SANDIA REPORT
SAND2021-2660
Printed March 2021

Sandia National Laboratories

Rapid QSTS Simulations for High-Resolution Comprehensive Assessment of Distributed PV

Robert J. Broderick, Matthew J. Reno, Matthew Lave, Joseph A. Azzolini, Logan Blakely, Jason Galtieri
Sandia National Laboratories

Barry Mather, Andrew Weekley, Randolph Hunsberger, Manohar Chamana, Qinmiao Li, Wenqi Zhang, Aadil Latif, Xiangqi Zhu
National Renewable Energy Laboratory

Santiago Grijalva, Xiaochen Zhang, Jeremiah Deboever, Muhammad Umer Qureshi
Georgia Institute of Technology

Francis Therrien, Jean-Sebastien Lacroix, Feng Li, Marc Belletête, Guillaume Hébert
CYME International T&D

Davis Montenegro, Roger Dugan
Electric Power Research Institute

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

Issued by Sandia National Laboratories, operated for the United States Department of Energy by National Technology & Engineering Solutions of Sandia, LLC.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

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@osti.gov

Online ordering: http://www.osti.gov/scitech

Available to the public from

U.S. Department of Commerce

National Technical Information Service

5301 Shawnee Rd

Alexandria, VA 22312

Telephone: (800) 553-6847

Facsimile: (703) 605-6900

E-Mail: orders@ntis.gov

Online order: https://classic.ntis.gov/help/order-methods/

ABSTRACT

The rapid increase in penetration of distributed energy resources on the electric power distribution system has created a need for more comprehensive interconnection modeling and impact analysis. Unlike conventional scenario-based studies, quasi-static time-series (QSTS) simulations can realistically model time-dependent voltage controllers and the diversity of potential impacts that can occur at different times of year. However, to accurately model a distribution system with all its controllable devices, a yearlong simulation at 1-second resolution is often required, which could take conventional computers a computational time of 10 to 120 hours when an actual unbalanced distribution feeder is modeled. This computational burden is a clear limitation to the adoption of QSTS simulations in interconnection studies and for determining optimal control solutions for utility operations. The solutions we developed include accurate and computationally efficient QSTS methods that could be implemented in existing open-source and commercial software used by utilities and the development of methods to create high-resolution proxy data sets. This project demonstrated multiple pathways for speeding up the QSTS computation using new and innovative methods for advanced time-series analysis, faster power flow solvers, parallel processing of power flow solutions and circuit reduction. The target performance level for this project was achieved with year-long high-resolution time series solutions **run in less than 5 minutes** within an acceptable error.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under Solar Energy Technologies Office (SETO) Agreement Number 30691.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

This page left blank

EXECUTIVE SUMMARY

The rapid increase in penetration of distributed energy resources on the electric power distribution system has created a need for more comprehensive interconnection modeling and impact analysis. Unlike conventional scenario-based studies, quasi-static time-series (QSTS) simulations can realistically model time-dependent voltage controllers and the diversity of potential impacts that can occur at different times of the year. Traditionally, distribution system analysis and planning tasks have relied on snapshot analyses of critical periods like the peak load time during the year. For radial distribution systems with current flowing unidirectionally from the transmission system through the substation to the load, analyzing the peak load period captured the most important aspects of voltage drop and thermal loading on distribution system equipment. With the integration of distributed energy resources and smart grid devices and controls at the grid-edge, distribution systems can experience reverse power flows, voltage problems and increased variability. This rapid change in distributed energy resource penetration has led utilities and researchers to perform more time-series analyses to understand the changes and impacts to distribution systems throughout the year. Since PV power output can be highly variable during cloud transients which causes significant fluctuations from one second to the next, the impact of this variability on other aspects of the distribution system like voltage regulation devices must be understood Modeling and simulations of the impact of solar PV and other distributed resources on various elements of the distribution feeder is imperative for their cost-effective integration.

Quasi-static time series (QSTS) simulations are a very effective tool to identify and mitigate the impacts associated with the integration of distributed energy resources. QSTS are high-resolution time-series simulations, generally with time-steps on the order of seconds, where each power flow is dependent on the previous solution. These characteristics enable QSTS simulations to realistically model time-dependent voltage controllers and the diversity of potential impacts that can occur throughout the year. However, the adoption of QSTS simulations has been limited by the computational time requirement, which can be on the order of several days of simulation time for a single unbalanced distribution feeder; to solve a 1-second resolution QSTS simulation with a yearlong time-horizon, 31.5 million power flow solutions are required. This significant computational burden and the lack of measured PV power inputs for such simulations are major challenges to their widespread adoption.

This report highlights novel solutions for improving the speed of QSTS and addressing the input data requirements to enable the widespread adoption of QSTS simulation in distribution system planning. The solutions we developed include accurate and computationally efficient QSTS methods that could be implemented in existing open-source and commercial software used by utilities and methods to create high-resolution proxy data sets.

First, several aspects regarding the need for QSTS simulations are discussed, including how these simulations can be used to assess the potential impacts of PV on the distribution system in a more comprehensive manner. The discussions include analyses on the various requirements for the input data to QSTS simulations and the standards for performing QSTS simulations. It was observed that QSTS simulation standards varied based on the application, but in general, the simulations should be performed using at least a 5-second time-step resolution and a yearlong time-horizon.

The second section of the report discusses the challenges in reducing the computational time of QSTS simulations and presents multiple unique approaches for speeding up QSTS computation using new and innovative methods for advanced time-series analysis, faster power flow solvers, parallel processing of power flow solutions, and circuit reduction. Overall, 8 methods were

developed for improving the computational time of the QSTS simulation. To demonstrate the speed improvement and accuracy of each method, five test circuits were developed with a range of complexities in size, number of buses, voltage regulation equipment, and amount of input data required. The rapid QSTS algorithms were evaluated for scalability, accuracy, and robustness on standard IEEE test circuits and real utility-scale distribution feeders with their low-voltage secondary networks modeled. There is also a discussion about combining multiple rapid QSTS algorithms for additional speed improvements.

The third section of the report discusses methods to more easily and accurately create the input data required for QSTS simulations that represent the high-frequency inputs for all load and PV nodes on a feeder. The challenge addressed here is that QSTS simulations require additional information about the temporal variations of customer load and PV generation throughout the year, but load is often measured at low temporal resolution (and usually only at the feeder head), and PV production is often not measured at all by the utility. Thus, to ensure accurate QSTS studies, there is a need for accurate load and PV data generation with high spatial and temporal resolution. PV modeling methods were presented for (a) creating a simulated high-frequency irradiance time-series based solely on low-frequency, readily available satellite-derived irradiance, and (b) producing high-frequency PV inputs at all simulated PV locations on a distribution feeder based on high-frequency simulated or measured irradiance time-series. With or without additional measurements, the methods presented here enabled load and PV inputs with representative variability to be generated, allowing for significantly more accurate QSTS simulations than those using simple load and PV assumptions.

