parameters selected are shown in Table 2.2. The data sheet for the line configurations built into DEx.py are obtained from [23-24]. The lines represent the link between the distribution elements—buses, meters, and transformers to the residential house objects.

Table 2.2: Distribution system line configuration set up in DEx.py

Size	Material	Stranding	GMR (ft.)	Resistance (ohm/mile)	Capacity (A)
700,000	Copper	37 STRD	0.0308	0.0947	1040

Each residential house object can have a maximum current consumption of,

$$\frac{5kW}{120V} = 41.67 A \quad (2.3)$$

From the line parameter information in Table 2.2, each distribution line can hold a maximum of,

$$\left\lceil \frac{1040A}{41.67A} \right\rceil = 25 \text{ houses} \quad (2.4)$$

Considering a safe margin for the amount of load to be placed on the line, we define a line loading factor $\gamma \in (0,1)$, which can be varied. Setting $\gamma = 0.88$, a constraint is created to place a maximum of 22 residential house objects per distribution line per phase for each load point. Thus, each load point in the feeder will be constrained to have a maximum of,

$$22 \times 3 = 66 \text{ houses.} \quad (2.5)$$

Thus, house objects are equal split at each load point to create a balanced load at the feeder head. This also ensures that distribution lines are never overloaded.

With the number of house objects and the distribution topology defined at each RBTS bus, DFGA proceeds with to create the distribution system with the constraints as defined earlier to place a maximum allowable of $LP_{max} = 22$ house objects per distribution line per phase for each load point in the feeder. If the constraint is violated, DFGA will add additional lines per phase in parallel to complete the full placement of house objects across the total load points in the feeder for each RBTS bus. The logic for the placement of parallel connections used by DFGA is explained using RBTS bus 2.

From Table 2.1 and using (2.2), RBTS bus 2 has a connected load of 20MW which is the equivalent of 4000 connected houses, with each house object having a peak load of 5kW. As defined in the feeder configuration earlier, bus 2 has 4 feeders with $LP_{total} = 22$ load points. Using

this available data, we determine the number of parallel distribution lines to be added per phase across the 4 feeders to distribute $H_{ideal} = 4000$ number of houses.

The parallel connections (p) at each load point for all three phases will be,

$$p = \frac{H_{ideal}}{LP_{total} \times LP_{max} \times 3}. \quad (2.6)$$

Therefore, the parallel connections to be created at each load point of bus 2 will be,

$$p = \frac{4000}{22 \times 22 \times 3} = 2.75. \quad (2.7)$$

The number 2.75 is rounded off to the next greatest integer, 3. Thus, $p = 3$, parallel connections are placed at each phase of the distribution line at each load point. This process is then replicated for the rest of the RBTS load bus. Table 2.3 represents the setup of the feeders, load points, and the parallel configuration created for each RBTS load bus.

Table 2.3: Setup of feeders, load points, and parallel configurations for each RBTS bus

RBTS Bus	Connected Load (MW)	Ideal House Objects (H_{ideal})	Total Feeders	Total Load Points (LP_{total})	Parallel connections per load point (p)
2	20	4000	4	22	3
3	85	17000	8	44	6
4	40	8000	7	38	4
5	20	4000	4	26	3
6	20	4000	4	39	2

Placing p parallel connections per load point to distribute H_{ideal} house objects at each feeder will lead to round-offs: the total house objects distributed across the total load points will be much greater than H_{ideal} .

This problem is illustrated by extending the parallel connection logic using (2.6) across all load buses in the RBTS network in Table 2.4.

Table 2.4: Deviation from H_{ideal} created by p parallel connection round-offs at each load point

RBTS Bus	Connected Load (MW)	Ideal House Objects (H_{ideal})	Actual House Objects (H_{actual})	% Increase in House Objects
2	20	4000	$3 \times 22 \times 66 = 4356$	8.9
3	85	17000	$6 \times 44 \times 66 = 17424$	2.5
4	40	8000	$4 \times 38 \times 66 = 10032$	25.4
5	20	4000	$3 \times 26 \times 66 = 5148$	28.7
6	20	4000	$2 \times 39 \times 66 = 5148$	28.7

With the parallel configuration approach defined using (2.6), the exact value of the parallel lines to be connected at each phase of the load point at RBTS bus 2 comes to 2.75. Rounding off this to the next nearest integer value, 3 gives rise to the round-off error that gets replicated throughout the total load points present at a given load bus. As indicated in Table 2.4, the percentage deviation for H_{actual} house objects placed at the load bus from H_{ideal} varies between 8.9% to 28.7% for the RBTS network model. This implies that at RBTS bus 6, with a connected load of 20MW, 5148 house objects will have a peak demand of,

$$5148 \times 5kW = 25.74MW \quad (2.8)$$

The effect of these round-offs become more dominant as the total house objects and load points varies with each bus configuration.

To minimize the effect due to round-offs, DFGA optimally refines the number of p parallel connections to be placed at each load point. This is achieved by initially placing an equal number of parallel connections at a certain number of load points for a given load bus. DFGA then re-

distributes the parallel connections among the remaining load points depending on the number of parallel connections left to be created. The objective here is to place the parallel connections in a manner that constrains the maximum deviation of H_{ideal} at a given load bus to be less than or equal to LP_{max} . A generalized equation formed to meet this constraint is as shown:

$$H_{actual}^{new} = [((LP_{total} - 1) \times p) + p'] \times LP_{max} \quad (2.9)$$

Where,

- i. H_{actual}^{new} is the maximum number of house objects that can be placed at a given load bus
- ii. p' represent the load point count for p parallel connections to be placed. The value of p should be less than p' else (2.9) is computed again by incrementing p

The proposed approach is explained for distributing $H_{ideal} = 17000$ house objects at RBTS bus 3 having $LP_{total} = 44$ load points using the following steps:

- i. Total house objects to be placed at each load point:

$$\left\lfloor \frac{17000}{44} \right\rfloor = 383 \quad (2.10)$$

- ii. Parallel connections to be placed at each load point:

$$\left\lfloor \frac{383}{66} \right\rfloor = 5 \quad (2.11)$$

- iii. 5 parallel connections will be placed till the 43rd load point at RBTS bus 3. The 44th load point will have the following parallel connection, p' :

$$p' = \left\lceil \frac{H_{ideal} - (LP_{total} - 1) \times p \times LP_{max}}{LP_{max}} \right\rceil \quad (2.12)$$

$$= \left\lceil \frac{17000 - (44 - 1) \times 5 \times 66}{66} \right\rceil \quad (2.13)$$

$$p' = 43 \quad (2.14)$$

- iv. Using (2.9), the maximum house objects that will be placed at RBTS bus 3 will be:

$$H_{actual}^{new} = [(44 - 1) \times 5] + 43 \times 66 \quad (2.15)$$

$$= 17028 \text{ house objects} \quad (2.16)$$

Using the above approach, $p = 5$ parallel connections will be created till the 43rd load point and $p' = 43$ parallel connections at the 44th load point to place $H_{actual}^{new} = 17028$ house objects at RBTS bus 3. The parallel connections at each load point are further re-arranged using (2.9) to limit their large numbers at any one load point. The re-arrangement is performed using (2.9) as shown:

$$17028 = [p \times x + LP_{total} - x] \times 66 \quad (2.17)$$

Using (2.17), the equation is solved with $p = 5$ to get, $x = 53$ which is greater than the total number of load points present at RBTS bus 3. Equation (2.17) is solved again by incrementing p to 6 to get $x = 42$, which is less than $LP_{total} = 44$, present at RBTS bus 3. Thus, the final arrangement of parallel connections come to 6 parallel connection placed till the 42nd load point at RBTS bus 3. At the remaining two load points, the distribution of the parallel connection will take place as follows:

Total parallel connection at RBTS bus 3:

$$(5 \times 43) + 43 = 258 \quad (2.18)$$

Total parallel connection till the 42nd load point:

$$6 \times 42 = 252 \quad (2.19)$$

Total parallel connections to be distributed among the remaining two load points (43 and 44):

$$258 - 252 = 6 \quad (2.20)$$

6 parallel connections can be distributed among the remaining two load points by placing 3 parallel connections per load point. This parallel connection re-arrangement approach has been extended to all the remaining RBTS load buses to create H_{actual}^{new} house objects for each bus. The result of this re-arrangement is detailed in Table 2.5. The change in the house objects created for each RBTS bus before and after the application of DFGA is shown in Figure 2.14.

