SANDIA REPORT
SAND2015-10303
Unlimited Release
Printed November 2015

Tools for Enhanced Grid Operation and Optimized PV Penetration Utilizing Highly Distributed Sensor Data

Matthew J. Reno, Jouni Peppanen, John Seuss, Matthew Lave, Robert J. Broderick, Santiago Grijalva

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2015-10303
Unlimited Release
Printed November 2015

Tools for Enhanced Grid Operation and Optimized PV Penetration Utilizing Highly Distributed Sensor Data

Matthew J. Reno, Matthew Lave, Robert J. Broderick
Photovoltaics and Distributed Systems Integration
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-1033

Jouni Peppanen, John Seuss, Santiago Grijalva
School of Electrical and Computer Engineering
Georgia Institute of Technology
777 Atlantic Drive NW
Atlanta, GA 30332-0250

Abstract

Increasing numbers of PV on distribution systems are creating more grid impacts, but it also provides more opportunities for measurement, sensing, and control of the grid in a distributed fashion. This report demonstrates three software tools for characterizing and controlling distribution feeders by utilizing large numbers of highly distributed current, voltage, and irradiance sensors. Instructions and a user manual is presented for each tool. First, the tool for distribution system secondary circuit parameter estimation is presented. This tool allows studying distribution system parameter estimation accuracy with user-selected active power, reactive power, and voltage measurements and measurement error levels. Second, the tool for multi-objective inverter control is shown. Various PV inverter control strategies can be selected to objectively compare their impact on the feeder. Third, the tool for energy storage for PV ramp rate smoothing is presented. The tool allows the user to select different storage characteristics (power and energy ratings) and control types (local vs. centralized) to study the tradeoffs between state-of-charge (SOC) management and the amount of ramp rate smoothing.

CONTENTS

CONTENTS 5
FIGURES 7
TABLES 7
NOMENCLATURE 8
1. INTRODUCTION 9
1.1. Requirements 10
2. DISTRIBUTION SYSTEM SECONDARY CIRCUIT PARAMETER ESTIMATION TOOL 11
2.1. Files 11
2.2. Sample Data 11
2.3. GUI Functionality 12
2.3.1. File Selection 13
2.3.2. Sample Size Selection 13
2.3.3. Measurement Error Level Selection 14
2.3.4. Options 14
2.3.5. Circuit Plot 14
2.3.6. Output 14
2.3.7. Estimate Parameters 14
2.3.8. Close Figures & Clear GUI 15
2.4. Parameter Estimation Methodology 15
2.5. Summary 16
3. MULTI-OBJECTIVE INVERTER CONTROL SNAPSHOT SIMULATION TOOL 19
3.1. Files 19
3.2. Circuit Data Format 19
3.3. Available Inverter Control Types 20
3.3.1. Zero Current Injection 20
3.3.2. Local Volt/Watt Control 20
3.3.3. Local Volt/Var Control 21
3.3.4. Centralized Fair Dispatch 22
3.3.5. Centralized Sensitivity-Based Dispatch 22
3.4. GUI Functionality 23
3.4.1. Load Circuit File 23
3.4.2. Circuit Curtailment Plot 24
3.4.3. Voltage Profile Plot 24
3.4.4. Control Type Selection 24
3.4.5. Load and Irradiance Selection 24
3.4.6. Result Totals 24
3.4.7. Operational Notes 24

3.5. Summary 25
4. ENERGY STORAGE FOR PV RAMP RATE SMOOTHING TOOL 27
4.1. Files 27
4.2. Sample Data 27
4.3. GUI Functionality 28
4.3.1. Battery Parameters 29
4.3.2. Battery Control 30
4.3.3. Time Frame 30
4.3.4. Figures 31
4.3.5. Simulation Animation 32
4.4. Summary 33
REFERENCES 34

FIGURES

Figure 1: GUI parameterEstimationGUI in Matlab 13
Figure 2: Summary of the 66-node test circuit parameter estimation accuracy with 1 week of measurement samples without measurement error 16
Figure 3: Accuracy of the 66-node test circuit R and X parameters estimated with 1 week of measurement samples without measurement error 17
Figure 4: Accuracy of the 66-node test circuit Z and X/R parameters estimated with 1 week of measurement samples without measurement error 17
Figure 5: Accuracy of the 66-node test circuit load voltages simulated with parameter estimated with 1 week of measurement samples without measurement error 18
Figure 6: Accuracy of the 66-node test circuit secondary circuit voltage drops (from the service transformer primary to the load buses) simulated with parameter estimated with 1 week of measurement samples without measurement error 18
Figure 7. Curve used by local Volt/Watt inverter control to determine power curtailment based on locally measured voltage. 21
Figure 8. Curve used by local Volt/Var inverter control to determine the proportion of available reactive power to output based on locally measured voltage. 22
Figure 9. Layout of inverter control simulation tool GUI. 23
Figure 10. PV system pairing with irradiance sensors (left) and time offset for 24 m/s cloud speed (right). 28
Figure 11. GUI in MATLAB 29
Figure 12. Battery parameters window. 29
Figure 13. Battery control window. 30
Figure 14. Battery control diagram. 30
Figure 15. Time frame window. 31
Figure 16. Figure and plotting window. 32
Figure 17. Simulation animation window. 33

TABLES

Table 1: Matlab scripts and functions that the GUI uses 11
Table 2: Matlab scripts and functions that the GUI uses 19
Table 3: MATLAB scripts and functions that the GUI uses 27

NOMENCLATURE

AMI	advanced metering infrastructure
DOE	Department of Energy
GHI	global horizontal irradiance
GUI	graphical user interface
OpenDSS	Open Distribution System Simulator™
POA	plane of array
PV	photovoltaic
SOC	state of charge
ZCI	zero current injection

1. INTRODUCTION

Increasing numbers of PV on distribution systems are creating more grid impacts [1, 2], but it also provides more opportunities for measurement, sensing, and control of the grid in a distributed fashion [3, 4]. In order to achieve very high PV penetration scenarios (well beyond 100% of peak load at the feeder level) it will be necessary to leverage distributed inverters to increase situational awareness and provide local voltage support. Utilizing highly distributed current, voltage, frequency, and irradiance sensors, we present methods to gain a more granular understanding of distribution grid operations, and to enhance and validate distribution feeder models. Based on this enhanced understanding, control methods are proposed for determining the optimal control strategies of PV and storage in order to maximize the amount of PV that can safely and reliably be installed on a distribution feeder. The portion of residential PV inverters that are communication enabled has increased recently and is expected to accelerate due to grid support requirements such as California Rule 21, highlighting the relevance and applicability of optimal control strategies for PV inverters to mitigate impacts.

This report demonstrates three software tools for characterizing and controlling distribution feeders by utilizing large numbers of highly distributed current, voltage, and irradiance sensors. Instructions and a user manual is presented for each tool. First, the tool for distribution system secondary circuit parameter estimation is presented. Second, the tool for multi-objective inverter control is shown. Third, the tool for energy storage for PV ramp rate smoothing is presented.

The parameter estimation tool is created for demonstrating the accuracy of the distribution system secondary circuit parameter estimation methodology discussed in [5]. The tool allows studying distribution system parameter estimation accuracy with user-selected AMI active power, reactive power, and voltage measurements and measurement error levels. The tool can be also used to estimate the secondary circuit parameters of other balanced 3-phase OpenDSS distribution system models given that certain assumptions and conventions are fulfilled.

The multi-objective inverter control tool is compared to the effectiveness of several photovoltaic (PV) inverter control strategies that have differing objectives. The tool allows the user to quickly see how well each control type mitigates PV-induced over-voltage violations at a given irradiance and load level. The efficacy of each control is further demonstrated by visualizing the overall real power curtailment and reactive power generation, as well as their distribution across the PV in the circuit. The tool interfaces with the OpenDSS open-source distribution power flow software. It uses OpenDSS circuit models and a combination of custom Matlab inverter controls and built-in OpenDSS inverter controls.

The energy storage for PV ramp rate smoothing tool allows the user to select different storage characteristics (power and energy ratings) and control types (local vs. centralized). The gains in the feedback control are also modifiable by the user to study the tradeoffs between state-of-charge (SOC) management and the amount of ramp rate smoothing. The tool includes example PV output data for a 1-week period. The simulation results can also be played in a video animation format through the simulation period. The purpose of the GUI is to demonstrate the effectiveness of energy storage in mitigating PV variability. The interactive nature of the GUI allows the user to investigate different control types and options.

1.1. Requirements

The tool can either be run using MATLAB or as an executable version.

- The MATLAB tool requires an actual version of MATLAB, OpenDSS [6], and MATLAB GridPV Toolbox [7, 8]. The GUI has been tested on a 64-bit Windows 7 with a 64-bit MATLAB R2015a and OpenDSS v. 7.6.4.70, and MATLAB GridPV toolbox v. 2.2.
- The executable version requires OpenDSS [6] and that the free R2014a MATLAB Runtime is installed (http://www.mathworks.com/products/compiler/mcr/).

By default, the GUI assumes that all the required input files and MATLAB scripts and functions are located in the current MATLAB folder.