Lastly, the report discusses how the rapid QSTS algorithms were implemented into three main software environments. The first is an open-source release of the algorithms in MATLAB, allowing other researchers to easily test and modify the code for any application. The second is the implementation of the algorithms into OpenDSS. The OpenDSS integration allows the academic and research communities that already use OpenDSS to immediately see the benefits of faster QSTS simulations without having to run any external code. The third is implementation into CYME, which enables utilities to have access to more powerful algorithms in the software environment that they are used to.

Overall, the work highlighted in this project report describes rapid QSTS algorithms that drastically reduced the computational time of quasi-static time-series simulations and presents methods for simulating PV power inputs from readily available data. The rapid QSTS algorithms are thus capable of accurately modeling solar PV systems as well as capturing the operation of voltage-regulating equipment (e.g., number of tap position changes over time).

In the coming years, QSTS simulations will become an integral part of distribution system analysis and planning tasks for integrating solar PV and other distributed resources. The significant computational time reduction and the novel methods for generating accurate PV power output profiles mean that electric utilities can more efficiently and accurately utilize QSTS simulations to evaluate PV interconnection requests. Beyond PV integration, the ability to perform fast and accurate QSTS simulations will benefit a variety of distribution system analysis and planning tasks, such as analyzing demand response controls, conservation voltage reduction, hosting capacity analyses, and feeder reconfiguration strategies.

ACRONYMS AND DEFINITIONS

Abbreviation	Definition
ADMS	advanced distribution management system
AMI	advanced metering infrastructure
ANSI	American National Standards Institute
BOI	Bus of Interest
BTVTS	Back track variable time step
CB	Critical bus
CPU	Central Processing Unit
CR	Circuit reduction
CYME	Company that develops the CYMDIST power system analysis tool
DER	Distributed Energy Resources
DG	Distributed Generation
DLL	Dynamic link library
DR	demand response
DOE	Department of Energy
DWT	Discrete Wavelet Transforms
EB	Event-based
EMT	Electromagnetic transient
EPRI	Electric Power Research Institute
ESS	energy storage system
EV	electric vehicle
GHI	global horizontal irradiance
GUI	graphical user interface
HC	Hosting capacity
HRIA	High resolution irradiance algorithm
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/output
LDC	Line Drop Compensation
LFwP	Load Flows with Profiles
LTC	Load Tap Changer
LTD	Long-term dynamic
LVR	Line voltage reguator
MAE	Mean absolute error
MATLAB	Matrix Laboratory- is a multi-paradigm numerical computing environment

Abbreviation	Definition
NREL	National Renewable Energy Laboratory
NRU	Newton Raphson Unbalanced
NSRDB	National Solar Radiation Database
OpenDSS	Open Distribution System Simulator™
OpenDSS-PM	Open Distribution System Simulator- Parallel Machine
OPF	Optimal power flow
PDVTS	Predetermined variable time step
PF	Power Factor
POA	plane of array
pu	per unit
PV	photovoltaic
QSTS	Quasi-Static Time-Series
RMS	root mean squared
SIND	Solar Integration National Database
SOC	state of charge
SNL	Sandia National Laboratories
SRRL	Solar Radiation Research Laboratory
SS	Steady State
TWP	Technical Work Plan
VDU	Voltage drop unbalanced
VQ	Vector Quantization
VREG	Voltage Regulator
VS	Variability score
VTS	Variable Time Step
VTS-BD	Variable Time Step with Back Tracking
WVM	Wavelet-based Variability Model
Xfmr	transformer
ZCI	zero current injection

1. INTRODUCTION

Without time-series analysis, many potential impacts of new DER, like the duration of time with voltage violations and the increase in voltage regulator operations, cannot be accurately analyzed. Understanding the voltage regulator operations is essential to determine the impact of DER on these expensive pieces of utility equipment. Furthermore, snapshot study methods that only analyze peak periods or a peak variability day often lead to over-estimation of normal operating issues. Paired with accurate load and generation time-series data or models, a QSTS can accurately quantify the magnitude, frequency, and duration of an impact.

QSTS involves steady-state power flows, but it is distinguished by the dependence on the previous power flow solution. Each time step of the solution cannot be solved independently because it relies on the information from the previous time step about the feeder state, regulator taps, control delays, etc. QSTS simulations specifically model these discrete controls and run the simulation as a time-series to capture the time-dependent states of any controllable elements. QSTS helps to understand the impact of a new DER and offers many practical advantages and uses over conventional tools:

This Office of Energy Efficiency and Renewable Energy systems integration project focuses on quasi-static time-series (QSTS) analysis to improve grid performance and reliability. The specific target for this project was to enable the SunShot target of greater than 100% of peak load for solar penetration on the distribution system by developing tools and methods for detailed and accurate analysis of distributed energy resource (DER) grid impacts. The use of QSTS analysis by the utilities has been limited by the available simulation software modeling capabilities, the lack of available input data, and the computational intensity of the QSTS analysis. The lack of fast and accurate time-series analysis results in many potential impacts not being studied, such as voltage regulator operations when variable generation sources are connected to the distribution system. By improving the computational speed of QSTS, longer time periods can be studied to provide better estimates of impacts than conservative snapshot or short time period estimates.

The specific challenges we addressed in this project are: data needs and computational burden to perform QSTS analysis. As seen in Table 1, existing QSTS algorithms back in 2015 could take anywhere from 10 to 120 hours to perform a 1- year simulation at 1-second resolution. The wide range in simulation times is due to the variety in circuit complexity, such as the number of buses and the number of voltage regulators. With the existing methods in 2015, even performing a QSTS simulation for a single day can take up to 20 minutes.

Table 1. Computation Times for 1-second Resolution QSTS

	Simulation Duration		
	1 Day	1 Month	1 Year
Existing Methods	1.6 – 20 minutes	0.8 - 10 hours	10 - 120 hours
Proposed Algorithm Target	3 minutes	4 minutes	5 minutes

The solution includes developing accurate and computationally efficient QSTS methods that could be implemented in existing open-source and commercial software used by utilities and the development of methods to create high-resolution proxy data sets. This project demonstrated multiple pathways for speeding up the QSTS computation using new and innovative methods for advanced time-series analysis, faster power flow solvers, parallel processing of power flow solutions,

and circuit reduction. The target performance level for this project was achieved with year-long high-resolution time series solutions, **run in less than five minutes** within an acceptable error.

1.1. Background

When this project started in 2015, QSTS analysis techniques and methods for the development of the appropriate data for PV integration studies had been presented regularly [1] since 2010. Many of these efforts by the principal investigators in this project [2-10] had shown the efficacy of QSTS analysis in determining PV impacts over days, months, or even a whole year with varying time steps from hourly to one second. However, the QSTS analysis techniques required a burdensome amount of high-resolution load and PV resource data and require long computational run-times. These QSTS analysis limitations, when applied to PV integration studies, impeded QSTS's acceptance and implementation in commercial distribution system analysis software solutions.