Table 2.5: Final house objects (H_{actual}^{new}) placed at each RBTS bus by re-arranging the number of parallel connections at each load point

RBTS Bus	Connected Load (MW)	Ideal House Objects (H_{ideal})	Actual House Objects (H_{actual})	Final House Objects (H_{actual}^{new})	% Increase in House Objects
2	20	4000	4356	4026	0.65
3	85	17000	17424	17028	0.164
4	40	8000	10032	80521	0.65
5	20	4000	5148	4026	0.65
6	20	4000	5148	4026	0.65

c) SIMULATION RUNTIME FILES

In this stage, DEx.py creates the configuration script files for the RBTS system at the transmission level in MATPOWER[®] and the distribution topology for each RBTS bus in GridLAB-D[®]. Figure 2.15 shows the successful execution of the distribution topology for each

RBTS bus from the GridLAB-D’s log output. A synthetic system load profile is used for indicating the peak demand of the connected house objects for each RBTS bus.

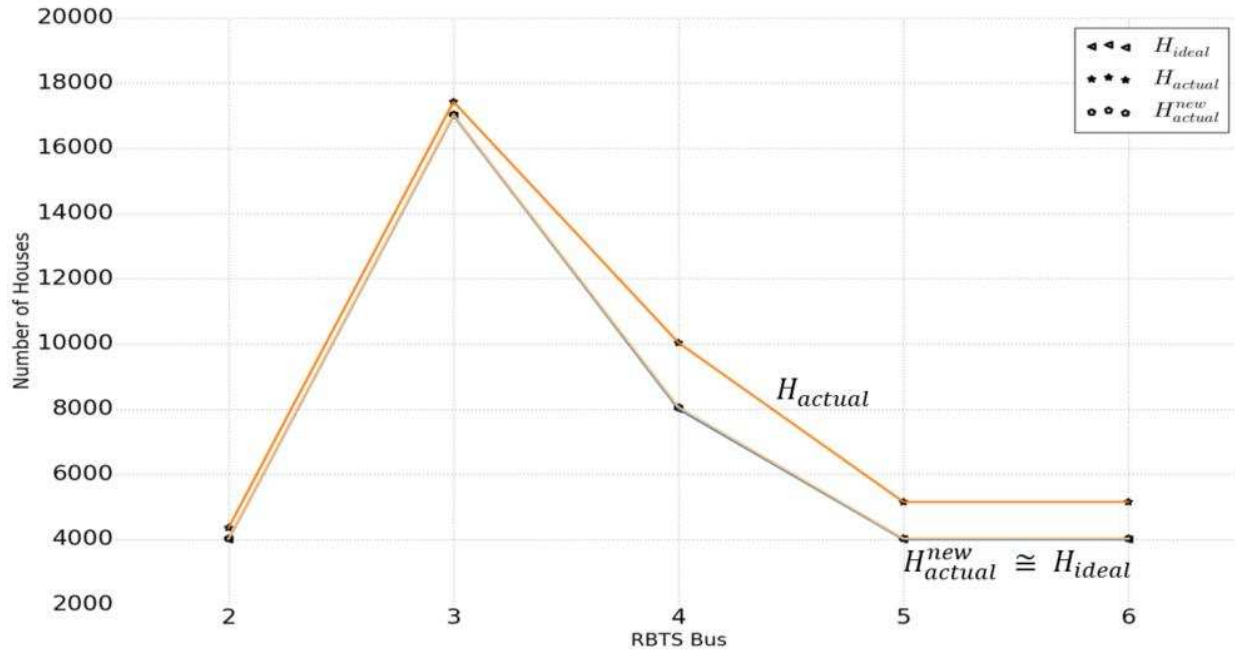


Figure 2.14: Before and after count of house objects or residential loads placed at each RBTS load bus using DFGA

2.4 CONCLUDING REMARKS

The proposed data exchange tool employs abstract mathematical processes to reliably create the transmission and distribution assets for power system simulation studies. The application of the tool for creating a user defined distribution topology populated with 17028 residential house objects linked to bus 3 in RBTS is shown. The tool acts as an interface for setting up the co-simulation variables, in this case, modeling the transmission and distribution assets for running in transmission and distribution simulators. The two key algorithms that are built into DEx.py for simplifying the modeling of distribution assets—the distribution feeder generation algorithm

(DFGA) and stochastic load profile generation algorithm (SLPGA) are elucidated in the following chapter. Further, details on the implementation of the synthetic system load profile is also explained in Chapter 4.

```

C:\WINDOWS\system32\cmd.exe
C:\Users\xtreme\Desktop\GridLAB equal Distribution>gridlabd case_Man
WARNING [INIT] : randunit() introducing extra randomness to prevent state stagn
ation and infinite loops
Processing 2000-01-02 00:00:00 MST...
Core profiler results
=====
Total objects          78967 objects
Parallelism            1 thread
Total time             19.0 seconds
  Core time            5.2 seconds (27.5%)
  Compiler             2.4 seconds (12.6%)
  Instances            0.0 seconds (0.0%)
  Random variables     0.0 seconds (0.0%)
  Schedules            0.0 seconds (0.0%)
  Loadshapes           0.4 seconds (2.2%)
  Enduses              0.5 seconds (2.9%)
  Transforms           0.3 seconds (1.5%)
  Model time          13.8 seconds/thread (72.5%)
Simulation time        1 days
Simulation speed       100k object.hours/second
Passes completed       54 passes
Time steps completed   27 timesteps
Convergence efficiency 2.00 passes/timestep
Read lock contention   0.0%
Write lock contention  0.0%
Average timestep      3200 seconds/timestep
Simulation rate        4547 x realtime

Model profiler results
=====
Class                  Time (s)  Time (%)  msec/obj
-----
collector              6.742    49.0%    674.2
house                  5.080    36.9%    0.1
ZIPload                1.362    9.9%     0.0
triplex_meter          0.299    2.2%     0.2
triplex_line           0.177    1.3%     0.1
transformer            0.045    0.3%     0.1
triplex_node           0.032    0.2%     0.1
node                   0.016    0.1%     0.1
overhead_line          0.015    0.1%     0.1
=====
Total                  13.768   100.0%   0.2
=====
C:\Users\xtreme\Desktop\GridLAB equal Distribution>

```

Figure 2.15: Simulation output of the distribution topologies created for each RBTS bus in GridLAB-D[®]

CHAPTER 3

AN ALGORITHMIC APPROACH FOR CREATING DIVERSE STOCHASTIC FEEDER DATASETS FOR POWER SYSTEM CO-SIMULATIONS²

This chapter describes the methodologies adopted to create distribution network assets linked from a transmission system and to populate the distribution network with stochastic end-use loads. Algorithms to model distribution system assets and its application using a standard test system are presented.

3.1 INTRODUCTION

The traditional method of performing three-phase power system studies assumes that the connected load is static and balanced in each of the three phases. The simulations studies pertaining to stability, reactive power flow, load flow, system planning, and contingency analysis are performed with this underlying assumption, and do not consider the time-varying loads. Title XIII of the Energy Independence and Security Act (EISA), passed by the United States Congress in December 2007, set the framework for coordinating national grid modernization efforts towards the Smart Grid Initiative [25]. From a Smart Grid perspective, integration of distributed generation, hybrid loads, home energy management systems, and real-time pricing models pose a significant risk in compromising power grid dynamics. Extensive simulation studies are needed to determine the performance of these advanced Smart Grid technologies to ensure their safe operation in the envisioned modern grid.

²This chapter is a verbatim reproduction of the work accepted and published to a peer reviewed conference proceedings as cited in the reference [12]. The required permissions for re-use of the material have been obtained from the copyright holders and are included in the Appendix B. The numbering of the figures and tables has been modified to satisfy the formatting requirements of the thesis.

Agent-based power system modeling tools, such as GridLAB-D[®], [16], provide a detailed modeling environment at the end-use level for performing distribution simulation studies over disparate timescales that are not possible with commercial or open source three-phase balanced network solvers [14, 15]. Performing distribution simulation requires detailed information about the connected loads at the end-use level. Using this information, we can aggregate the individual distribution level loads to perform load scheduling, electric vehicle integration, and other Smart Grid technology integration studies.