2. DISTRIBUTION SYSTEM SECONDARY CIRCUIT PARAMETER ESTIMATION TOOL

This chapter provides instructions for using the parameter estimation Matlab tool. The tool is created for demonstrating the accuracy of the distribution system secondary circuit parameter estimation methodology discussed in [5]. The tool allows studying distribution system parameter estimation accuracy with user-selected AMI active power, reactive power, and voltage measurements and measurement error levels.

The tool is tested on a 66-node 3-phase balanced distribution system with 10 secondary circuits. Details of the test circuit can be found in [1]. The tool can be also used to estimate the secondary circuit parameters of other balanced 3-phase OpenDSS distribution system models given that certain assumptions and conventions are fulfilled. Otherwise, the underlying Matlab functions may require minor modifications.

2.1. Files

The Matlab script and function files that the GUI uses are listed in Table 3.

Table 1: Matlab scripts and functions that the GUI uses

Name	Functionality
estimateSecondaryCircuitParameters	Estimate the secondary circuit parameters for the circuit DSSCircObj using the provided measurements
mainScript	Runs the main script to analyze parameter estimation on the 66-node 3-phase test circuit
parameterEstimationGUI	MATLAB code for the GUI
plotCircuit	Plots a given secondary circuit
rotateXLabels	Rotates Matlab figure xticklabels
runTimeSeries	Runs a timeseries powerflow storing the bus voltages at each time step

2.2. Sample Data

The required input files (default file name) are: the load active power (in kW) measurement csv-file (PtrueFile.csv), load reactive power (in kVAr) measurement csv-file (QtrueFile.csv), load voltage measurement (in Volts) csv-file (VtrueFile.csv), the dss-file containing the definitions for the OpenDSS model of the analyzed circuit (compileCircuit.dss), and the dss-file containing the loadshape definitions for reloading the loadshapes for a compiled OpenDSS circuit model (ReloadLoadShapes.dss). The file compileCircuit.dss additionally redirects to BusCoords.dss file that contains the bus coordinates for plotting purposes.

The measurement file PtrueFile.csv, QtrueFile.csv, and VtrueFile.csv generation is explained in detail in [1]. Next, a brief summary is provided. The active power measurement profiles in the PtrueFile.csv file were created by randomly assigning each of the 66-node circuit load a PecanStreet Inc. [9] meter and the corresponding hourly active power measurements. Each load in the circuit was assigned a peak kW and kVAr based on the number of loads in the given secondary circuit and the kVA rating of the service transformer feeding the secondary circuit. Then, the assigned active power measurement profiles were scaled so that the mean of each profile equals the load rated kW. All values exceeding a selected peak load kW were set randomly to 60-100% of the load kW and all negative or zero load values were set to random values 5-15% of load kW. The load reactive power consumption profiles in the QtrueFile.csv file were calculated from the active power profiles by utilizing a random power factor. Meter i reactive power consumption at time k was calculated with

$$Q_{i,k} = P_{i,k} * \sqrt{(1 - (PF)_{i,k}^2}, \quad (1)$$

where $P_{i,k}$ is meter i active power measurement at time k (in the PtrueFile.csv file) and $(PF)_{i,k}$, the meter i power factor at time k, was set to random uniform number: $(PF)_{i,k}$~$\mathrm{Uniform}(0.9{,}1.0)$. The load voltage "measurements" in the VtrueFile.csv file were acquired by solving the time series power flow simulation with the loads varying according to their real and reactive power profiles in PtrueFile.csv and QtrueFile.csv.

Detailed explanation of the test 66-node test circuit can be found in [1]. The compileDSS.dss file contains the circuit description for OpenDSS including the definitions for the voltage source (substation), loadshapes, transformers, lines, and loads that are modeled with fixed active power and reactive power profiles. Detailed explanation of the different element definitions can be found in OpenDSS manual.

2.3. GUI Functionality

The tool is implemented as a Matlab graphical user interface. The GUI can be executed in Matlab by typing the command "parameterEstimationGUI" or by double clicking the icon parameterEstimationGUI.fig in the "Current Folder" in Matlab. The main window of the GUI is shown in Figure 11.

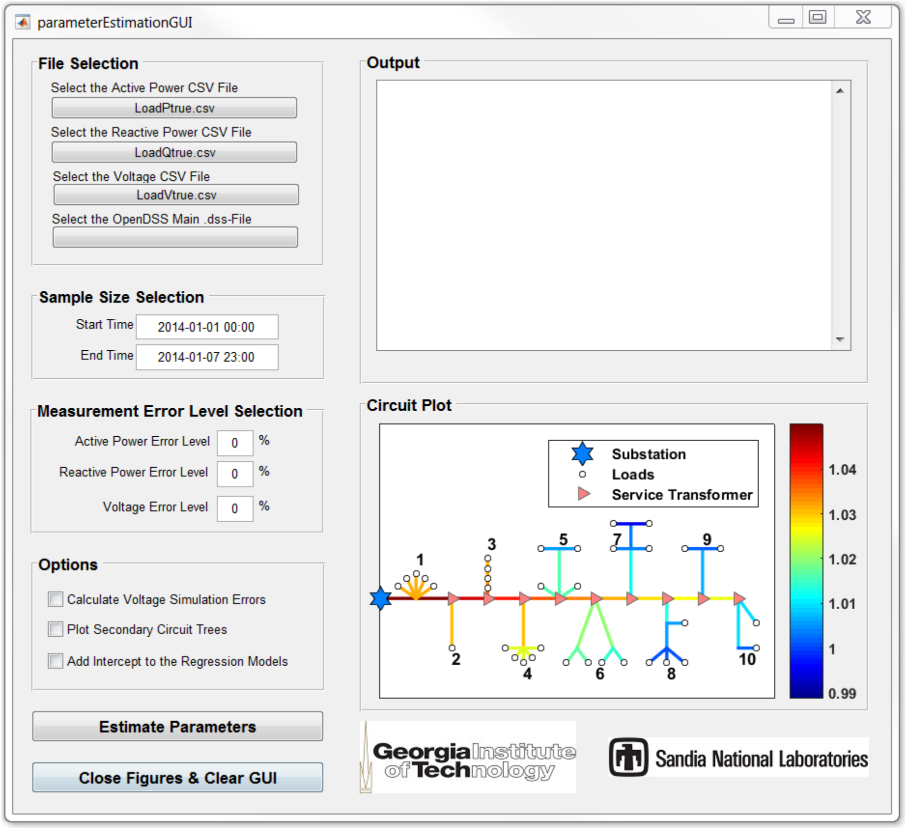

Figure 1: GUI parameterEstimationGUI in Matlab

2.3.1. File Selection

The upper left corner of the GUI includes “File Selection” panel, where the user can select csv-files for the load active power, reactive power, and voltages. The files must have the following structure. The first row has load names, and the following rows contain the measurements for each time stamp. The first column has the measurement time stamps and each other column contains the measurements for one of the meters. The time stamps must have the format “yyyy-mm-dd HH:MM”, where “yyyy” is the four digit years, “mm” is the two-digit months, “dd” is the two-digit day, “HH” is the two-digit hours, and “MM” is the two-digit minutes. The meter order in the file must match with the loadshape column defined in the .dss-file.

The user can also select the .dss-file that contains the OpenDSS model definitions for the distribution feeder. The selected file must include all the necessary circuit definitions (potentially by redirecting to other files) including service transformers, secondary circuit lines, and loads. The circuit is expected to be 3-phase. Coordinates are needed in order to plot the circuit line diagram in the GUI. The loadshape definitions must match with the loadshape definitions in “ReloadLoadShapes.dss” file, which is needed for reloading the load shapes into OpenDSS.

2.3.2. Sample Size Selection

Below the “File Selection” panel is located the sample size selection panel where the user can select the desired range of the measurement data that is used to estimate the parameters. The range is selected by typing the start time stamp and the end time stamp that are given in the same format as the input .csv-files have.

2.3.3. Measurement Error Level Selection

The "Measurement Error Level Selection" panel allows the user to select, e, the percentage level of measurement error that is applied to the measurements. The error is added to each measurement value x with

$$x = x(1 + \frac{e}{100}\hat{x})(-1 + 2\epsilon), \tag{2}$$

where $\hat{x}$ is the value of the largest sample of x and ϵ is a uniformly (0,1) distributed random value, i.e., $\epsilon \sim \mathrm{Uniform}(0,1)$. Separate measurement error levels can be selected for active power, reactive power, and voltage measurements. Parameter estimation is more sensitive to voltage measurement error and thus, voltage error levels beyond 1% are not reasonable [1].

2.3.4. Options

The user can select a desired number of the provided three options.

- "Calculate Voltage Simulation Errors" option makes the program to run another time series power flow with the estimated parameters to calculate the voltage simulation errors.
- "Plot Secondary Circuit Trees" plots the tree layouts of each secondary circuit when the secondary circuit parameters are estimated.
- "Add Intercept to the Regression Models" adds an intercept term to each regression models.

2.3.5. Circuit Plot

The "Circuit Plot" window plots the feeder model with the aid of the "plotCircuitLines.m" function [3]. The plotting is optimized for the 66-node 3-phase test circuit. For this circuit, also the secondary circuit names (numbers) are plotted. By default, the circuit line colors are contoured with the bus per-unit voltages.