Research that compares different methods for shortening time-series power flow simulations based on reducing the amount of input data and thus the required number of individual power flow calculations was done in [11]. The author focuses on reducing the time resolution of the input profiles through down sampling and on finding similar time-steps in the input profiles through vector quantization. This project extended this proof-of-concept work to include more detailed distribution system models with voltage regulators and higher resolution data inputs.

In 2015, some commercial vendors had recently implemented basic QSTS functionality into their software packages, but such implementations are inappropriate for use in accurate PV integration studies due to the limited length of time analyzed or the time step of the analysis itself. The QSTS functionality of tools in 2015 needed to be dramatically improved to solve in a timely manner with more accurate data that represents load and solar variability. One of the key challenges that had not been demonstrated in 2015, was the development of QSTS methods and tools that could maintain power flow analysis accuracy while improving speed and being scalable to any type of distribution system.

This project squarely addresses the two major QSTS analysis limitations for PV integration analysis identified above, by developing advanced QSTS methods that greatly decrease the required computational time and burden and by developing high-resolution proxy data sets for QSTS PV interconnection simulations.

1.2. Project Objectives

The objectives of this project are to accelerate QSTS simulation capabilities through the use of new and innovative methods for advanced time-series analysis, such as variable time step, vector quantization, event-based, intelligent sample selection, machine learning, parallel processing of power flow solutions separable by time and space, and circuit reduction. Each of these methods contributed to speeding up the QSTS computation. This project demonstrates how to precisely simulate grid impacts while dramatically reducing the computational time required to solve the power flow time-series – making QSTS analysis increasingly a preferred PV impact assessment method.

The project goal was to reduce computational time (10-120 hours) for a complete yearlong QSTS power flow analysis to 5 minutes or less. The extended computational time for QSTS analysis makes it unusable by utilities that need an accurate yearlong analysis of the expected PV impacts. Additionally, the data inputs, inclusive of loads connected to a distribution circuit and power produced by interconnected PV systems, for QSTS analysis are burdensome for utilities because

utilities often do not collect and store the necessary data at the required time-resolution. The outcome of this project was the development of tools and methods implemented in both commercial and open-source power flow analysis software to allow QSTS to be performed while maintaining accuracy while improving speed and being scalable to any system size.

The impact of this project helped enable the SunShot target of greater than 100% of peak load for solar penetration on the distribution system through using more detailed and accurate methods for analyzing photovoltaic (PV) grid impacts and mitigation solutions. This supports the national goals of a clean energy progression. With these new improved time-series analysis and tools, many potential impacts, like the duration of time with voltage violations and the increase in voltage regulator operations, can now be accurately analyzed with the tools available to utilities. New study methods using the QSTS results of this project will allow for accurate and cost-effective estimation of PV-caused issues and thus enhance the adoption of PV and other DER particularly as penetration of DER increases.

The Technical Work Plan (TWP) for the entire project can be summarized by the activities defined in 6 major tasks:

- Fast Time-Series Approximations
- Improved Power Flow Solution Algorithms
- Circuit Reduction
- Parallelization of QSTS Analysis
- Implementation of Accelerated QSTS
- High-Resolution Input Data.

The first four tasks focused on improving the computational efficiency of QSTS calculations; the fifth task integrates and implements successful improvements into commercial and open-source software, and the last task addresses the need for high-resolution proxy data sets required for accurately representing PV system impacts.

2. NEED AND STANDARDS FOR QSTS

To examine the impact of PV on the distribution system, many different types of studies and analyses can be performed. Commercial circuit analysis tools have historically provided the capability to perform steady-state power flows to analyze the distribution system at specific snapshots in time, but PV output is highly variable and the potential interaction with control systems may not be adequately analyzed with traditional snapshot tools and methods [12, 13]. The challenge of running only peak periods or a peak variability day is to know how to extrapolate these results accurately to the full year of operations. Knowing the voltage regulator tap operations caused by solar and load variability is essential to determine the actual impact of DGPV on these crucial and expensive pieces of utility equipment. Paired with accurate load and generation time-series data or models, QSTS can accurately measure these effects.

This chapter discusses several aspects of QSTS simulations and how high time-resolution data can be used to assess the potential impacts of PV on the distribution system in a more comprehensive manner. The main advantage of using QSTS simulation is its capability to properly analyze and capture the time-dependent aspects of power flow [6, 14-16]. QSTS produces sequential steady-state power flow solutions where the converged state of each iteration is used as the beginning state of the next. Examples of the time-dependent aspects of power flow include the interaction between the daily changes in load and PV output and the impact on voltage regulation equipment. Another advantage of QSTS is the ability to quantify both the magnitude of an impact as well as the frequency and duration of the impact [17]. QSTS analyzes the power system by solving a series of steady-state power-flow solutions with a fast-enough time step to capture control actions with moderately slow, generally multi-second, response times. QSTS with time steps of 1–60 seconds is sufficient to capture the response of typical distribution equipment, including voltage regulators [6], and can be simulated for periods from hours to years.

2.1. Motivation and Need for QSTS

The proposed R&D activities will enable the SunShot targets of greater than 100% of peak load for solar penetration on the distribution system by using more detailed and accurate methods for analyzing photovoltaic (PV) grid impacts and mitigation solutions. Without improved time-series analysis and tools, many potential impacts, like the duration of time with voltage violations and the increase in voltage regulator operations, cannot be accurately analyzed with the tools currently used by utilities to model PV impacts. Furthermore, the currently used study methods often lead to over-estimation of PV-caused issues (which limits the adoption of PV) and are particularly ill-suited for the analysis of many multiples of PV connections on each distribution circuit, which will become common in the future as PV penetration increases.

2.1.1. Comparison to snapshot methods

Conventionally, distribution system analysis has focused on steady-state power flow simulations, harmonic analysis, and system protection studies. These types of studies have traditionally been sufficient for distribution system engineers' planning needs such as designing feeder layouts, planning expansions and upgrades, and determining control settings. However, emerging technologies and capabilities such as energy storage systems (ESS), electric vehicles (EVs), distributed photovoltaic (PV) advanced inverters, demand response (DR), and Advanced Distribution Management Systems (ADMS) are changing the paradigm for distribution system planning and operations. Commercial circuit analysis tools have provided the capability to perform steady-state power at snapshots in time, such as the peak load period that was historically the most

extreme condition. Traditional snapshot tools and methods may not be adequate to accurately analyze the interactions of new distributed energy resources (DER) being interconnected.