To the knowledge of the authors, no direct tool exists for creating detailed distribution models that include stochasticity over a wide time range. In [26], the GridMat MATLAB tool box developed for GridLAB-D provides a basic interface for configuring the end-use residential models available within GridLAB-D. In [27], the stochastic demand profile for residential loads is implemented using a bottoms-up approach by performing power measurement studies on different types of appliances. In contrast, our algorithm, developed on an open source platform, can create complex distribution models populated with residential loads having stochastic demand profiles to simulate a large distribution system linked to a transmission, sub-transmission, or substation node.

Our algorithm simplifies the topology creation process for GridLAB-D users. Obtaining valid and consistent load profile data of low voltage residential consumers is difficult due to privacy concerns. To overcome the problem of modeling the residential loads, two popular approaches exist: namely, top-down and bottom-up. We deploy the bottom-up approach to generate a physical layer of the distribution system to match the installed capacity at a transmission node. We employ the top-down approach to generate low voltage residential load profiles using one-dimensional random walk process that matches the substation load curves [28]. Any transmission node consisting of a static load in a power system model, created in a three-phase

power flow software, can be expanded to incorporate our distribution model for performing co-simulation studies using transmission and distribution simulators. In this paper, we make the following contributions:

1. Creation of diverse feeder datasets for simulating large number of residential loads in a distribution network, and
2. Design of stochastic demand patterns for residential loads.

Section II in this paper describes the abstract skeleton model that is used for the creation of distribution elements that can be linked with a transmission node for performing co-simulation studies. In Section III, the use of random walk theory for the generation of the stochastic demand patterns for residential loads is presented. The setup and results of the simulation study using a candidate test system is described in Section IV. Section V concludes.

3.2 A MATHEMATICAL MODEL FOR CREATING THE DISTRIBUTION SYSTEM TOPOLOGY

a) TOPOLOGY OF THE DISTRIBUTION NETWORK

Figure 3.1 depicts the distribution topology that is linked from a transmission node i present in a three-phase power system model. It is assumed that the transmission node has a static load, represented by an installed capacity L , connected to it. Each transmission node can be expanded into a hierarchal distribution network of up to 4 levels. For each transmission node i , the distribution hierarchy consists of $1, \dots, n$ feeders or primary feeders (Level 1). Each primary feeder will have at least one sub-feeder (Level 2).

Each primary feeder and sub-feeder consists of at least one load point LP (Level 3). Each load point may consist of additional sub-load points (Level 4). The upper bounds for the number

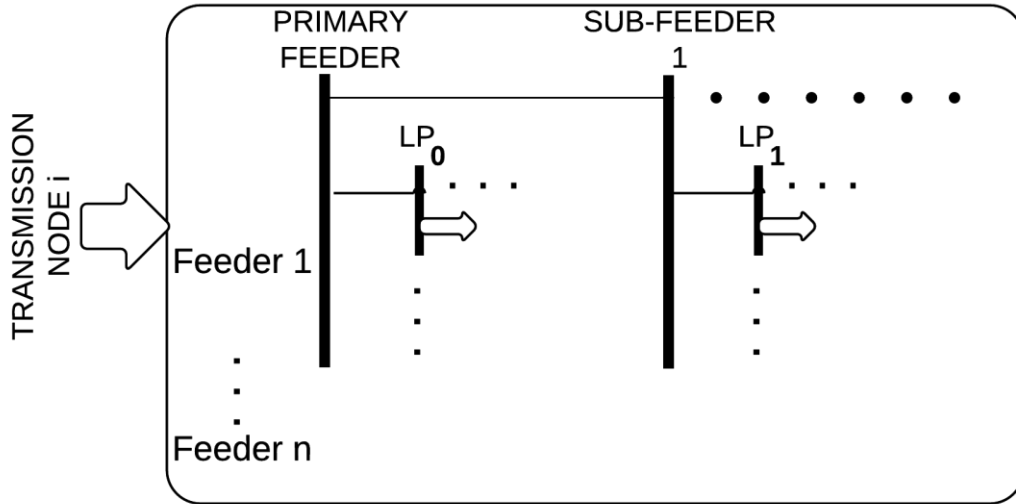


Figure 3.1: The distribution topology linked from a transmission node i

of primary feeders, sub-feeders per primary feeder, load points per feeder, and additional sub-load points per load point per feeder for each transmission node set by the user to create a new distribution topology. Alternatively, the user can re-create an existing distribution network topology for this purpose. LP_{total} represents the sum of the total load points and sub-load points in the distribution network for each transmission node i . The link between the primary feeders and sub-feeders can be configured as an overhead line or an underground cable. The load points and sub-load points in the distribution topology represent the connection points for the low voltage (LV) network where the residential loads are connected as shown in Figure 3.2. The LV network consists of step down distribution transformers, triplex nodes, and triplex lines for each of the three phases. The residential loads are connected at the end points of each phase of the LV network at distribution voltage level. The residential loads to be distributed across the distribution topology made up of LP_{total} is determined by L at a transmission node i .

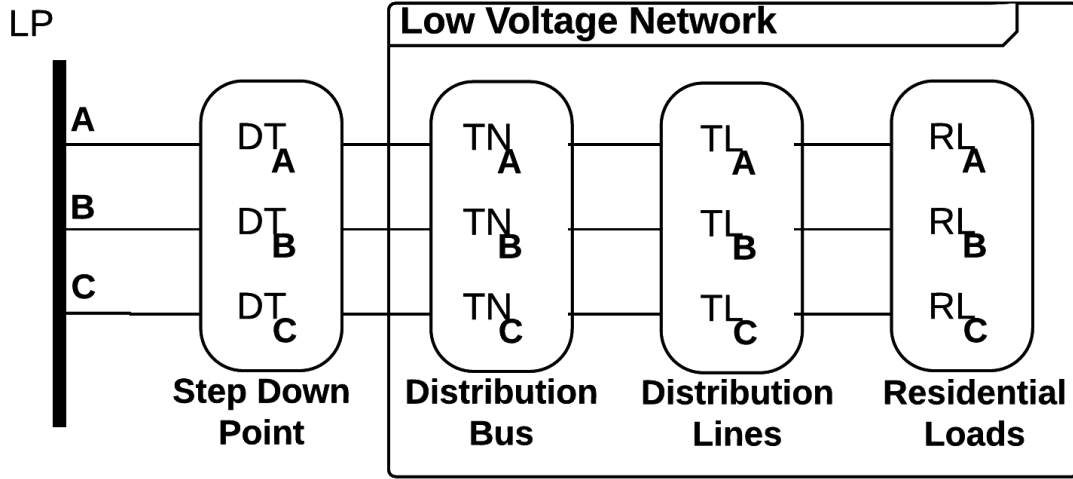


Figure 3.2: The low voltage network topology for each load point (LP)

b) PARAMETERS FOR THE DISTRIBUTION TOPOLOGY

For the correct operation of the distribution topology, the LV network at each LP must adequately supply its connected residential loads (RL_A, RL_B, RL_C) without overloading any of the distribution elements at each phase. At the lowest level, the distribution lines (TL_A, TL_B, TL_C) represent the weakest link, each having a maximum current carrying capacity I_t . Each LP out of LP_{total} is therefore constrained by design to only connect the maximum residential loads that can be safely powered by the connected three phase distribution lines and is given by,

$$LP_{max} = \left\lceil \frac{3\gamma V I_t}{P_{limit}} \right\rceil, \quad (3.1)$$

where V is the distribution voltage level, γ represents the loading factor for each distribution line and $\gamma \in (0,1)$, and P_{limit} is the peak power consumption limit for each residential load. The electrical parameters associated with the distribution elements in the LV network shown in Figure 3.2 can be configured using the IEEE test distribution feeder data [23, 29].

c) DISTRIBUTION FEEDER GENERATION ALGORITHM

The Distribution Feeder Generation Algorithm (DFGA) translates the installed capacity L at a transmission node to residential loads distributed across LP_{total} in a user defined distribution topology. H_{ideal} represents the total residential loads from L given by,

$$H_{ideal} = \left(\frac{L}{P_{limit}} \right), \quad (3.2)$$

and LP_{actual} represents the H_{ideal} residential loads to be connected at each LP given by,

$$LP_{actual} = \left\lfloor \frac{H_{ideal}}{LP_{total}} \right\rfloor. \quad (3.3)$$