2.3.6. Output

The output windows plot the parameter estimation progress, some error messages and key results. Other results are summarized in separate plots.

2.3.7. Estimate Parameters

This button estimates the secondary circuit parameters of the OpenDSS circuit model defined in the user-selected .dss-file. The parameters are estimated with the selected period of load data in the user-selected load data.csv-files.

2.3.8. Close Figures & Clear GUI

This button closes all open Matlab figures and clears the GUI output console.

2.4. Parameter Estimation Methodology

Once the button "Estimate Parameters" is clicked, the secondary circuit series resistance and reactance parameters of the selected circuit model are estimated. The parameters are estimated with the methodology discussed in [1].

The parameter estimation method relies on a number of assumptions and simplifications including the following:

- The feeder is purely radial
- Every transformer in the circuit model is a step-down service transformer feeding a radial secondary circuit consisting purely on lines and loads
- There is one and only one load at every leaf bus of each secondary circuit tree
- The parameter estimation considers only positive sequence series impedance parameters
- The OpenDSS circuit model is assumed to provide the line per-length impedances and line lengths in the same units
- The active power, reactive power, and voltages of all loads are measured.
- The primary circuit is assumed to be perfectly modeled.

These and other assumptions are discussed in detail in [1].

After the button "Estimate Parameters" is pressed, in the essence the following tasks are done:

1. Validate input files and start and end time stamps
2. Import load power and voltage measurements to Matlab and write TempOpenDSSfileP.csv and TempOpenDSSfileQ.csv that contain the loadshapes for the selected time period
3. Add noise to the measurements with the approach elaborated above
4. Run time series power flow with "runTimeSeries.m"
    - Compile OpenDSS circuit model
    - At each time step, solve power flow and store bus voltages
5. Get transformer simulated medium-voltage measurements referred to the low-voltage side
6. Estimate parameters with "estimateSecondaryParameters"
    - Get load current and voltage measurements
    - Store necessary circuit information
    - Merge secondary circuit components whose parameters cannot be estimated
    - For each secondary circuit, proceed from leaf nodes to the root node estimate the R and X parameters
    - Plot the secondary circuit trees (if user selected the option "Plot Secondary Circuit Trees")
    - Set the estimated parameters to the OpenDSS circuit model (if user selected the option "Calculate Voltage Simulation Errors")
7. Print results to the GUI console the create result plots

8. Run another time series power flow with the estimated parameters with "runTimeSeries.m" (if user selected the option "Calculate Voltage Simulation Errors")
9. Plot a summary of the simulated voltages (if user selected the option "Calculate Voltage Simulation Errors").

2.5. Summary

Automated distribution secondary circuit parameter estimation can be a useful tool for improving the accuracy and scope of existing utility feeder models. This chapter introduced a Matlab tool for studying distribution system secondary circuit parameter estimation accuracy. The functionality, methodology, and required assumptions and simplifications are briefly discussed. For further information about the methodology, the reader is directed to [1]. This tool is mainly intended for analyzing how secondary circuit parameter estimation accuracy depends on the measurement data and various model parameters.

The tool includes a 66-node test circuit with load active power, reactive power, and voltage measurements. The parameter estimation accuracy with 1 week of measurement samples without measurement error is summarized in Figure 2. The average estimation error is very small and even the worst case error is acceptable. The accuracy can be increased by selecting a larger number of measurement samples. Figure 3 shows the detailed accuracy of the estimated R and X parameters, and Figure 4 shows the detailed accuracy of the estimated Z and X/R parameters. As discussed in [1], the accuracy of individual estimated parameters depends on the measurement data characteristics, secondary circuit topology, etc. Figure 5 shows the accuracy of the load voltages simulated with the parameters estimated with 1 week of load data. Finally, Figure 6 demonstrates how accurately the secondary circuit voltage drops are captured with the estimated parameters.

Parameter estimation accuracy is strongly driven by the characteristics of the utilized measurement data. Therefore, the tool allows the user to study how parameter estimation accuracy can be increased by selecting a longer period of measurement data and how measurement error has a negative impact on the estimation accuracy.

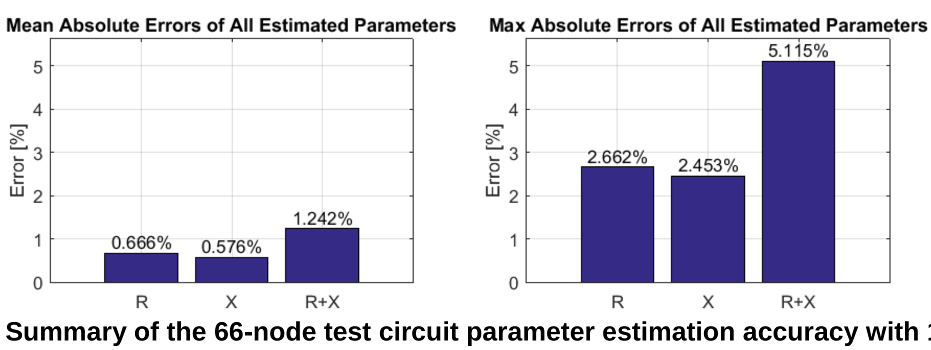

Figure 2: Summary of the 66-node test circuit parameter estimation accuracy with 1 week of measurement samples without measurement error

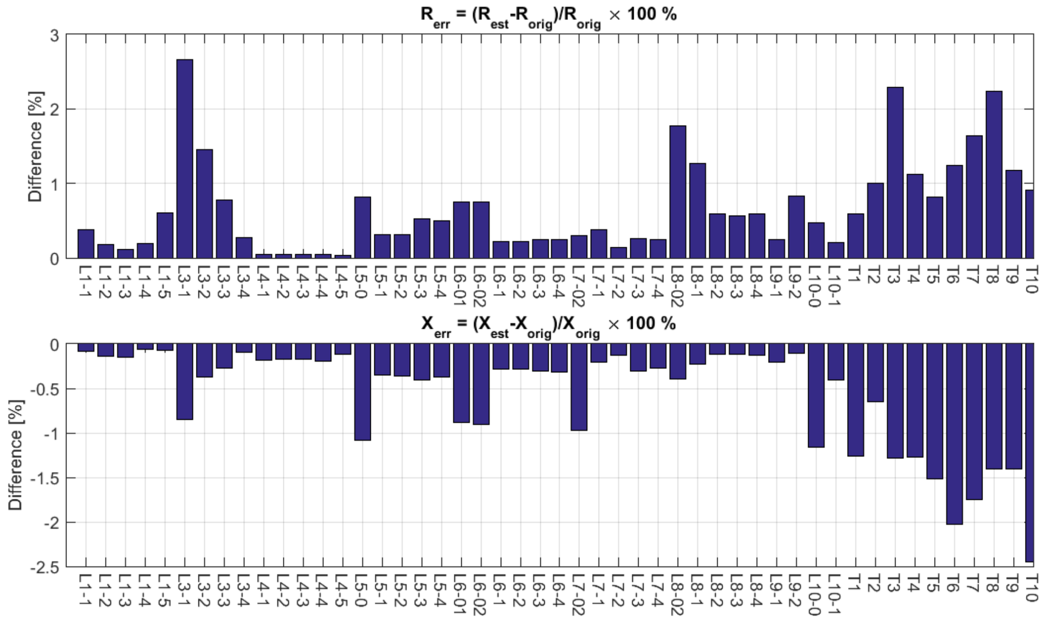

Figure 3: Accuracy of the 66-node test circuit R and X parameters estimated with 1 week of measurement samples without measurement error

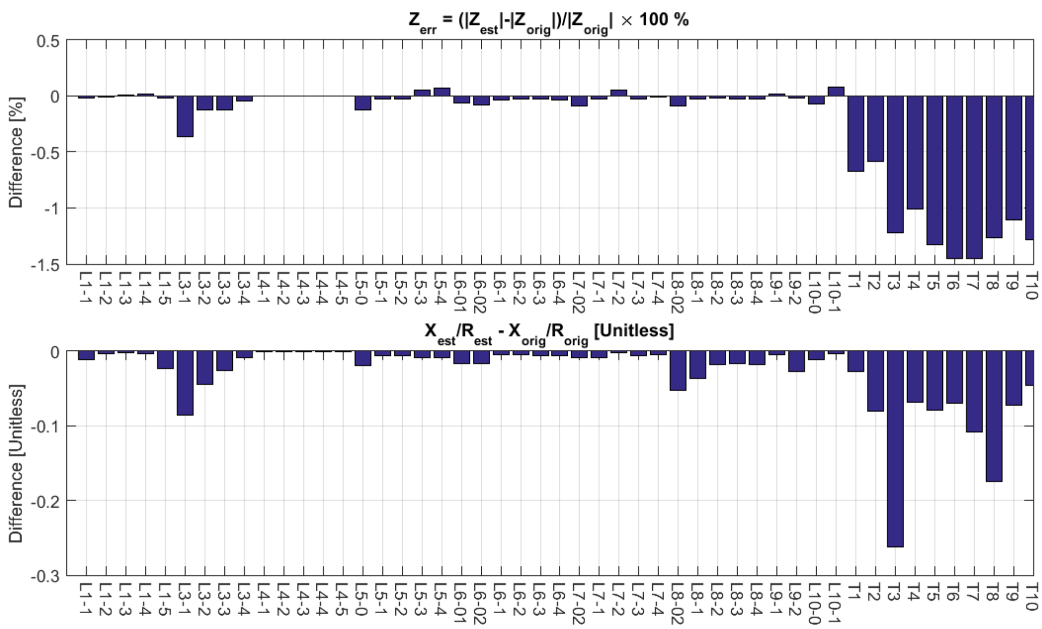

Figure 4: Accuracy of the 66-node test circuit Z and X/R parameters estimated with 1 week of measurement samples without measurement error