A draft of the IEEE guide on conducting DER distribution impact studies for distributed resource interconnection discusses four types of special system impact studies: (1) dynamic simulation, (2) electromagnetic transient (EMT) simulation, (3) harmonic and flicker study, and (4) quasi-static simulation [18]. Dynamic simulation is practical to simulate stability issues or voltage and frequency ride-through [18]. EMT simulation is often used for protection design and fault analysis [18, 19]. Harmonic and flicker studies provide insights into the feeder's power quality [18]. Last, quasi-static simulation is a versatile study used to understand equipment control operation, voltage regulation, and reactive power management [18]. In this work, we focus on the quasi-static time-series (QSTS) simulation to study the impact of various DERs. QSTS simulation is best defined by the IEEE draft guide:

"Quasi-static simulation refers to a sequence of steady-state power flow, conducted at a time step of no less than 1 second but that can use a time step of up to one hour. Discrete controls, such as capacitor switch controllers, transformer tap changers, automatic switches, and relays, may change their state from one step to the next. However, there is no numerical integration of differential equations between time steps." [18]

QSTS involves steady-state power flows, but it is distinguished by the dependence on the previous power flow solution. Each time step of the solution cannot be solved independently because it relies on the information from the previous time step about the feeder state, regulator taps, control delays, etc. QSTS simulations specifically model these discrete controls and run the simulation as a time-series to capture the time-dependent states of any controllable elements. QSTS helps to understand the impact of a new DER and offers many practical advantages and uses over conventional tools:

1) Analysis is not limited to specific time periods, such as peak load, which may no longer be the most critical times with high penetrations of DER
2) Enables the study of control algorithms, such as energy storage or ADMS control, and interactions between control equipment, such as between PV advanced inverter volt-var and distribution voltage regulators [20]
3) Simulates impacts like voltage fluctuations that are caused by variable resources such as distributed PV
4) Calculates the interaction between the daily changes in load and PV output and perform energy and loss evaluations over actual profiles of load and generation
5) Determines the steady-state voltage conditions for quickly changing circuit load and or generation
6) Calculates the time duration of extreme conditions, such as the number of hours a customer is expected to see an over-voltage condition or the amount of time a conductor or transformer is overloaded each year

Without time-series analysis, many potential impacts of new DER, like the duration of time with voltage violations and the increase in voltage regulator operations, cannot be accurately analyzed. Understanding the voltage regulator operations is essential to determine the impact of DER on these expensive pieces of utility equipment. Furthermore, snapshot study methods that only analyze peak periods or a peak variability day often lead to over-estimation of normal operating issues. Paired with accurate load and generation time-series data or models, a QSTS can accurately quantify the magnitude, frequency, and duration of an impact.

2.1.2. Previous applications of QSTS: Where/When QSTS is Used

The notion of time-series power flow simulations is discussed in the literature for impact studies of different DER: solar PV [2, 4-6, 10, 14, 16, 21-28], wind [29, 30], electrical vehicles [31, 32], ESS [33, 34], and different load models [35-38]. QSTS simulation is also used for impact studies of control schemes in different power equipment: smart inverters [16, 22, 25, 39-45], and voltage regulating devices [4, 46-49]. The common objective for using QSTS analysis is to capture the time-dependent effects and controller actions on a feeder. QSTS analysis can be used to perform various types of studies on a feeder, such as studying the impact of DERs or control schemes on the voltage quality [2, 4-6, 10, 14, 16, 21, 24-26, 32, 39, 47, 48, 50]. The sequential time-series simulation can determine the range of voltage magnitude as well as the duration of any voltage violations on the circuit. QSTS simulation is also used to study the operation of voltage regulating devices [2, 4, 5, 10, 21, 22, 24, 26, 32, 51] caused by large power flow fluctuation that certain DER creates. Other types of studies performed with QSTS analysis includes: equipment loading assessment [30, 33, 52], system losses [10, 23, 24, 31, 33, 48, 49], or power flow direction [21].

2.2. QSTS Accuracy Metrics

The most important aspect of running a QSTS simulation is to accurately produce a set of output metrics that will quantify the impacts of PV on a distribution feeder. A set of metrics was developed to ensure that all quantities of interest are captured, and their acceptable accuracy thresholds are shown in Table 2. These thresholds have been established in [53]. The motivation for these metrics comes from [54], where the authors present a detailed discussion on voltage issues arising in feeders with significant PV penetration.

The accuracy of the improved rapid QSTS algorithms is determined by comparing the below metrics to the brute-force 1-year at 1-second base case simulation. Each method must maintain a certain amount of accuracy for each metric, as shown with the accuracy thresholds. These accuracy thresholds are a working draft set of numbers that have been updated as we moved through the project, but they should not currently be seen as strict guidelines.

Table 2. Accuracy metrics for QSTS analysis.

	Result Metrics for QSTS	# of Values	Accuracy Threshold
1.	Highest/lowest voltage on the feeder	2	0.005 pu
2.	Highest/average/lowest voltage at each node	3 per node	0.005 pu
3.	Total number of tap changes per regulator	1 per regulator	10%
4.	Total number of switching operations per capacitor	1 per capacitor	20%
6.	Time duration that the feeder is above/below ANSI limits	2	24 hours
7.	Highest thermal percent loading per component	1 per line/xfmr	5%
8.	Duration of thermal overload per component	1 per line/xfmr	24 hours
9.	Summation of line losses for the feeder for the year	1	5%
10.	Feeder head max/average/min reactive power	3	100 kVAr
11.	VArs produced/consumed by each PV system for the year	-	5%
12.	Curtailment (kWh) with volt/watt control for each PV system for the year	-	5%

2.3. Standards for QSTS

Many motivations and applications for QSTS are discussed in the previous section showing the efficacy of QSTS analysis in determining PV impacts over days, months, or even a whole year with varying time steps from hourly to 1 second [4, 5, 7-10, 55-57]. However, there is very little information

available for the requirements and standards for performing QSTS simulations. This section of the report provides a novel analysis of the QSTS requirements for the input data time-resolution, the simulation time-step resolution, and the length of the simulation. The requirements are application-specific to what is being quantified: voltage regulation device operations (regulators and switching capacitors), power quality analysis, time outside normal operations, and line losses. Each of these applications will serve as the evaluation metrics for calculating the errors of QSTS simulations relative to the yearlong 1-second resolution simulation described in Section III. For each evaluation criteria, maximum acceptable error thresholds have been set based on feedback from distribution system engineers on their expectations of the performance of QSTS simulations.

All results in this section are from simulations in the open-source software OpenDSS from EPRI [58]. OpenDSS is commonly used to model solar on the grid because of its time-series capabilities [10, 25, 59]. The program was designed to help distribution planners analyze various issues with distributed generation. All power flows were solved with OpenDSS and the results transferred to MATLAB through a COM interface [60]. The results shown are from the test circuits described in Section 3.3.