If $LP_{actual} > LP_{max}$, then the number of parallel distribution lines have the same rating I_t created in each phase in LV connected at each LP is given by,

$$p_o = \left\lceil \frac{LP_{actual}}{LP_{max}} \right\rceil. \quad (3.4)$$

Introducing the parallel connections creates rounding-off errors at each LP and causes H_{ideal} to deviated by a value,

$$H_{ideal} = LP_{max} \times LP_{total} \times p_o, \quad (3.5)$$

which would translate to a value greater than L . The deviation between H_{ideal} and H_{actual} is represented by,

$$0 \leq \delta \leq LP_{max} \times LP_{total}. \quad (3.6)$$

The DFGA tries to minimize the deviation δ to $0 \leq \delta \leq LP_{max}$ using parallel connections by finding H_{actual}^{new} such that,

$$(H_{actual}^{new} - H_{ideal}) = \delta'. \quad (3.7)$$

The variables instantiated by DFGA for the δ minimization process are:

- i. p , which represents the parallel distribution lines created at each LV phase up to $(LP_{total} - 1)$ and is given by,

$$p = \left\lfloor \frac{LP_{actual}}{LP_{max}} \right\rfloor, \quad (3.8)$$

and

- ii. p' , which represents the parallel distribution lines created at each LV phase up to the last LP given by,

$$p' = \left\lfloor \frac{H_{ideal} - (LP_{total} - 1) \times p \times LP_{max}}{LP_{max}} \right\rfloor. \quad (3.9)$$

Then, H_{actual}^{new} represents the total residential loads corresponding to LP_{total} load points to yield δ' given by,

$$H_{actual}^{new} = [(LP_{total} - 1) \times p + p'] \times LP_{max}. \quad (3.10)$$

The value of p may tend to be unrealistically high; we implement a further optimization, as explained in Algorithm 3.1, to maintain H_{actual}^{new} by re-distributing p and p' .

3.3 STOCHASTIC LOAD PROFILE GENERATION FOR RESIDENTIAL LOADS

a) OVERVIEW

The distribution topology creation in Section 3.2 describes the physical network model based on Figure 3.1 using an abstract mathematical algorithm subject to user-defined constraints for each of its constituent distribution elements. For the topology thus created using DFGA, we devise a methodology that will create time-varying stochastic load profiles over a defined time

period for H_{actual}^{new} residential loads distributed across LP_{total} load points from an installed capacity L present at a transmission node i .

Algorithm 3.1 Algorithm to re-distribute $(p + p')$ parallel connections across LP_{total} load points maintaining H_{actual}^{new}

```

1:  $L[1, \dots, LP_{total}]$  is row vector such that  $L[i]$  denotes the
   parallel connections needed for load point  $i$ 
2: Initialize  $L[1, \dots, LP_{total}] = \{0\}$ 
3:  $\emptyset = [H_{actual}^{new} - (LP_{max} \times LP_{total})] / (p - 1)LP_{max}$ 
4: while  $\emptyset > LP_{max}$  do
5:      $p = p + 1$ 
6:     recompute  $\emptyset$ 
7: end while
8:  $q = (H_{actual}^{new} / LP_{max}) - \emptyset \times p$ 
9: Assign  $L[1, \dots, LP_{total}] = \{p\}$ 
10:  $k = \emptyset + 1$ 
11: while  $q > 0$  do
12:      $L[k] = L[k] + 1$ 
13:      $q = q - 1$ 
14:      $k = k + 1$ 
15:     if  $k > LP_{total}$  then
16:          $k = \emptyset + 1$ 
17:     end if
18: end while

```

b) STOCHASTIC LOAD PROFILE GENERATION FOR H_{actual}^{new} RESIDENTIAL LOADS USING THE RANDOM WALK PROCESS

Residential loads on a distribution network seldom have the same load profile. Generating load profiles for H_{actual}^{new} residential loads on a distribution network requires a stochastic approach, while still aggregating to a realistic system load. One-dimensional random walk theory is used to

generate the load profile for the residential loads that are distributed across a distribution feeder network.

To generate the time-varying load profile that can accurately depict a daily, monthly, or yearly variation in energy usage for the H_{actual}^{new} residential loads, a top-down load distribution methodology is employed. Here the load profile for each residential load is scaled from a time-varying system load at the transmission node i , $sys(t)$, and used as the input for the one-dimensional random walk process as defined by (3.11).

$$y(t) = y(t - 1) + \epsilon(t), \quad (3.11)$$

where $\epsilon(t)$ is the random variable distributed as $N(\mu, \sigma)$ representing a normal distribution with a mean μ and a standard deviation σ , $y(t)$ is the current position of the random walk curve, t is the time step, and the initial position is defined to be zero. The value of $y(t)$ varies randomly at each time instant t based on the sampled value of $\epsilon(t)$. Equation (3.11) is used to create the load distribution vector $\beta_i(t)$ that generates load profiles for H_{actual}^{new} residential loads which will sum to $sys(t)$ and also ensure that none of the H_{actual}^{new} residential loads exceeds the set P_{limit} for all times.

c) STOCHASTIC LOAD PROFILE GENERATION ALGORITHM

The Stochastic Load Profile Generation Algorithm (SLPGA) generates the time-varying stochastic load profiles for H_{actual}^{new} residential loads. The profiles are scaled from an existing or custom-defined $sys(t)$ by $\beta_i(t)$ over a defined time horizon T , say 24 hours. The random variable $\epsilon(t)$ is used to generate the H_{actual}^{new} random time series over T distributed as $N(0,1)$ and inputted to the one-dimensional random walk process shown in (3.11) to generate the random walk curves for H_{actual}^{new} residential consumers. The time-varying random walk curves generated using (3.11)

normalized in the interval (0,1) is represented by $\beta_i^{pre}(t)$, known as the pre-load distribution factor such that the $y(t) = \sum_{i=1}^{H_{actual}^{new}} \beta_i^{pre} = 1$. Because of the random walk process, if we directly scale the values stored in $\beta_i^{pre}(t)$ with $sys(t)$, the resulting residential load profiles may exceed the set peak limit of the residential load, P_{limit} , at any time t . To ensure that the constraint set by P_{limit} is always met at any t , we re-distribute the $\beta_i^{pre}(t)$ vector to create the load distribution vector β_i as explained in Algorithm 3.2.

3.4 SIMULATION STUDY

a) OVERVIEW

To demonstrate the creation of stochastic demand profiles using SLPGA for H_{actual}^{new} residential loads distributed across a defined distribution topology consisting of LP_{total} load points created using DFGA, we select a single transmission node consisting of an installed capacity L from the candidate system.

b) THE CANDIDATE SYSTEM

The candidate system used in this simulation study is the Roy Billinton Test System (RBTS) that is shown in Figure 3.3. RBTS consists of generation, transmission, and distribution subsystems, and has a peak load of 185MW [13]. Each load point in RBTS has a fixed number of residential, industrial, and commercial consumers. However, for the purpose of this study, we assume that all loads are residential, thus accounting for approximately 37,000 consumers, each having a maximum rated P_{limit} of 5kW.

c) SIMULATION PARAMETERS

In this paper, bus 2 of the RBTS system is selected as the transmission node for which the detailed distribution topology is created. Bus 2 has an installed capacity (L) of 20MW; this

Algorithm 3.2 Algorithm to re-distribute $\beta_i^{pre}(t)$ to $\beta_i(t)$ subject to the $\beta^{chk}(t)$ constraint at each time instant t

$\beta_i^{pre}(t)$ is an array sized $T \times N$ consisting of $[\beta_1, \beta_2, \dots, \beta_N]$