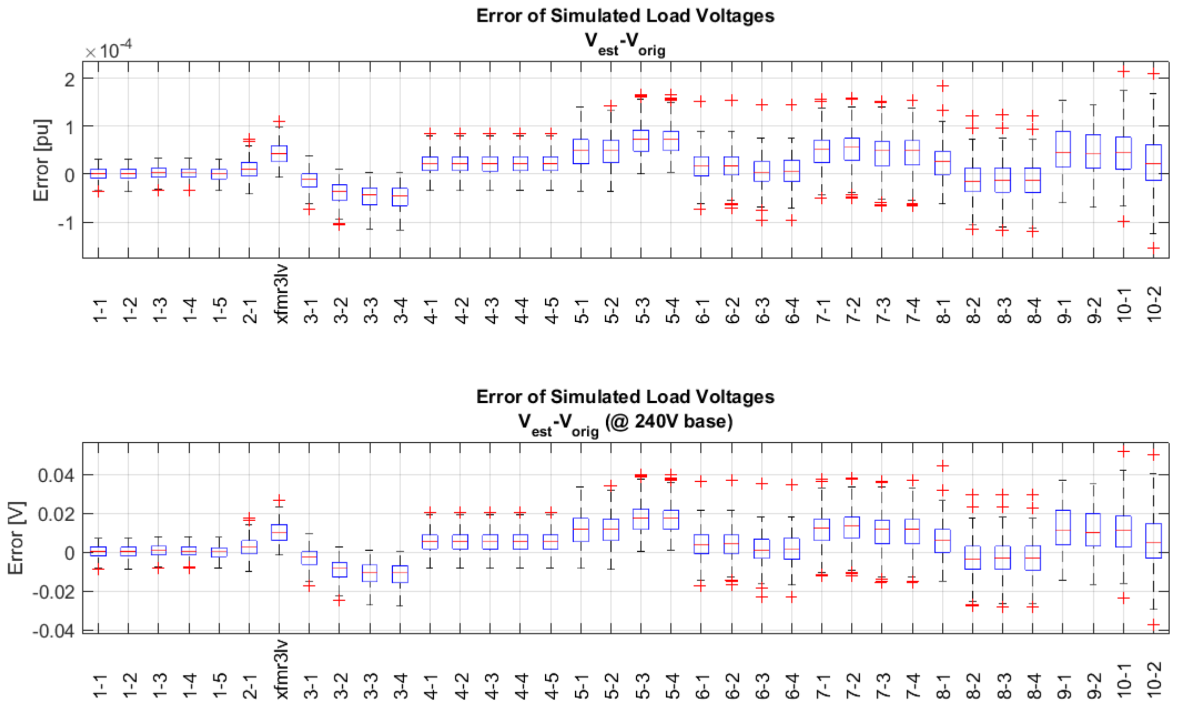

Figure 5: Accuracy of the 66-node test circuit load voltages simulated with parameter estimated with 1 week of measurement samples without measurement error

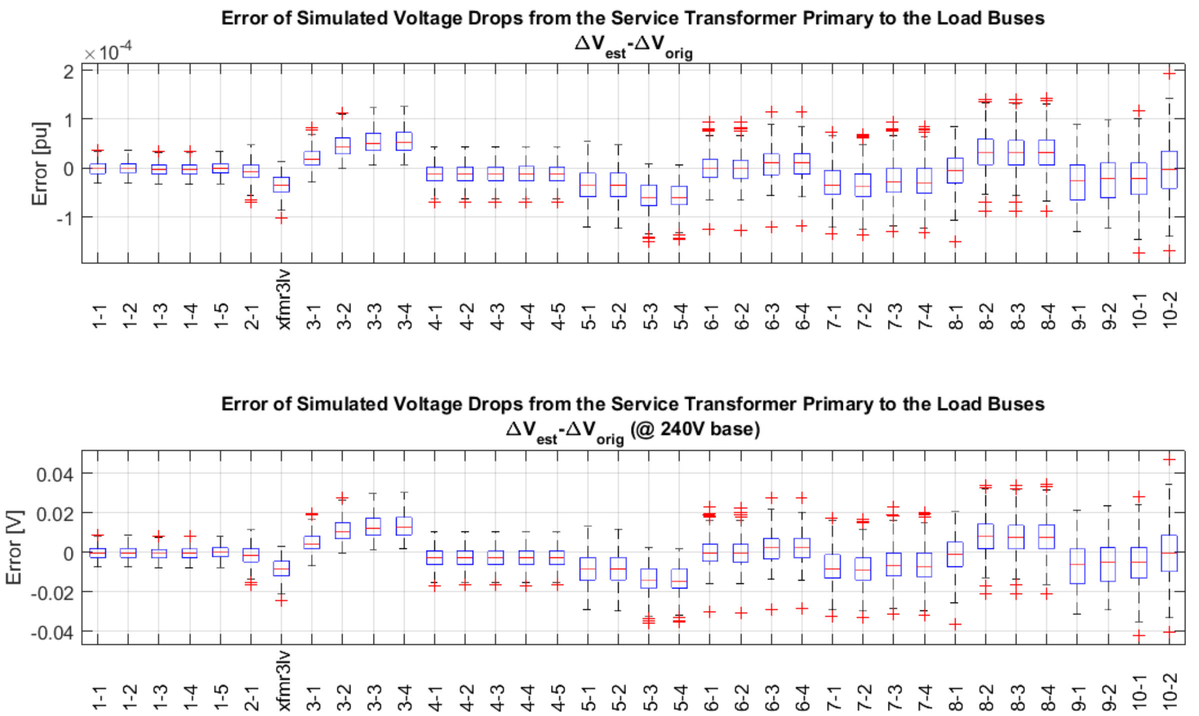

Figure 6: Accuracy of the 66-node test circuit secondary circuit voltage drops (from the service transformer primary to the load buses) simulated with parameter estimated with 1 week of measurement samples without measurement error

3. MULTI-OBJECTIVE INVERTER CONTROL SNAPSHOT SIMULATION TOOL

This chapter provides instructions for using the multi-objective inverter control simulation Matlab tool. The purpose of the tool is to compare the effectiveness of several photovoltaic (PV) inverter control strategies that have differing objectives. The tool allows the user to quickly see how well each control type mitigates PV-induced over-voltage violations at a given irradiance and load level. The efficacy of each control is further demonstrated by visualizing the overall real power curtailment and reactive power generation [10, 11], as well as their distribution across the PV in the circuit. The tool interfaces with the OpenDSS open-source distribution power flow software. It uses OpenDSS circuit models and a combination of custom Matlab inverter controls and built-in OpenDSS inverter controls.

3.1. Files

All Matlab files used in the GUI are listed in Table 3. The necessary format of the OpenDSS used by the GUI is described in Section 1.3

Table 2: Matlab scripts and functions that the GUI uses

Name	Functionality
CurtailmentGUI.fig	Main GUI file used to run tool
CurtailmentGUI.m	Contains all code to run OpenDSS power flows and inverter controls

The demonstration circuit that comes with the tool is a 5469-node 3-phase unbalanced distribution system with 2079 single-phase PV systems placed at each load. The PV systems are sized proportional to 60% of the peak load (250% minimum daytime load) of the customer to which they are connected. The inverter rating of each system is assumed to be equal to the PV system DC rating. Details of the test circuit can be found in [12]. The tool can be used to simulate the performance of the inverter controls on other OpenDSS distribution system models, given any arbitrary placement of PV.

3.2. Circuit Data Format

The only required user data is the circuit *.dss files in OpenDSS format. The OpenDSS file must include PV systems with the following parameter conditions:

- Phases = 1
- pmpp > 0
- kVA = 1.01*pmpp
- irradiance = 1
- VarFollowInverter=yes
- Enabled=yes

A detailed description of the demonstration circuit can be found in [12]. The example circuit in Master.dss contains the circuit description for OpenDSS including the definitions for the voltage source (substation), loadshapes, transformers, lines, and loads that are modeled with fixed active power and reactive power profiles. Detailed explanation of the different element definitions can be found in OpenDSS manual.

In addition to the PV system requirements, buses must have geographical coordinates defined in order for the circuit diagram plot to function properly. Lines must also have lengths defined for the voltage profile plot to work properly.

3.3. Available Inverter Control Types

There are currently five inverter controls available in the tool. This section will describe the objective and theory behind each of these controls. Further information about the performance of these controls can be found in [12].

3.3.1. Zero Current Injection

The zero-current injection (ZCI) control operates on the simple principle that the PV will never output more power than is consumed by its local load. By preventing any reverse power flow into the distribution network, no voltage rise should occur, and all over-voltages will be mitigated. This is a fair control if the PV are all sized equally proportional to the load to which they are connected, as is the case in the demonstration circuit. This control does not require a communication network.