2.3.1. Data Needs - Input Data Resolution

The application of QSTS simulations requires more data to represent the time-varying PV output coincident with time-varying load. The necessary data set can become very large depending on the resolution and length of simulation desired. Any time that large datasets are used, there will be significant effort required for cleaning, processing, and validating the accuracy of the data. QSTS simulation introduces new and more complex data requirements for power flow simulation. The data requirements for QSTS can be divided into three categories: model data, load data, and PV data.

The implementation of QSTS may require the gathering of additional distribution system model data, including time delay control settings on voltage regulation devices such as capacitors and VREGs. The details of voltage regulation controls, such as reset modes and delays, are not necessary for snapshot power flow simulations, but this information becomes critical for accurate time-series analysis results.

The most ideal input for PV QSTS simulations is high-resolution irradiance data locally measured at the feeder time-coincident to the load data measurements, but there are very few 1-second resolution irradiance data sources in the United States. Commonly, either low-resolution data is used for the analysis, or high-resolution data from a distance geographical location is transformed to the study location. Using a basic diurnal PV output pattern, such as a clear sky model [61], could still provide some valuable insights into potential interactions between the load and PV generation. When using high-resolution irradiance data, it is important to correctly model the reduction in variability of the PV power due to the geographical smoothing that will occur over the area of the PV plant. Methods like the Wavelet-based Variability Model (WVM) can be used to estimate the PV power output using an irradiance point sensor [62-64].

QSTS simulations require the availability of historical time series load data that is often not easily available at the required time resolution. It is common for utilities to record load data at 15-minute or 1-hour resolution, which may be too low to analyze some distribution system impacts that function on the order of seconds. The most ideal input for PV QSTS simulations is high-resolution irradiance data locally measured at the feeder time-coincident to the load data measurements, but there are very few 1-second resolution irradiance data sources. Some previous work has shown that

high-resolution data (sampled at <1-minute resolution) is more critical for modeling PV than load in distribution system simulations [65], and that using 1-minute resolution PV data instead of 1-second resolution can result in up to 18% error in the number of voltage regulator tap changes [66]. Here we perform a similar study for the errors introduced by using lower resolution input data in QSTS simulations on the test feeder. The 1-second irradiance data is averaged over the larger sample periods to represent a standard datalogger and then the QSTS resamples it back to the 1-second time-step resolution with linear interpolation.

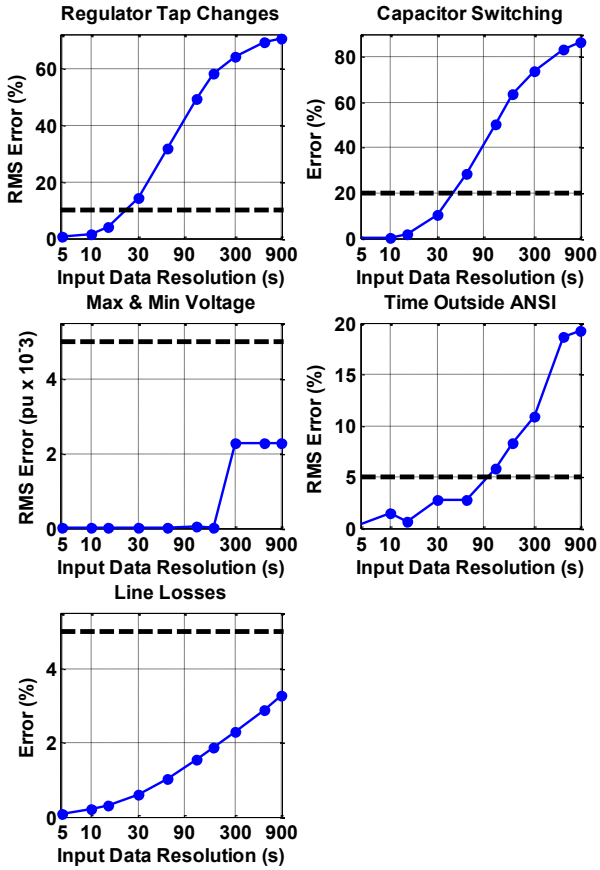


Figure 1. Errors in a yearlong QSTS simulation by using input data at a coarser resolution than 1-second.

Figure 1 shows the errors for using lower resolution PV irradiance input data. The root-mean-squared (RMS) error of the number of tap changes for the three regulators is shown in the top left. The maximum and minimum voltages that occur anywhere on the feeder at any time of the year are shown in the middle left plot. The middle right plot is the RMS error of both the time below and time above the ANSI C84.1 allowable voltage ranges. The error for the yearly number of capacitor switches and total line losses are also shown. For each evaluation metric, the acceptable error threshold is shown with the dashed black line.

2.3.2. Simulation Duration

Previous research has shown how extended time horizon simulations are necessary to capture the seasonal impacts of PV on the distribution system. A more detailed discussion on performing PV interconnection studies and additional analysis examples is shown in [17]. As an example, the feeder in Figure 2 has a substation LTC with LDC and two switched capacitors. The substation

transformer serves a total of four feeders, and the other three feeders were simulated as lumped loads. This analysis investigates operations of voltage regulation equipment during a 9-month simulation with the central PV system connected at the furthest three-phase point on the feeder that could support the PV plant without exceeding conductor ampacity ratings. Coincident feeder load data and local high-resolution irradiance measurements are used for the simulation. The hypothetical PV plant has a nominal capacity of 7.5 MVA output at unity PF, which is equal to 100% of feeder peak load. The simulation was run at 1-second resolution from January 1, 2011 through September 30, 2011. Table 3 shows a comparison of the base case and PV case with regard to LTC and switched capacitor operations for the 9-month simulation.

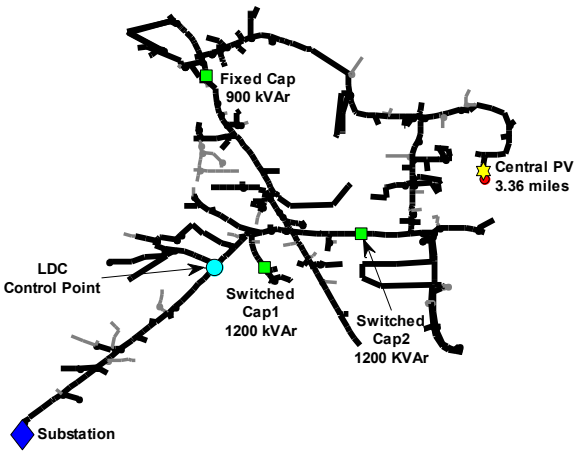


Figure 2. Distribution feeder with one central plant

Table 3. Device Operation Comparisons for the Base Case and PV Case.