- 1: for each t where $N = H_{actual}^{new}$, $T =$ total time period of $sys(t)$, and $t \in [0, T)$
- 2: $\beta^{chk}(t)$ is a row vector consisting of $[\beta_1^{chk}, \beta_2^{chk}, \dots, \beta_T^{chk}]$ where $\beta_i^{chk} = P_{limit}/sys(i)$ for each t
- 3: θ is a user defined tolerance or threshold set to 1e-6 in our simulation study
- 4: **for each** t in β_i^{pre} **do**
- 5: **if** $max(\beta_i^{pre}(t)) > \beta^{chk}(t)$ **then**
- 6: $\beta_{sum} = sum([\beta_i - \beta_{chk}])$ only if $\beta_i > \beta^{chk}(t)$
- 7: $\beta_{count} = count(\beta_i)$ only if $\beta_i < \beta^{chk}(t)$
- 8: $\beta_{add} = \beta_{sum}/\beta_{count}$
- 9: Replace β_i with $\beta^{chk}(t)$ if $\beta_i > \beta^{chk}(t)$
- 10: Initialize $\beta'_{count} = \beta_{count}$
- 11: **while** $\beta'_{count} > 0$ and $\beta_{sum} \geq \theta$ **do**
- 12: $\beta_{add} = \beta_{sum}/\beta_{count}$
- 13: **for each** j in $\beta_i^{pre}[t]$ **do**
- 14: **if** $\beta_i^{pre}[t][j] < \beta^{chk}(t)$ **then**
- 15: **if** $\beta_i^{pre}[t][j] + \beta_{add} > \beta^{chk}(t)$ **then**
- 16: $\beta_{sum} = \beta_{sum} - \beta^{chk}(t) + \beta_i^{pre}[t][j]$
- 17: $\beta_i^{pre}[t][j] = \beta^{chk}(t)$
- 18: $\beta'_{count} = \beta_{count} - 1$
- 19: **else**
- 20: $\beta_i^{pre}[t][j] = \beta_i^{pre}[t][j] + \beta_{add}$
- 21: $\beta_{sum} = \beta_{sum} - \beta_{add}$
- 22: **end if**
- 23: **end if**
- 24: **end for**
- 25: **end while**
- 26: **end if**
- 27: **end for**

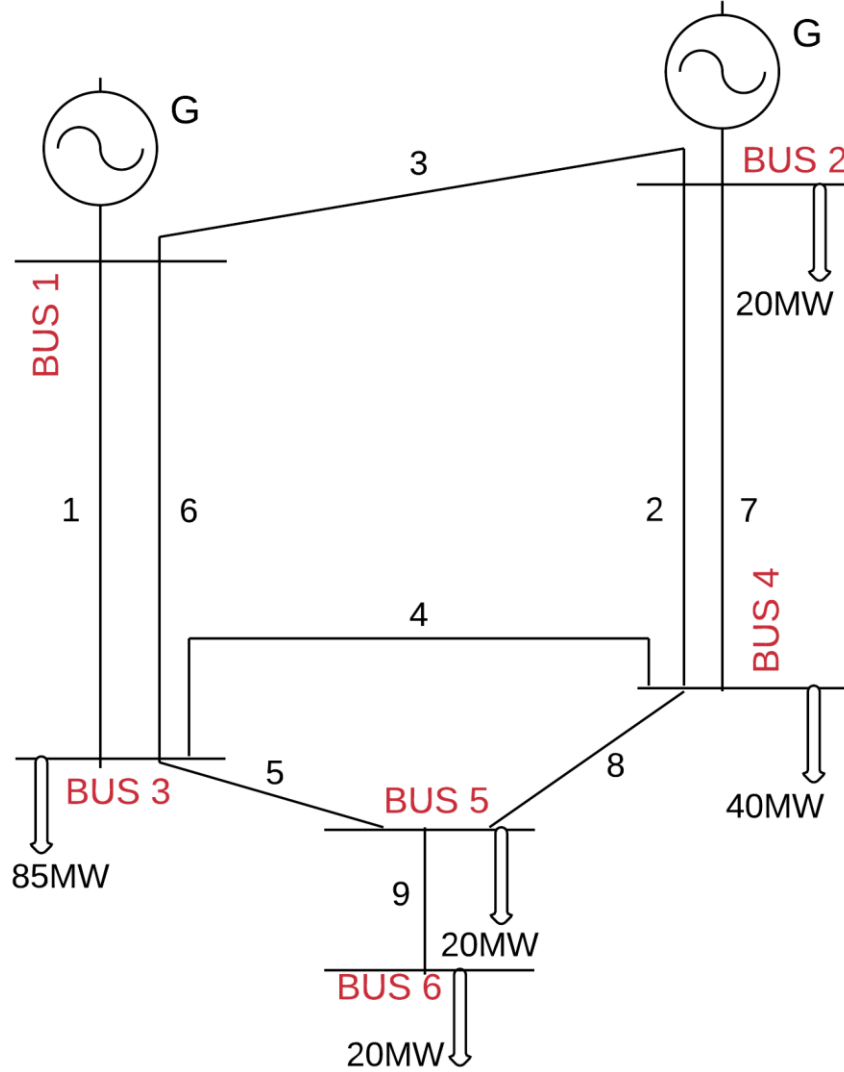


Figure 3.3: Single line diagram of the RBTS candidate system from [13]

translates to $H_{ideal} = 4000$, as each connected load has $P_{limit} = 5kW$. As explained in Section 3.2, the total number of residential loads created by DFGA at bus 2 will be $H_{actual}^{new} = 4026$. The same setup can be extended to any of the remaining transmission nodes from the RBTS candidate system or any other transmission systems. The details of the distribution topology for bus 2 of the RBTS system is shown in Table 3.1. For the distribution topology consisting of a 4-level hierarchy

created by the DFGA, the stochastic load profiles for 4026 residential loads are generated using the SLPGA.

Table 3.1: RBTS bus 2 distribution network topology from [13]

Node	Installed Capacity	Primary Feeders	Load Points
2	20MW	4	22

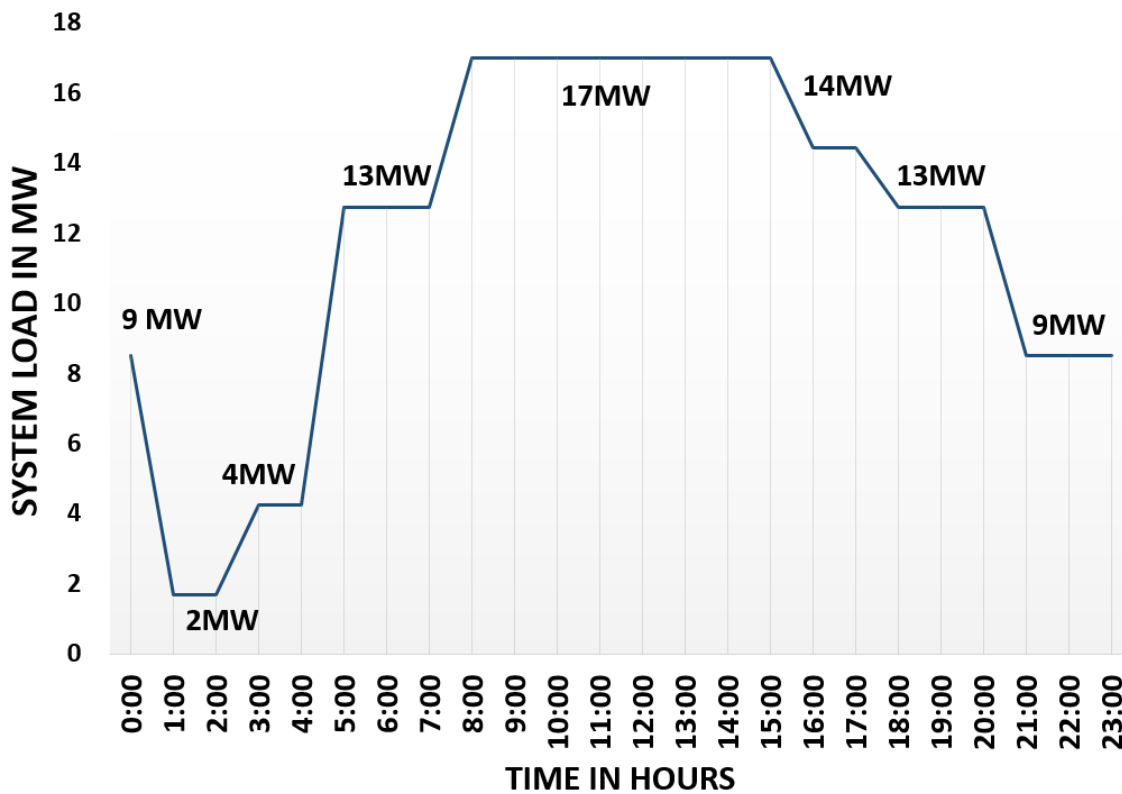


Figure 3.4: System load, $sys(t)$, for bus 2 of RBTS with peak at 17MW

d) SYSTEM LOAD PROFILE

As described in Section 3.3, the one-dimensional random walk process is used to generate the residential load profile for 4026 residences, scaled from a custom-created system load at RBTS

bus 2, $sys(t)$, which is shown in Figure 3.4. Bus 2 has a peak demand set at 17MW between 9:00 to 15:00 for a $T = 24$ -hour period. The step size, t , is set as one hour.

e) RESULTS

The results of the DFGA and SLPGA for generating the stochastic load profiles of 5 randomly selected residential loads from 4026 loads at RBTS bus 2 are given below.