3.3.2. Local Volt/Watt Control

If no communication network is available and PV are allowed to back-feed the network, then Volt/Watt control can be employed to curtail the PV based on the locally measured voltage. In this tool, this controller has been set to begin to act if $V_{PCC} > 1.044pu$ is measured at the PV. As shown below in Figure 7, the control will linearly curtail the PV output up for voltages up to $V_{PCC} = 1.05pu$, at which point the PV will be fully curtailed. This control is not fair since different PV will have different voltages based on their location in the circuit, causing them to curtail proportionally different amounts.

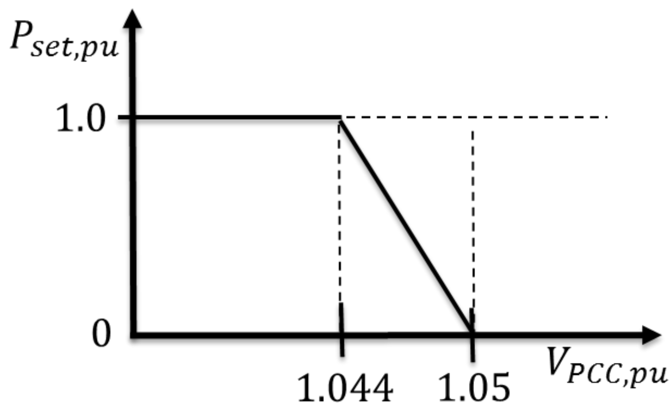

Figure 7. Curve used by local Volt/Watt inverter control to determine power curtailment based on locally measured voltage.

3.3.3. Local Volt/Var Control

Again assuming no communication, this control type demonstrates how the spare reactive power capacity of the PV can be used to mitigate over-voltages. For conditions like the example circuit where the inverter rating matches the PV DC rating, this control type will only work for irradiance levels, I, less than the rated irradiance $I_R = 1000W/m^2$ so that vars will be available. The ratio of vars available to each PV's maximum power rating is then given in (1).

$$\frac{Q_{avail}}{P_{PMPP}} = \sqrt{1 - \left(\frac{I}{I_R}\right)^2} \quad (1)$$

Similar to the Volt/Watt control presented in Section 2.2, this control type also utilizes a curve to determine how much of the available reactive power to output, which is shown in Figure 8. The inverter outputs capacitive vars at low voltage and absorbs inductive vars a high voltage, saturating on both ends at the presently available reactive power limit. There is a deadband around the nominal desired voltage where the inverter does not output vars to prevent the controller from constantly operating once the PCC voltage has moved into an acceptable range. The voltage points that define the knees of the curve, $\boldsymbol{x}$, in Figure 8 are variable and in this tool they have been to $\boldsymbol{x} = [0.95\ 0.99\ 1.01\ 1.05]$ to reflect the ANSI Range-A standards.

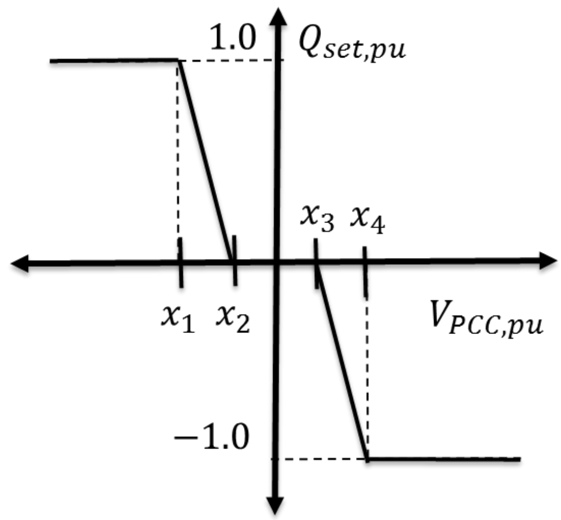

Figure 8. Curve used by local Volt/Var inverter control to determine the proportion of available reactive power to output based on locally measured voltage.

3.3.4. Centralized Fair Dispatch

This control type and the next both assume there exists a bidirectional communication system on the feeder that can transmit the voltage and power of all PV systems and dispatch power set-points, hence the "centralized" designation. The objective of this first approach is to fairly distribute the curtailment of PV power until the maximum circuit voltage is $V_{max} < 1.05$. The control achieves this by iteratively ramping down the power outputs of all PV uniformly until the voltage constraint is achieved.

3.3.5. Centralized Sensitivity-Based Dispatch

This last control approach uses network knowledge to selectively change the individual PV power outputs to mitigate network over-voltage violations with the minimum amount of curtailment possible. It achieves this by constructing a sensitivity matrix as shown in (2). Each column of the matrix corresponds to a PV system and is constructed by first reducing the output of that PV by an amount, Δp, equal to the size of the smallest PV system on the feeder. Then the corresponding change in all PV voltages from the baseline values with all PV systems at maximum output is stored in that column.

$$\begin{gathered} \boldsymbol{A} = [\boldsymbol{a_1}\ \boldsymbol{a_2} \dots \boldsymbol{a_j} \dots \boldsymbol{a_n}] \\ \boldsymbol{a_j} = \boldsymbol{V_j} - \boldsymbol{V_0}, \qquad s.t.\, p_j = p_{j,0} - \Delta p \end{gathered} \tag{2}$$

The amount to curtail each PV is determined by multiplying the inverse of the sensitivity matrix, $\boldsymbol{A}$, with the voltage deviation from $1.05pu$. Since this is a linear approximation to a nonlinear problem, a small scalar gain is applied to the power dispatch signal and iteratively increased until the maximum network voltage is less than $1.05pu$. By the objective of this control, it unfairly curtails the systems that will reduce the overall network voltage the greatest amount per kW reduction in their output.

3.4. GUI Functionality

This section will instruct the user how to operate of the GUI. The GUI layout is shown below in Figure 9. Each element of the GUI is highlighted with a red box and is numbered corresponding to the subsection that describes its functionality.

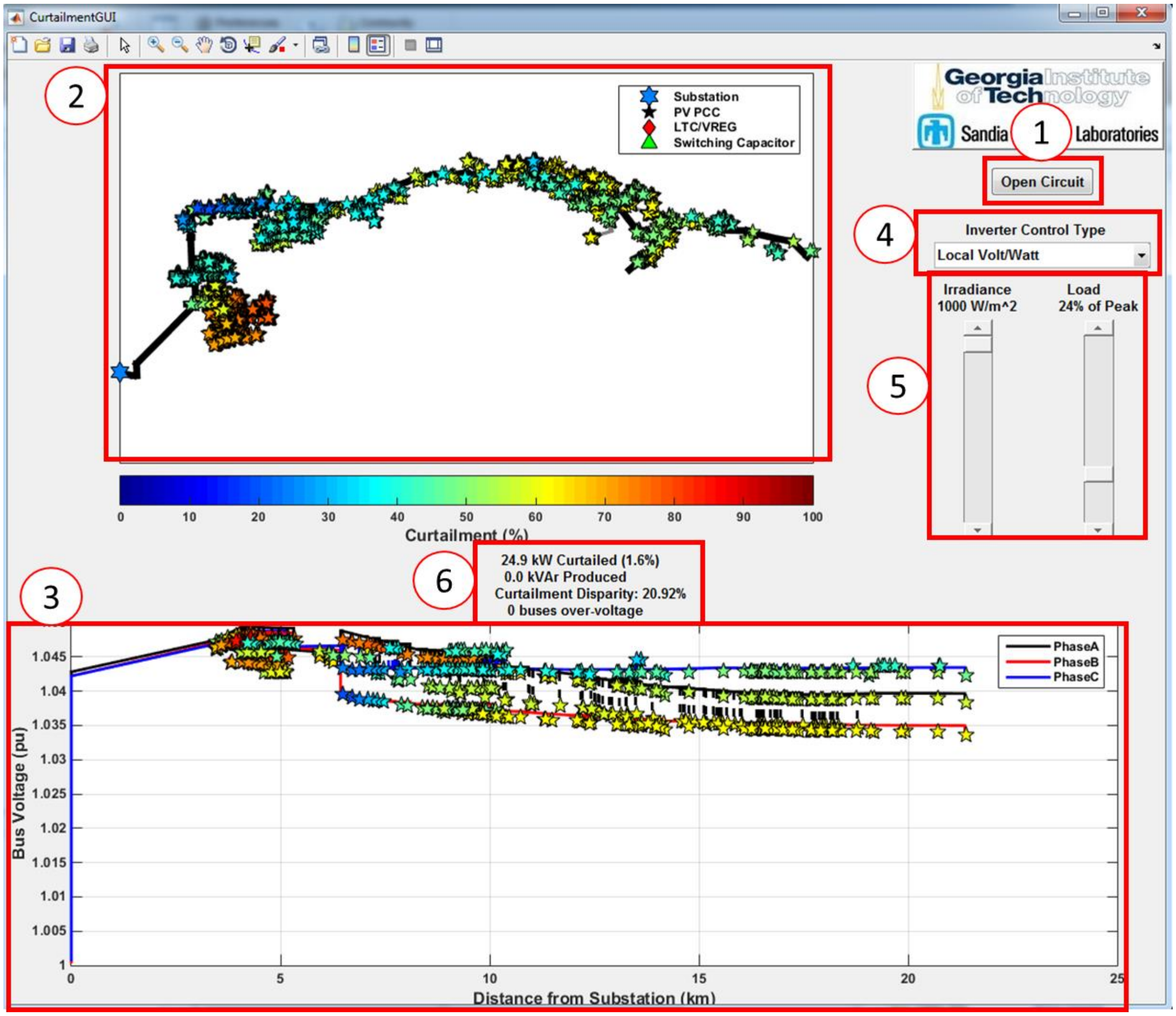

Figure 9. Layout of inverter control simulation tool GUI.