Device	Operations Base Case	Operations With PV (Differential)	Percent Change
LTC	459	394 (-65)	-14%
Cap 1	12	6 (-6)	-50%
Cap 2	16	28 (+12)	+75%

The addition of PV resulted in a net reduction in operations observed over the 9 months for the LTC and Cap 1 (nearest the substation) and an increase in the operations for Cap 2. The substation LTC is a ±8 step device, unlike the more common ±16 step devices, which means that each tap change results in twice the voltage change per step and fewer operations. Figure 3 shows a column plot with the total LTC operations by month for the 9-month simulation of both the base case and PV case. The differences shown in Figure 3 highlight the periods where PV causes the greatest decrease in operations, which is during the summer months, and a small amount of additional operations, which occurs in the winter months.

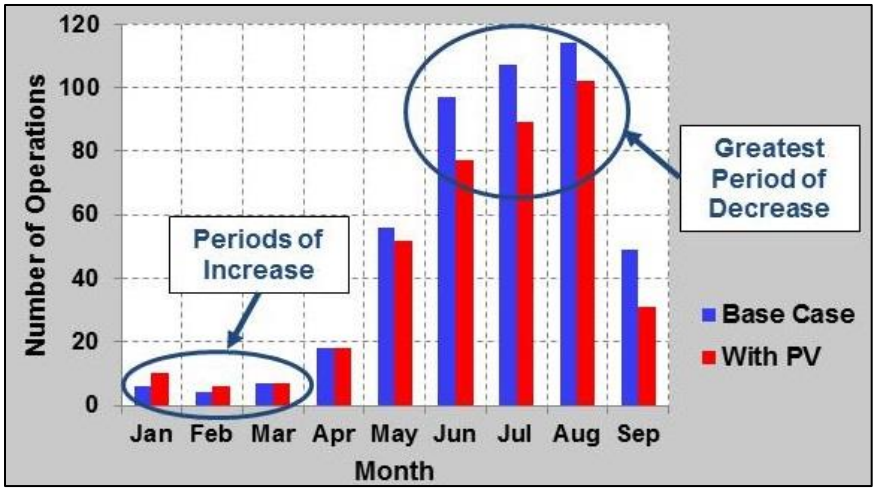


Figure 3. LTC Operations by Month, Base Case and With PV, 9-Month Simulation

Note that the distribution system is connected to a stiff 115 kV transmission system. This means the number of LTC operations will be lower than an LTC connected to either a weaker grid (low short-circuit current) or a lower voltage transmission system. The LTC is connected to three additional distribution feeders, so a high penetration of PV on one feeder may not considerably affect the substation current or the number of LTC operations. PV variability can have a more significant impact on distribution system LTC's for a weaker grid and when there are fewer feeders on the transformer. This example illustrates the value of doing a yearlong analysis instead of only studying a 1-month period that could have predicted a significant increase in LTC tap changes if only January had been used.

Distribution system analysis depends on the system load and other input data, all of which are heavily seasonally dependent. In [55], distributed PV was shown to reduce the number of voltage regulator tap changes some times of the year and increase it during others. In order to capture the seasonal variations, an extended yearlong QSTS simulation is often required. The necessary length of time for a QSTS simulation is studied in this section by performing 1-second resolution QSTS simulations of the test feeder for a subset of the number of days in the year. To study possible different sampling methods, a Monte Carlo analysis was performed with a random sampling of the 365 days of the year. The Monte Carlo analysis is performed 100,000 times for each number of sampled days to create a distribution of error, which is shown in Figure 4. For each Monte Carlo run, random days are selected throughout the year, and the number of tap changes during those days is used to predict the tap changes for the entire year. When only running two days, the error in estimated tap changes can be anywhere from -100% to 350% error. Note in Figure 4 that by selecting more days, e.g. 18 days instead of 2 days, the mean absolute error (MAE) for the estimated number of tap changes decreases from 53% to 17%.

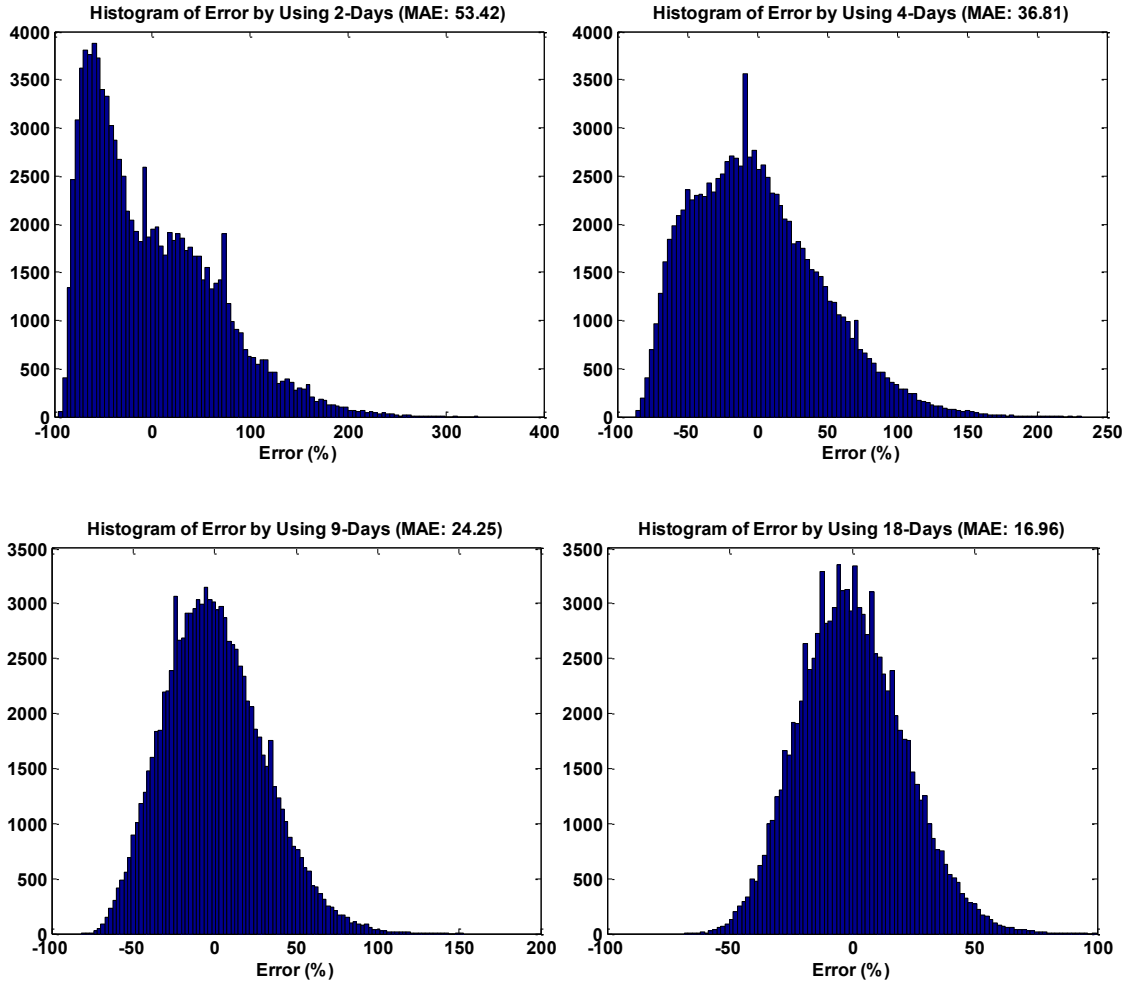


Figure 4. Monte Carlo simulation randomly selecting a few days from the yearlong QSTS simulation of the IEEE 13-node to predict the number of tap changes for the year.