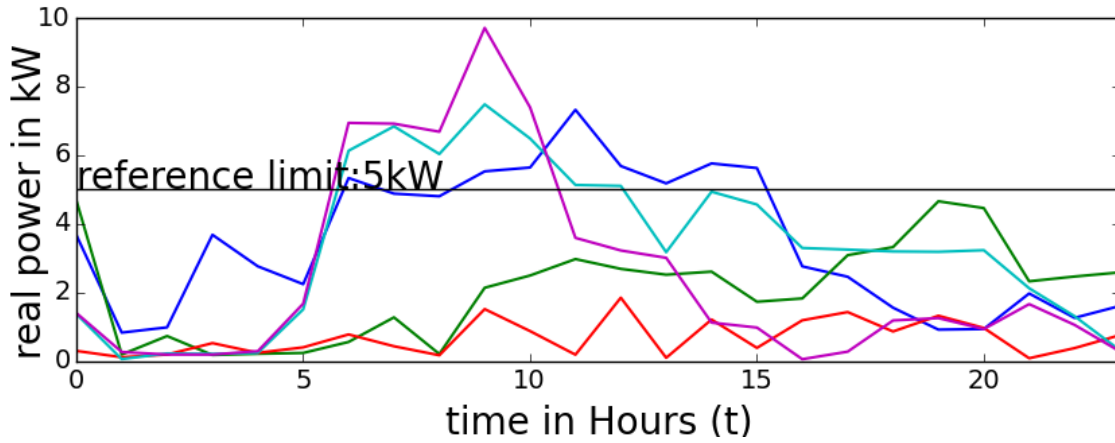


Figure 3.5: $\beta_i^{pre}(t)$ for 5 randomly selected residential loads out of 4026

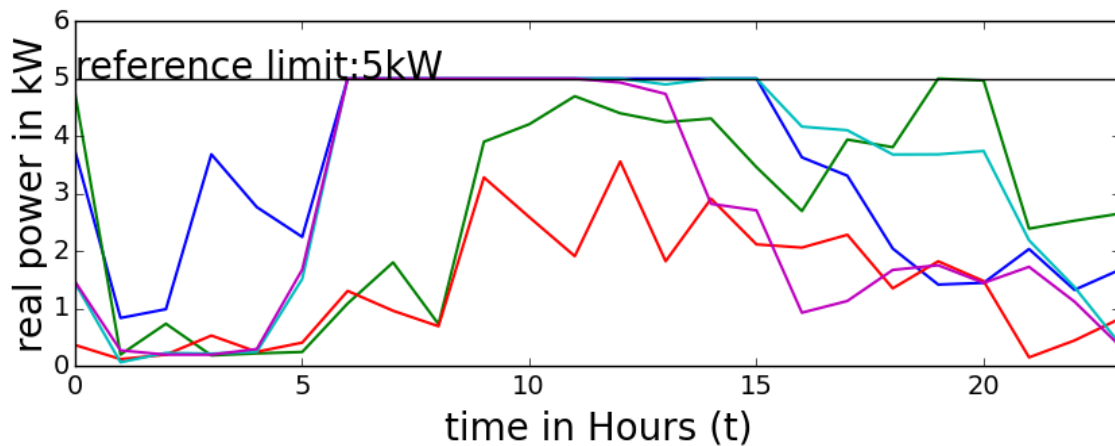


Figure 3.6: $\beta_i(t)$ for 5 randomly selected residential loads out of 4026

The demand profile for 5 residential loads scaled using $\beta_i^{pre}(t)$ at all t is shown in Figure 3.5. The demand profile of the same 5 residential loads out of 4026 scaled using $\beta_i(t)$ at all t is shown in Figure 3.6. The summation of the 4026 residential loads generated using $\beta_i^{pre}(t)$ and $\beta_i(t)$ matches the system load $sys(t)$ at all t and indicated by the 3D plot in Figure 3.7. All codes and simulation files are available in the public domain [30].

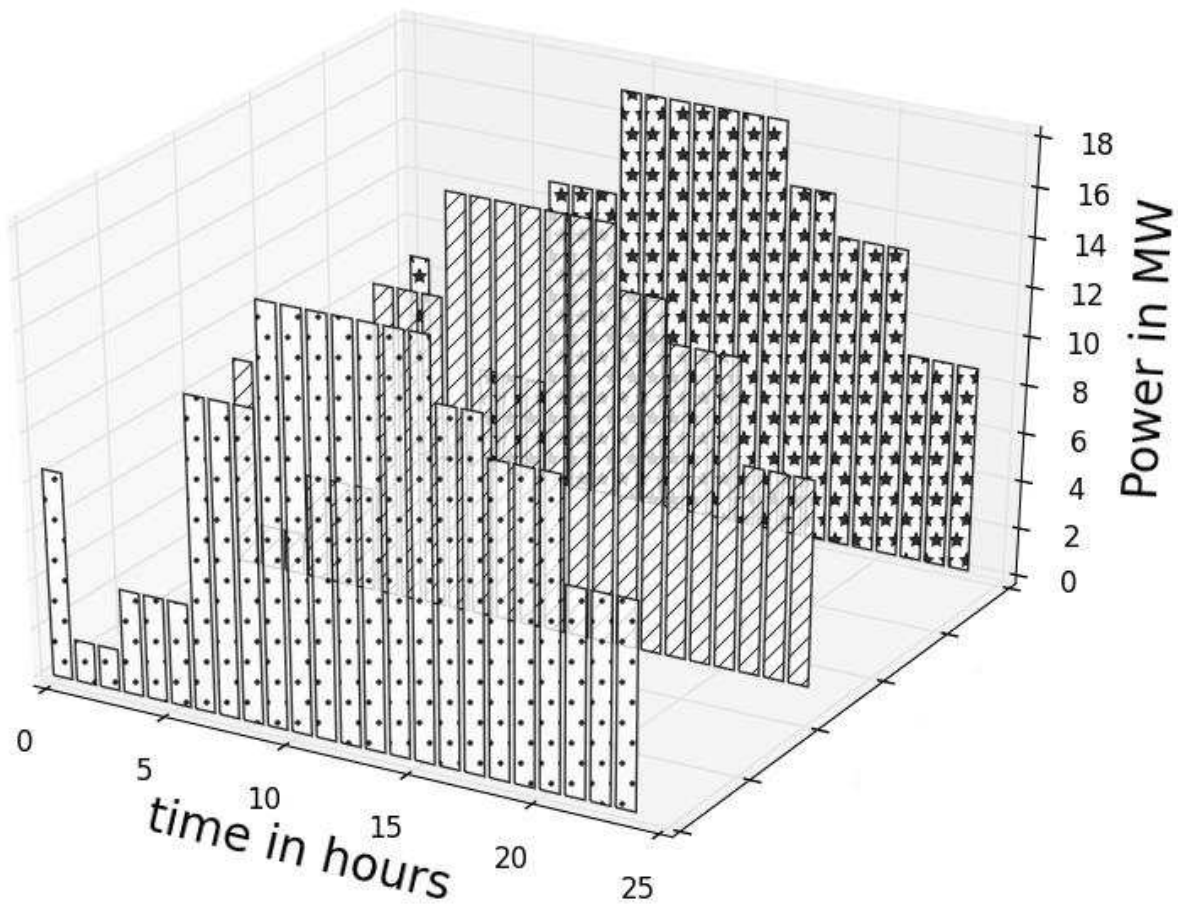


Figure 3.7: 3D plot of the sum of 4026 residential loads generated using $\beta_i^{pre}(t)$ (shown as dotted plot) and $\beta_i(t)$ (shown as hatched plot) compared with the custom created system load $sys(t)$ (shown as starred plot)

3.5 CONCLUSIONS

The algorithms for the creation of the distribution network model consisting of diverse feeder datasets and generation of stochastic load profiles associated with those feeders were developed in the Python coding environment.