3.4.1. Load Circuit File

The upper right corner of the GUI has a button labeled “Open Circuit”. Pressing this button opens up a window prompting the user to select the Master.dss file of their OpenDSS circuit. The file name does not actually matter, but the user should select the main file of the circuit they wish to test. The circuit must be formatted as described in Section 1.3

3.4.2. Circuit Curtailment Plot

The large plot on the top displays the geographical layout of the feeder topology. The feeder model must have coordinates defined for its buses for this plot to be possible. Each black line in this plot represents a medium-voltage line in the feeder and each grey line represents a low-voltage line. The feeder substation, voltage regulators, and switching capacitors are represented by the symbols shown in the legend. However, depending on the number of PV in the feeder, these symbols may be eclipsed by the PV markers, represented as a star at each PV placement location. The PV markers are colored by the percent real power being curtailed based on the control type selected, irradiance, and loading level.

3.4.3. Voltage Profile Plot

The large plot on the bottom of the GUI displays the voltage profile of each of the feeder's three phases as a function of distance from the feeder. The bold solid lines represent the medium voltage lines and the dashed lines represent the step down in per-unit voltage that occurs across the service transformers and short secondary lines to the loads. The location of the PV units along the voltage profile are again indicated by stars colored by the amount of real power actively being curtailed.

3.4.4. Control Type Selection

The user can use this drop-down menu to select a control type from the ones listed in Section 2. Selecting a new control type will re-solve the circuit with that control applied to all PV inverters.

3.4.5. Load and Irradiance Selection

The user can uniformly scale all loads and the irradiance seen by all PV with the sliders on the right side of the GUI. Depending on the size and number of PV systems on the network, the load may need to be scaled significantly lower than the PV before any over-voltages occur. With no over-voltage violations, the control may not do anything.

3.4.6. Result Totals

The total number of over-voltage violations, PV power curtailed, and PV vars generated are summed and presented in this box in the center of the GUI. Also presented is a "fairness" metric indicating how equally the PV share curtailment proportionally. This metric is the standard deviation of the distribution of percent curtailment among all PV. Together, these four values are the overall results for the snapshot in time with that particular control type.

3.4.7. Operational Notes

An OpenDSS file must be loaded using the "Open Circuit" button before any other operations are attempted or there will be a Matlab error. To exit the GUI, simply hit the red 'x' in the upper-right of the window.

3.5. Summary

This tool provides a quick simulation of the impact of various advanced PV inverter control strategies on realistic distribution networks at varying levels of load and irradiance. Several control strategies are available to be applied uniformly to all PVs in a user-defined distribution feeder model. The strategies have different objectives and assumptions that yield results that differ in their effectiveness, efficiency, and fairness to all PV systems. The tool gives the user insight on the location of critical PV systems in their network and it presents an objective comparison between control strategies that may be incorporated into further decision-making processes about how best to control their distributed PV systems.

4. ENERGY STORAGE FOR PV RAMP RATE SMOOTHING TOOL

This chapter provides a user manual for the energy storage for PV ramp rate smoothing MATLAB tool. The tool allows the user to select different storage characteristics (power and energy ratings) and control types (local vs. centralized). The gains in the feedback control are also modifiable by the user to study the tradeoffs between state-of-charge (SOC) management and the amount of ramp rate smoothing. The tool includes example PV output data for a 1-week period. The simulation results can also be played in a video animation format through the simulation period. The purpose of the GUI is to demonstrate the effectiveness of energy storage in mitigating PV variability [13]. The interactive nature of the GUI allows the user to investigate different control types and options.

4.1. Files

The MATLAB script and function files that the GUI uses are listed in Table 3.

Table 3: MATLAB scripts and functions that the GUI uses

Name	Functionality
batterySmoothingDisplay2.m	MATLAB code for the GUI, animations, and storage controls
batterySmoothingDisplay2.fig	MATLAB figure file with the information about the layout of the GUI
PVdata.mat	Saved high-resolution PV output data

4.2. Sample Data

The energy storage ramp rate smoothing tool includes 1-week of high resolution 1-second PV output data. The data is stored in PVdata.mat, and it is required that this file is located in the same folder as the GUI tool. Irradiance measurements at 1-second resolution from an array of 7 irradiance sensors in San Diego, California [14], were used to generate 91 unique PV power output timeseries profiles, one for each medium-voltage interconnection point (service transformer). For situations with multiple customers connected to a single transformer, each customer was assigned the same irradiance profile. In all, 306 PV systems totaling 2.8 MW were simulated. The 91 power profiles were created by first pairing each service transformer with the irradiance sensor network based on their relative latitude. The color coding in Figure 10 left shows the irradiance sensor (open circle) assigned to each transformer (solid dot). Then, the irradiance was time-shifted based on the distance between the (assumed) location of the irradiance sensor and the transformer. The time shift was calculated as the distance divided by the cloud speed, assuming clouds propagate from west to east. Based on a year of cloud speeds at the feeder location, the maximum speed of 24 m/s was simulated in order to demonstrate the worst case PV variability. The resulting time offsets are shown in Figure 10 right.

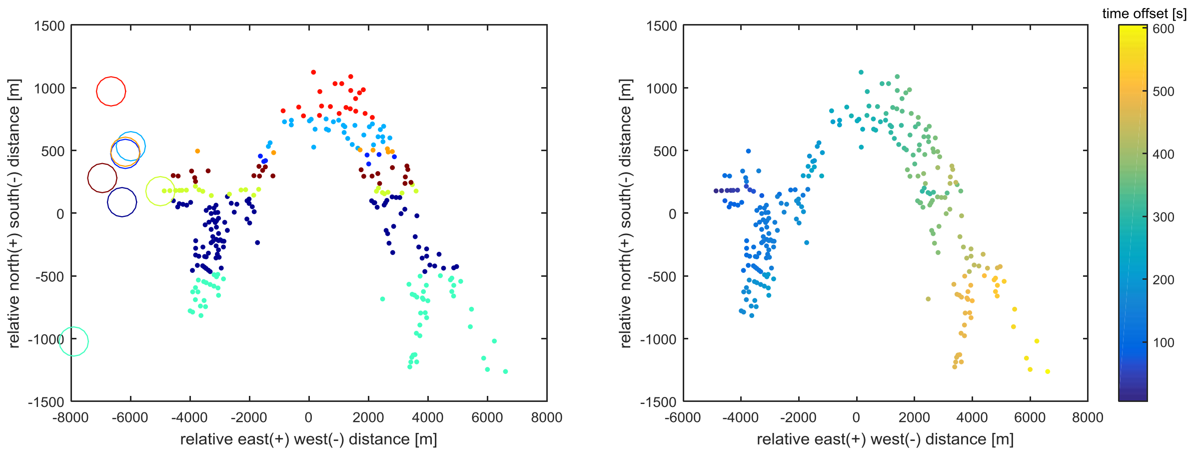

Figure 10. PV system pairing with irradiance sensors (left) and time offset for 24 m/s cloud speed (right).

For each transformer, the time-shifted measured irradiance was converted to latitude tilt plane of array (POA) irradiance using the Erbs decomposition and Hay/Davies transposition models. The Sandia Array Performance Model and Sandia Inverter models were used to obtain PV power output from the POA irradiance. When repeated for each transformer, this produced this 91 unique power profiles contained in PVdata.mat.

4.3. GUI Functionality

The tool is implemented as a MATLAB graphical user interface. The GUI can be executed in MATLAB by typing the command "batterySmoothingDisplay2" or by double clicking the icon batterySmoothingDisplay2.fig in the "Current Folder" in MATLAB. The main window of the GUI is shown in Figure 11.