Other metrics than tap changes can also be investigated using the Monte Carlo simulation to determine the possible error by running fewer days than the full year, as shown in Figure 5. Note that some metrics are easier to predict using a fewer number of days, but that even using half the year (180 days), some of the Monte Carlo simulations always fall outside the allowable error range shown by the dashed black line. The resulting range of errors demonstrate that there may be some promising methods to select the best days of the year and be at the bottom of error bars, but the median value (black dot) is generally consistently high for only doing a simulation for part of the year. Figure 5 also shows that the time the feeder will be outside the allowable ANSI voltage range during the year cannot be estimated by only solving part of the year.

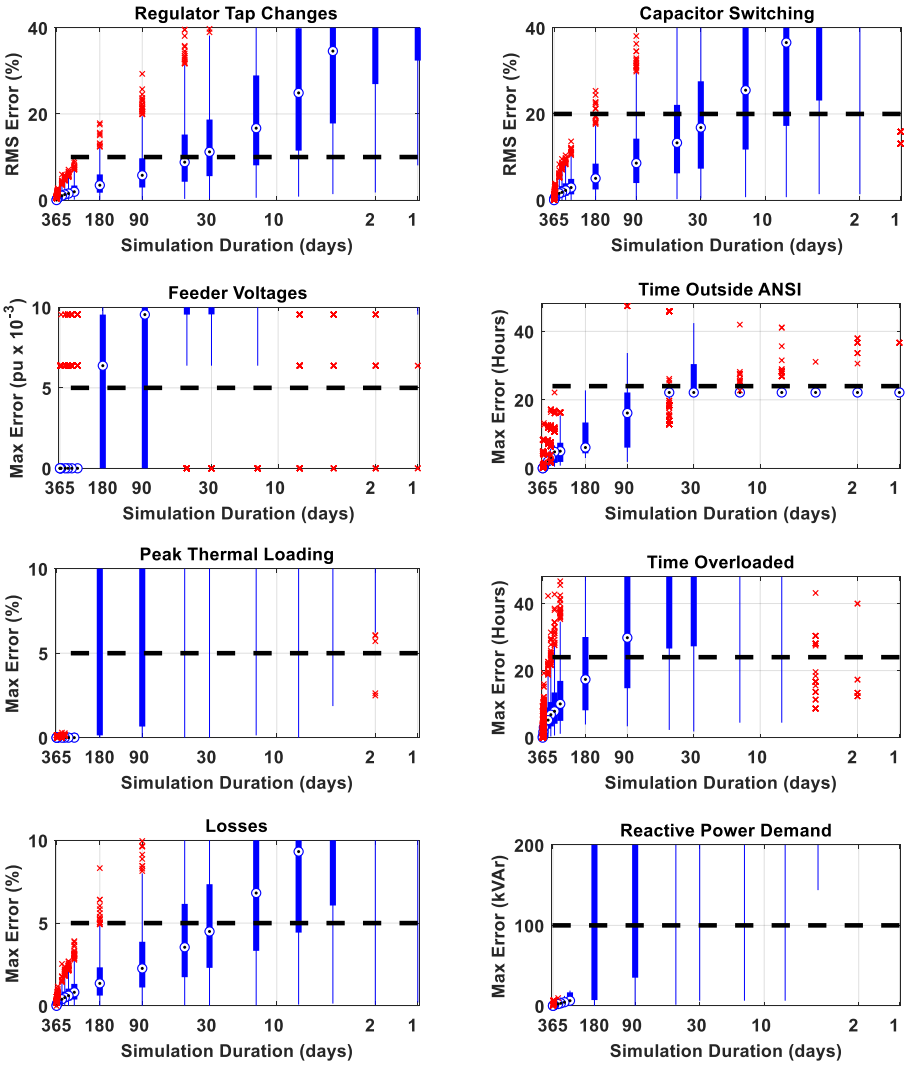


Figure 5. Errors in estimating a yearlong QSTS simulation when only solving a subset of the days for IEEE 13-node

A 100,000 Monte Carlo simulation was performed for five different distribution systems. For each simulation, the metrics are predicted based on a subset of the days, and the accuracy is calculated compared to the yearlong QSTS simulation. Figure 6 shows the error for each type of metric for the five feeders. In order to bound the error within a reasonable range, Figure 6 shows the maximum error returned during the Monte Carlo simulations, which represents the worst-case scenario for error that can be returned by running less than a yearlong QSTS.