The developed algorithms demonstrate the abstract mathematical model for the creation of a detailed distribution topologies consisting of diverse feeders using a limited number of configurable constraints as well as the application of stochastic time series curves for the generation of stochastic load profiles for residential loads that can be used as a complete distribution load model while performing power system studies. By employing a mathematical approach for distribution network creation, complex distribution load models can be swiftly created for a wide time range that provides a near approximation of an existing distribution system. This work is envisioned to enable future studies of detailed co-simulations that link transmission and distribution grids.

CHAPTER 4

CONCLUSION AND FUTURE WORK

4.1 CONCLUSION

Some distribution feeder datasets are created using the proposed feeder modeling algorithms. Design considerations and assumptions for the creation of the distribution assets were addressed. The creation of detailed distribution topologies populated with residential class of consumers using a limited number of configurable constraints is presented with the distribution feeder generation algorithm. The stochastic time series for the generation of load profiles for each residential house objects connected within the distribution feeder were developed and simulated by the stochastic load profile generation algorithm. A graphical user interface (GUI) is developed and implemented for translating the three phase network information stored in a commercial power system simulator to the open source transmission and distribution simulators. The three phase network data recreated in MATPOWER[®] was tested and verified with the source model presented in the commercial simulator. The distribution topologies created using DEx.py were developed and tested by simulating in GridLAB-D[®].

The proposed algorithms are demonstrated on the RBTS candidate system. The two-step algorithms for creation of distribution topologies are modeled and simulated for bus 2 of RBTS. DFGA accurately creates the distribution system populated with 4026 residential house objects that aggregate to the connected load at bus 2 in RBTS. DFGA creates the distribution assets for RBTS bus 2 by limiting the number of residential house objects to be placed at each phase on a load point based on a user configured distribution line. It also optimally rearranges the parallel connections to be placed at each load point at RBTS bus 2 for minimizing the deviation of H_{ideal} residential house objects to a large number. For the same topology, SLPGA creates stochastic load

profiles for each of the 4026 residential house objects derived from a synthetic system load profile with the added constraint that the profiles do not exceed the peak power consumption limit throughout the duration of the time series.

4.2 FUTURE WORK

The immediate scope for future work deals with populating the distribution topologies with diversified load classes ---residential, industrial, and commercial to complete the distribution system modeling. Incorporating all three load categories will help to perform simulations pertaining to stability studies and demand response behavior as residential load class has a disparity in its energy consumption as compared to the commercial and industrial load class. From the user interface perspective, moving the python code base to a web based platform would drastically reduce the downtime of compiling the code and provide a single input-output channel for easy creation of the runtime co-simulation script files. Another future use case scenario presents itself in the form of modeling multiple GridLAB-D[®] feeders as a separate entity and running them in parallel on a High Performance Computing (HPC) environment for long term time step simulations at reduced computational burden. The co-simulation platform will enable to study the impact of the pervasive penetration of the advanced Smart Grid technologies on the electric grid.

REFERENCES

- [1] What is the Smart Grid? (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/xZwldQ>
- [2] Frameworks and data initiatives for Smart Grid and Cyber-Physical systems. (Date last accessed on Sept.18,2016). [Online]. Available: <https://goo.gl/HZG492>
- [3] Smart Grid. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/Xzjm9v>
- [4] What is Smart Grid and why is it important? (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/cOQSKt>
- [5] NIST and the Smart Grid. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/xs18be>
- [6] An assessment of energy technologies and research opportunities. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/NEcuZ5>
- [7] DOE Voice of Experience | Integrating Intermittent Resources. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/dlHlGz>
- [8] M. Pöchacker, A. Sobe, and W. Elmenreich, "Simulating the smart grid," *2013 IEEE Grenoble Conference*, Grenoble, 2013, pp.1-6. doi: 10.1109/PTC.2013.6652259.
- [9] Renewable Systems Interconnection Study-Utility Models, Analysis, and Simulation Tools. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/qW2sa8>
- [10] T. Strasser, M. Stifter, F. Andrén, and P. Palensky, "Co-Simulation Training Platform for Smart Grids," in *IEEE Transactions on Power Systems*, vol. 29, no. 4, pp. 1989-1997, July 2014. doi: 10.1109/TPWRS.2014.2305740.
- [11] T. M. Hansen *et al.*, "Enabling Smart Grid Cosimulation Studies: Rapid Design and Development of the Technologies and Controls," in *IEEE Electrification Magazine*, vol. 4, no. 1, pp. 25-32, March 2016. doi: 10.1109/MELE.2015.2509899.
- [12] R. Kadavil, T. M. Hansen and S. Suryanarayanan, "An algorithmic approach for creating diverse stochastic feeder datasets for power systems co-simulations," *2016 IEEE Power and Energy Society General Meeting (PESGM)*, Boston, MA, pp. 1-5, July 2016. doi: 10.1109/PESGM.2016.7741708.
- [13] R. Billinton and S. Jonnavithula, "A test system for teaching overall power system reliability assessment," in *IEEE Transactions on Power Systems*, vol. 11, no. 4, pp. 1670-1676, Nov 1996. doi: 10.1109/59.544626.
- [14] PowerWorld Simulator. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/Afh72>
- [15] R. D. Zimmerman, C. E. Murillo-Sanchez and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education," in *IEEE*

- Transactions on Power Systems*, vol. 26, no. 1, pp. 12-19, Feb. 2011. doi: 10.1109/TPWRS.2010.2051168.
- [16] D. P. Chassin, K. Schneider and C. Gerkenmeyer, "GridLAB-D: An open-source power systems modeling and simulation environment," *2008 IEEE/PES Transmission and Distribution Conference and Exposition*, Chicago, IL, 2008, pp. 1-5. doi: 10.1109/TDC.2008.4517260.
- [17] PowerWorld Corp. Auxiliary File Format. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/nW6D7P>.
- [18] W. Group, "Common Format For Exchange of Solved Load Flow Data," in *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-92, no. 6, pp. 1916-1925, Nov. 1973. doi: 10.1109/TPAS.1973.293571.
- [19] Python Software Foundation. Python 3.0. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/dCaicT>
- [20] DNV GL Energy. A Review of Distributed Energy Resources. (Date last accessed on Sept. 18, 2016). [Online]. Available: <http://goo.gl/44DoWC>
- [21] University of Washington Electrical Engineering. Power Systems Test Case Archive. (Date last accessed on Sept. 18, 2016). [Online]. Available: <http://www2.ee.washington.edu/research/pstca/>
- [22] A. Bokhari *et al.*, "Experimental Determination of the ZIP Coefficients for Modern Residential, Commercial, and Industrial Loads," in *IEEE Transactions on Power Delivery*, vol. 29, no. 3, pp. 1372-1381, June 2014. doi: 10.1109/TPWRD.2013.2285096.
- [23] W. H. Kersting. "Distribution system modeling and analysis." *Electric Power Generation, Transmission, and Distribution*, Third Edition. CRC press, 2012. 1-58.
- [24] J. J. Grainger, and W. D. Stevenson. *Power system analysis*. McGraw-Hill, 1994.
- [25] Title XIII Smart Grid. *110th Congress of the United States of America, Energy Independence and Security Act of 2007*. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/KqyBFG>
- [26] M. A. Al Faruque and F. Ahourai, "GridMat: Matlab toolbox for GridLAB-D to analyze grid impact and validate residential microgrid level energy management algorithms," *ISGT 2014*, Washington, DC, 2014, pp. 1-5. doi: 10.1109/ISGT.2014.6816479.
- [27] J. Dickert and P. Schegner, "A time series probabilistic synthetic load curve model for residential customers," *2011 IEEE Trondheim PowerTech*, Trondheim, 2011, pp. 1-6. doi: 10.1109/PTC.2011.6019365.
- [28] W. A. Woodward, H. L. Gray, and A. C. Elliott. "Nonstationary time series models". *Applied time series analysis*. CRC press, 2011.
- [29] IEEE test distribution feeders data. (Date last accessed on Sept. 18, 2016). [Online]. Available: <https://goo.gl/xC9WDi>