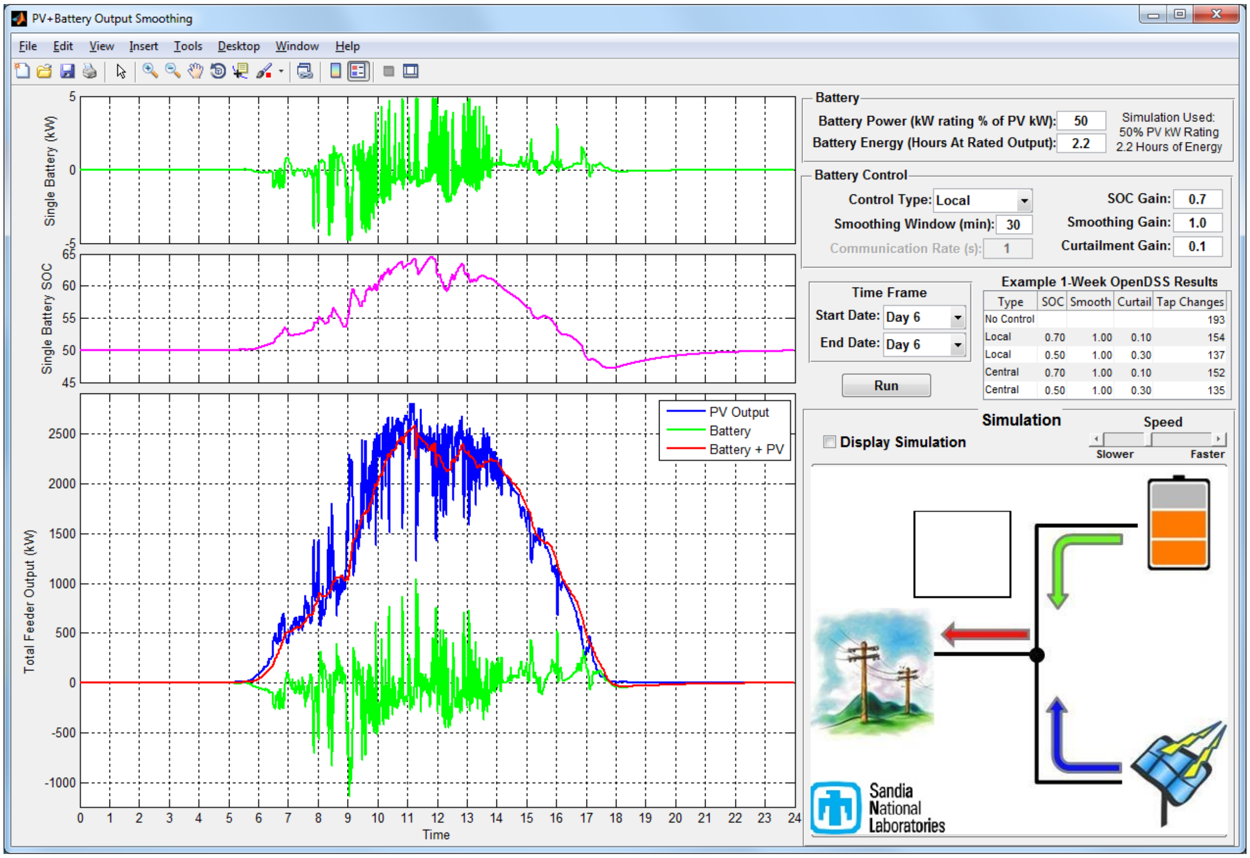

Figure 11. GUI in MATLAB

4.3.1. Battery Parameters

In the top right corner of the GUI are the user modifiable parameters for the battery size. It is assumed that each of the 306 PV systems includes a local energy storage device for ramp rate smoothing. The PV sizes are proportional to each customer's peak load. The battery power ratings (kW) are assumed to also be proportional to the customer's PV size. The battery size (kWh of energy) is quantified by the number of hours it can output full rated power. The basecase simulation in OpenDSS used 50% PV kW rating and 2.2 hours of energy. The battery control must operate inside the parameters of the battery, so independent of the control logic, the battery cannot output more power than its kW rating or charge/discharge outside of the potential energy storage capabilities.

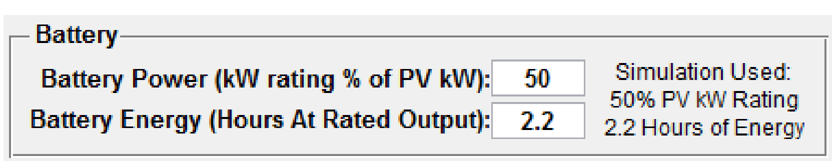

Figure 12. Battery parameters window.

4.3.2. Battery Control

Below the battery parameters are the battery controls. The controls can be done locally or using centralized dispatch. The local control does ramp rate smoothing assuming that it only knows the local PV output. The centralized control communicates with all PV/battery systems in order to know the aggregate PV output and dispatch the individual storage devices based on the PV ramps and battery's SOC. For the centralized control, the communication rate is a user input. For example, if the communication rate is 60 seconds, then the central controller will only be updated with the PV systems' power output and batteries' SOC every 60 seconds. The dispatch signal to control the batteries will also happen at the same communication update rate.

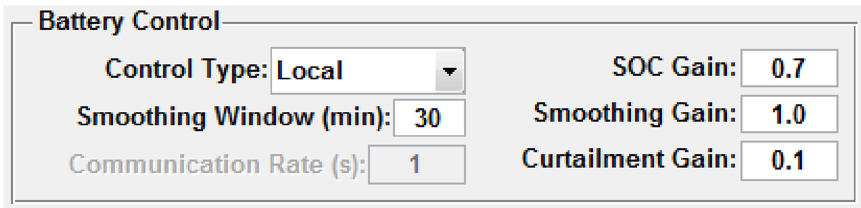

Figure 13. Battery control window.

The battery control diagram is shown in Figure 14. It is based on the PV output smoothing work in [15]. G_1, G_2, and G_3 are the SOC gain, smoothing gain, and curtailment gain, respectively. T_W is the length of the moving average time window. SOC_{REF} is the target SOC reference point, generally around 0.5 to keep the battery available for smoothing both up and down ramps in PV output. For local ramp rate smoothing control, the PV inverter power output block comes directly for the PV system co-located with the energy storage device. For the centralized control, the PV block represents the summation of all PV generation on the feeder at the time of the communication update.

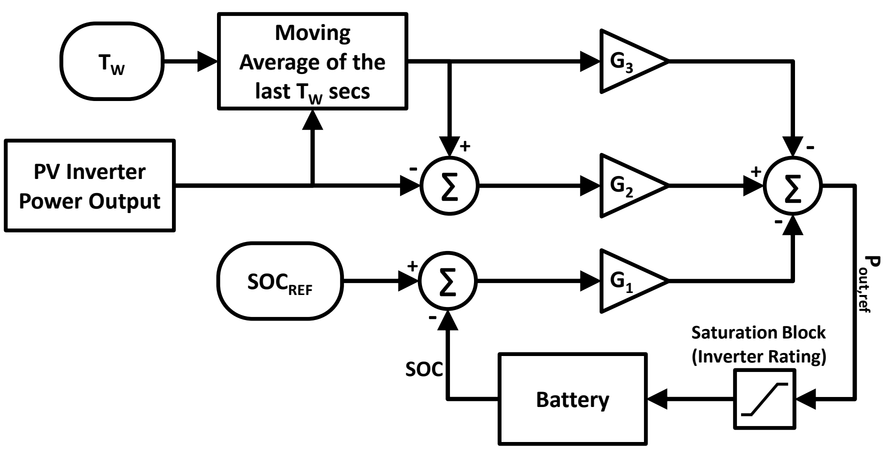

Figure 14. Battery control diagram.

4.3.3. Time Frame

The time frame window selects which days to simulate. Since there is a week of sample data, up to 7 days can be simulated. Any range of days (1 to 7) can be simulated, and which days are included can be modified by changing the drop down menus. The demo GUI tool only includes data for Day 6 in order to decrease the file size for the ability to email an example.

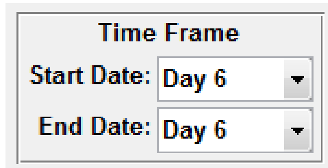

Figure 15. Time frame window.

4.3.4. Figures

The plots display the results for the time frame that was simulated. The plotting tools (such as the zoom tool) at the top of the figure can be used to interact with the plots. The top two plots show the results for a single PV system: power output (kW) and SOC, respectively. In all plots the power notation is out of the device, with positive being production. The third plot shows the aggregate output from all PV systems and batteries on the feeder. The net output (battery + PV) is much less variable due to the ramp rate smoothing provided by the battery.

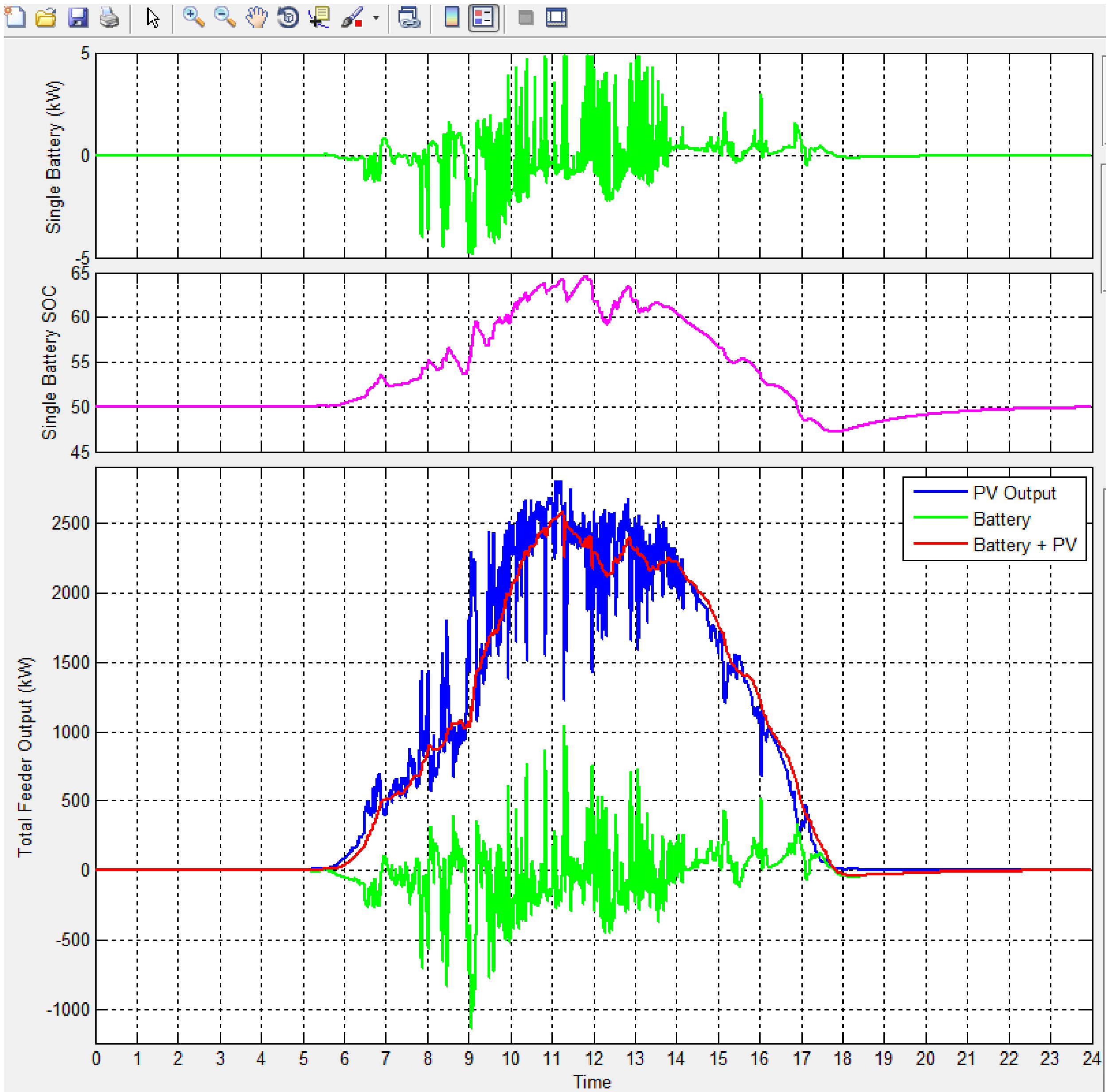

Figure 16. Figure and plotting window.

4.3.5. Simulation Animation

The simulation can be animated by selecting the "Display Simulation" checkbox. When the box is checked, the simulation will run while animating the figure plots (Section 4.3.4). The animation diagram in the bottom right will also update as the animation plays. The arrows demonstrate the direction of power flow for the PV system (blue), battery (green), and net power (red). The magnitude of each of the power flows is similarly demonstrated by the bar graph in the animation. The battery SOC is displayed inside the battery icon by the level of the blue fill. The animation speed can be adjusted by moving the speed slider. The animation can also be stopped at any time by unchecking the "Display Simulation" box.

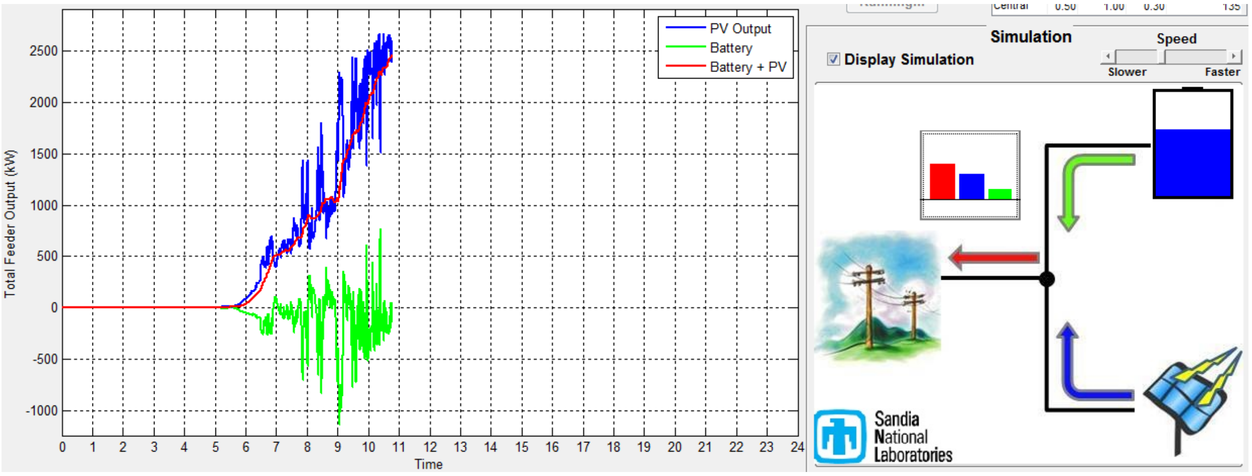

Figure 17. Simulation animation window.

4.4. Summary

A MATLAB GUI for demonstrating energy storage for PV ramp rate smoothing has been created. The tool allows the user to select different storage characteristics (power and energy ratings) and control types (local vs. centralized). The gains in the feedback control are also modifiable by the user to study the tradeoffs between state-of-charge (SOC) management and the amount of ramp rate smoothing. The interactive nature of the GUI allows the user to investigate different control types and options. The GUI demonstrates the effectiveness of energy storage in mitigating PV variability. The centralized control communicates with all PV systems and batteries on the feeder. With this increased visibility, the centralized control can mitigate solar variability with less energy storage and less wear and tear on the batteries. The centralized control is limited by the communication infrastructure and the update rate. For slower update rates of greater than 15-seconds, the advantages of the centralized controller begin to disappear.

REFERENCES

[1] R. J. Broderick, J. E. Quiroz, M. J. Reno, A. Ellis, J. Smith, and R. Dugan, "Time Series Power Flow Analysis for Distribution Connected PV Generation," Sandia National Laboratories SAND2013-0537, 2013.

[2] M. J. Reno, K. Coogan, S. Grijalva, R. J. Broderick, and J. E. Quiroz, "PV Interconnection Risk Analysis through Distribution System Impact Signatures and Feeder Zones," in *IEEE PES General Meeting*, 2014.

[3] J. Peppanen, J. Grimaldo, M. J. Reno, S. Grijalva, and R. Harley, "Modeling of Distribution Systems with Extensive Deployment of Smart Meters," *IEEE PES General Meeting*, 2014.

[4] J. Peppanen, M. J. Reno, M. Thakkar, S. Grijalva, and R. G. Harley, "Leveraging AMI Data for Distribution System Model Calibration and Situational Awareness," *IEEE Transactions on Smart Grid*, 2015.

[5] J. Peppanen, M. J. Reno, R. J. Broderick, and S. Grijalva, "Distribution System Secondary Circuit Parameter Estimation for Model Calibration," Sandia National Laboratories SAND2015-7477, 2015.

[6] EPRI,"*Open Distribution System Simulator (OpenDSS)"*, Available: http://sourceforge.net/projects/electricdss/

[7] M. J. Reno and K. Coogan, "Grid Integrated Distributed PV (GridPV)," Sandia National Labs SAND2013-6733, 2013.

[8] M. J. Reno and K. Coogan, "Grid Integrated Distributed PV (GridPV) Version 2," Sandia National Labs SAND2014-20141, 2014.

[9] Pecan Street Inc., "*Pecan Street Inc."*, Available: http://www.pecanstreet.org/about/

[10] J. Seuss, M. J. Reno, R. J. Broderick, and S. Grijalva, "Improving Distribution Network PV Hosting Capacity via Smart Inverter Reactive Power Support," in *IEEE PES General Meeting*, Denver, CO, 2015.

[11] J. Seuss, M. J. Reno, R. J. Broderick, and R. G. Harley, "Evaluation of Reactive Power Control Capabilities of Residential PV in an Unbalanced Distribution Feeder," in *IEEE Photovoltaic Specialists Conference*, 2014.

[12] J. Seuss, M. J. Reno, M. Lave, R. J. Broderick, and S. Grijalva, "Multi-Objective Advanced Inverter Controls to Dispatch the Real and Reactive Power of Many Distributed PV Systems," Sandia National Laboratories SAND2015, 2015.

[13] M. Lave, M. J. Reno, and R. J. Broderick, "Characterizing Local High-Frequency Solar Variability and the Impact to Distribution Studies," *Solar Energy*, 2015.

[14] M. Lave, J. Kleissl, and E. Arias-Castro, "High-frequency irradiance fluctuations and geographic smoothing," *Solar Energy*, vol. 86, 2012.

[15] A. Ellis and D. Schoenwald, "PV Output Smoothing with Energy Storage " Sandia National Laboratories SAND2012-1772, 2012.

5. DISTRIBUTION

1	MS1033	Robert J. Broderick	6112
1	MS1033	Abraham Ellis	6112
1	MS1140	Matthew J. Reno	6113
1	MS1140	Ross Guttromson	6113
1	MS0899	Technical Library	9536 (electronic copy)

Sandia National Laboratories