- [30] Smart Grid Resource Allocation SGRA main project page. (Date last accessed on Sept. 18, 2016). [Online]. Available: <http://projects-web.engr.colostate.edu/sgra/>

APPENDIX A

COPYRIGHT PERMISSION FROM IEEE FOR CHAPTER 2

9/19/2016

Rightslink® by Copyright Clearance Center



RightsLink®

Home

Account Info

Help



Title: Enabling Smart Grid
Cosimulation Studies: Rapid
Design and Development of the
Technologies and Controls

Author: Timothy M. Hansen

Publication: IEEE Electrification Magazine (E-
M)

Publisher: IEEE

Date: March 2016

Copyright © 2016, IEEE

Logged in as:

Rahul Kadavil

LOGOUT

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

- 1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
- 2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
- 3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

- 1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
- 2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
- 3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK

CLOSE WINDOW

Copyright © 2016 Copyright Clearance Center, Inc. All Rights Reserved. [Privacy statement](#). [Terms and Conditions](#). Comments? We would like to hear from you. E-mail us at customercare@copyright.com

APPENDIX B

COPYRIGHT PERMISSION FROM IEEE FOR CHAPTER 3

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

TITLE OF PAPER/ARTICLE/REPORT, INCLUDING ALL CONTENT IN ANY FORM, FORMAT, OR MEDIA (hereinafter, "the Work"):

An Algorithmic Approach for Creating Diverse Stochastic Feeder Datasets for Power System Co-simulations [working title]

COMPLETE LIST OF AUTHORS:

Rahul Kadavil, Timothy M. Hansen, and Siddharth Suryanarayanan

IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book): IEEE Power & Energy Society Conference

COPYRIGHT TRANSFER

1. The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the above Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

CONSENT AND RELEASE

2. In the event the undersigned makes a presentation based upon the Work at a conference hosted or sponsored in whole or in part by the IEEE, the undersigned, in consideration for his/her participation in the conference, hereby grants the IEEE the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive, in any format or medium, whether now known or hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IEEE and live or recorded broadcast of the Presentation during or after the conference.

3. In connection with the permission granted in Section 2, the undersigned hereby grants IEEE the unlimited, worldwide, irrevocable right to use his/her name, picture, likeness, voice and biographical information as part of the advertisement, distribution and sale of products incorporating the Work or Presentation, and releases IEEE from any claim based on right of privacy or publicity.

4. The undersigned hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data or other material from the works of others, the undersigned has obtained any necessary permissions. Where necessary, the undersigned has obtained all third party permissions and consents to grant the license above and has provided copies of such permissions and consents to IEEE.

Please check this box if you do not wish to have video/audio recordings made of your conference presentation.

See reverse side for Retained Rights/Terms and Conditions, and Author Responsibilities.

GENERAL TERMS

- The undersigned represents that he/she has the power and authority to make and execute this assignment.
- The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
- In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall become null and void and all materials embodying the Work submitted to the IEEE will be destroyed.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.

(1) _____
Author/Authorized Agent for Joint Authors

09/18/16
Date

U.S. GOVERNMENT EMPLOYEE CERTIFICATION (WHERE APPLICABLE)

This will certify that all authors of the Work are U.S. government employees and prepared the Work on a subject within the scope of their official duties. As such, the Work is not subject to U.S. copyright protection.

(2) _____
Authorized Signature

Date

(Authors who are U.S. government employees should also sign signature line (1) above to enable the IEEE to claim and protect its copyright in international jurisdictions.)

CROWN COPYRIGHT CERTIFICATION (WHERE APPLICABLE)

This will certify that all authors of the Work are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties. As such, the Work is subject to Crown Copyright and is not assigned to the IEEE as set forth in the first sentence of the Copyright Transfer Section above. The undersigned acknowledges, however, that the IEEE has the right to publish, distribute and reprint the Work in all forms and media.

(3) _____
Authorized Signature

Date

(Authors who are British or British Commonwealth Government employees should also sign line (1) above to indicate their acceptance of all terms other than the copyright transfer.)
rev. 032515

IEEE COPYRIGHT FORM (continued)

RETAINED RIGHTS/TERMS AND CONDITIONS

General

1. Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.
2. Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.
3. In the case of a Work performed under a U.S. Government contract or grant, the IEEE recognizes that the U.S. Government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract/grant so requires.
4. Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. The IEEE Intellectual Property Rights office must handle all such third-party requests.
5. Authors whose work was performed under a grant from a government funding agency are free to fulfill any deposit mandates from that funding agency.

Author Online Use

6. **Personal Servers.** Authors and/or their employers shall have the right to post the accepted version of IEEE-copyrighted articles on their own personal servers or the servers of their institutions or employers without permission from IEEE, provided that the posted version includes a prominently displayed IEEE copyright notice and, when published, a full citation to the original IEEE publication, including a link to the article abstract in IEEE Xplore. Authors shall not post the final, published versions of their papers.
7. **Classroom or Internal Training Use.** An author is expressly permitted to post any portion of the accepted version of his/her own IEEE-copyrighted articles on the author's personal web site or the servers of the author's institution or company in connection with the author's teaching, training, or work responsibilities, provided that the appropriate copyright, credit, and reuse notices appear prominently with the posted material. Examples of permitted uses are lecture materials, course packs, e-reserves, conference presentations, or in-house training courses.
8. **Electronic Preprints.** Before submitting an article to an IEEE publication, authors frequently post their manuscripts to their own web site, their employer's site, or to another server that invites constructive comment from colleagues. Upon submission of an article to IEEE, an author is required to transfer copyright in the article to IEEE, and the author must update any previously posted version of the article with a prominently displayed IEEE copyright notice. Upon publication of an article by the IEEE, the author must replace any previously posted electronic versions of the article with either (1) the full citation to the IEEE work with a Digital Object Identifier (DOI) or link to the article abstract in IEEE Xplore, or (2) the accepted version only (not the IEEE-published version), including the IEEE copyright notice and full citation, with a link to the final, published article in IEEE Xplore.

INFORMATION FOR AUTHORS

Author Responsibilities

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements as stated in section 8.2.1 of the IEEE PSPB Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE's publishing policies may be found at http://www.ieee.org/publications_standards/publications/rights/authorresponsibilities.html. Authors are advised especially of IEEE PSPB Operations Manual section 8.2.1.B12: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE PSPB Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."

Author/Employer Rights

If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IEEE assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

IEEE Copyright Ownership

It is the formal policy of the IEEE to own the copyrights to all copyrightable material in its technical publications and to the individual contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and, at the same time, to facilitate the appropriate re-use of this material by others. The IEEE distributes its technical publications throughout the world and does so by various means such as hard copy, microfiche, microfilm, and electronic media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various compendiums, collective works, databases and similar publications.

THIS FORM MUST ACCOMPANY THE SUBMISSION OF THE AUTHOR'S MANUSCRIPT.

Questions about the submission of the form or manuscript must be sent to the publication's editor.

Please direct all questions about IEEE copyright policy to:

IEEE Intellectual Property Rights Office, copyrights@ieee.org, +1-732-562-3966 (telephone)

APPENDIX C

APPLICATION OF THE FEEDER MODELING ALGORITHMS ON BUS 2 IN THE ROY BILLINTON TEST SYSTEM (RBTS)

For the simulation study in Chapter 3, bus-2 from the RTBS system was chosen as the transmission node with $LP_{total} = 22$ and $L = 20\text{MW}$. With P_{limit} set as 5kW for a residential load, the ampacity of a distribution line at each phase in the LV network, at each LP is set at 1040 A, and γ is set at 88% to obtain $LP_{max} = 66$. For bus 2, we have to distribute $H_{ideal} = 4000$ residential loads across $LP_{total} = 22$ load points for which LP_{actual} (181) is greater than LP_{max} . If we create $p_o = 3$ parallel distribution lines across each phase in LV for each LP , the total residential loads become $H_{actual} = 4356$, which translates to 21.78MW and is greater than L . The DFGA optimizes δ' by computing the parallel connections to be placed at each LP for $H_{actual}^{new} = 4026$ residential loads across $LP_{total} = 22$ load points.