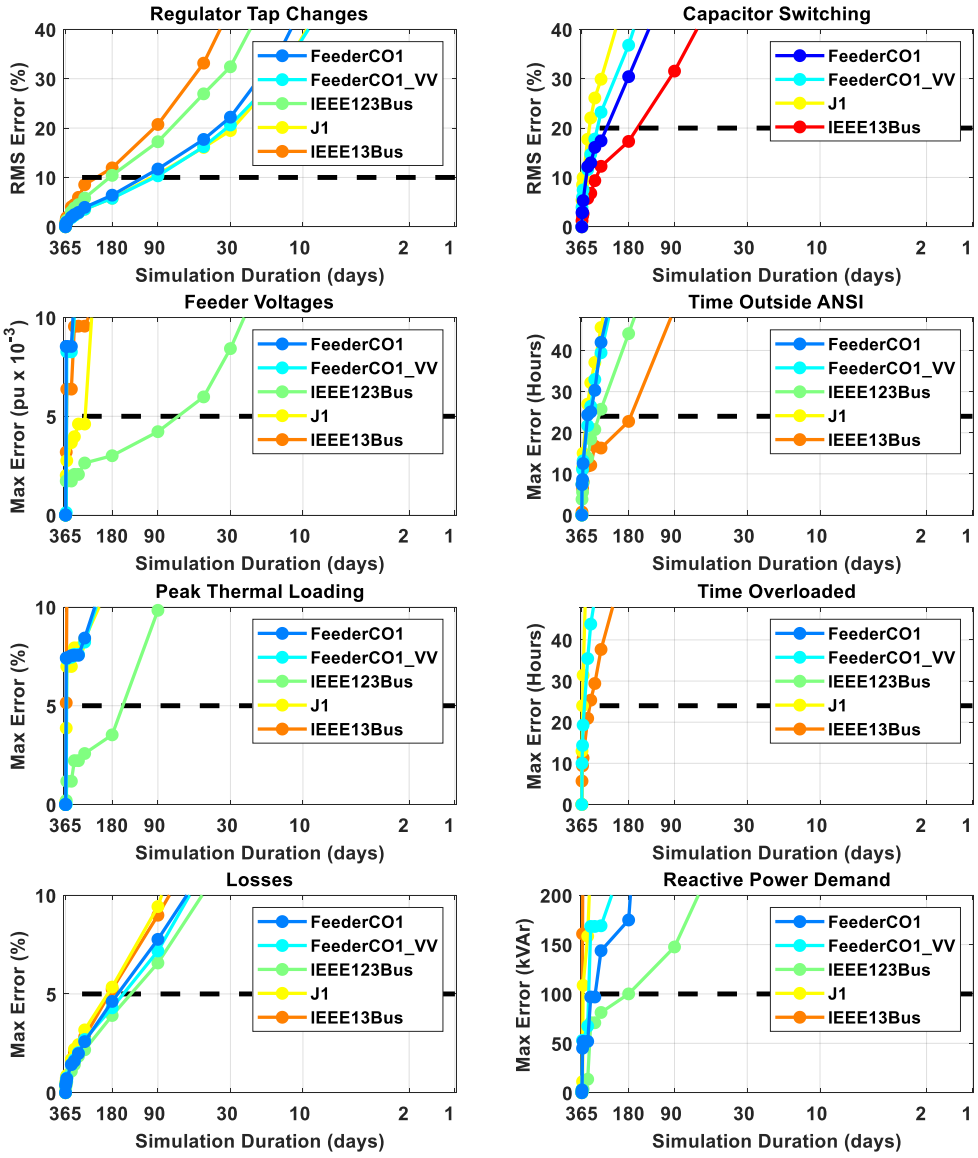


Figure 6. Errors in predicting the operation of different feeders over a year using a subset of the days for a QSTS simulation.

From the results above, selecting days randomly to represent the year introduces significant error. In Chapter 3 of the Rapid QSTS method, it is studied if there are more intelligent ways to select a subset of days [67], and then use machine learning methods to predict the rest of the year from the sampled days [68-70].

2.3.3. Simulation Time Step Resolution

Voltage regulation equipment includes a controller action delay (typically around 30-seconds) where the voltage must remain out of a specified band during the delay period before regulation equipment will initiate an action. This delay function keeps the voltage regulation equipment from reacting to voltage transients. To capture the response of typical distribution equipment, QSTS simulations should have time steps of 1–60 seconds to simulate times when the voltage came back in band during the delay to reset the counter [6]. For any time step longer than the delay, the QSTS

simulation will not take the control action until the next power flow is solved and the controller delay has expired. If the simulation time step is long, the state of the system is less likely to still be out of band when the system is next solved, so the number of operations decreases as the time step lengths are increased. In [55], it is recommended that the simulation time step should be shorter than the shortest time variable in the system (e.g. 30-second time step if the time delays are 30-second).

The requirements for QSTS time-step are studied quantitatively by performing a yearlong QSTS of the IEEE 13-node at various time steps, and the errors for time-steps longer than 1-second are shown in Figure 7. The results demonstrate that high-resolution QSTS simulations are not required to estimate the line losses or extreme voltages, but voltage regulation equipment operations have a noticeable error at time-steps greater than 5-second resolution and significant error at greater than 20-second time steps.

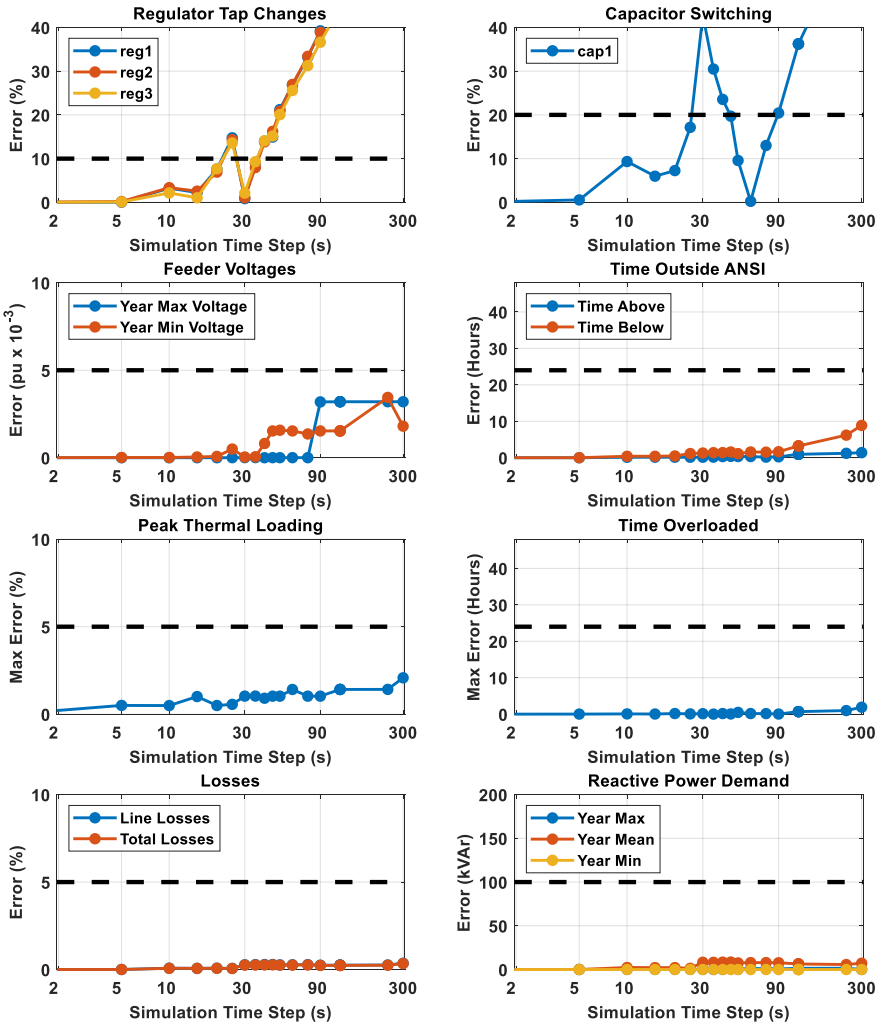


Figure 7. Errors introduced by performing a yearlong QSTS simulation at time step resolutions greater than 1-second for the IEEE 13-node system.

Each of the five distribution systems was run at increasing length in time step, and the accuracy is compared to the yearlong 1-second resolution QSTS simulation.

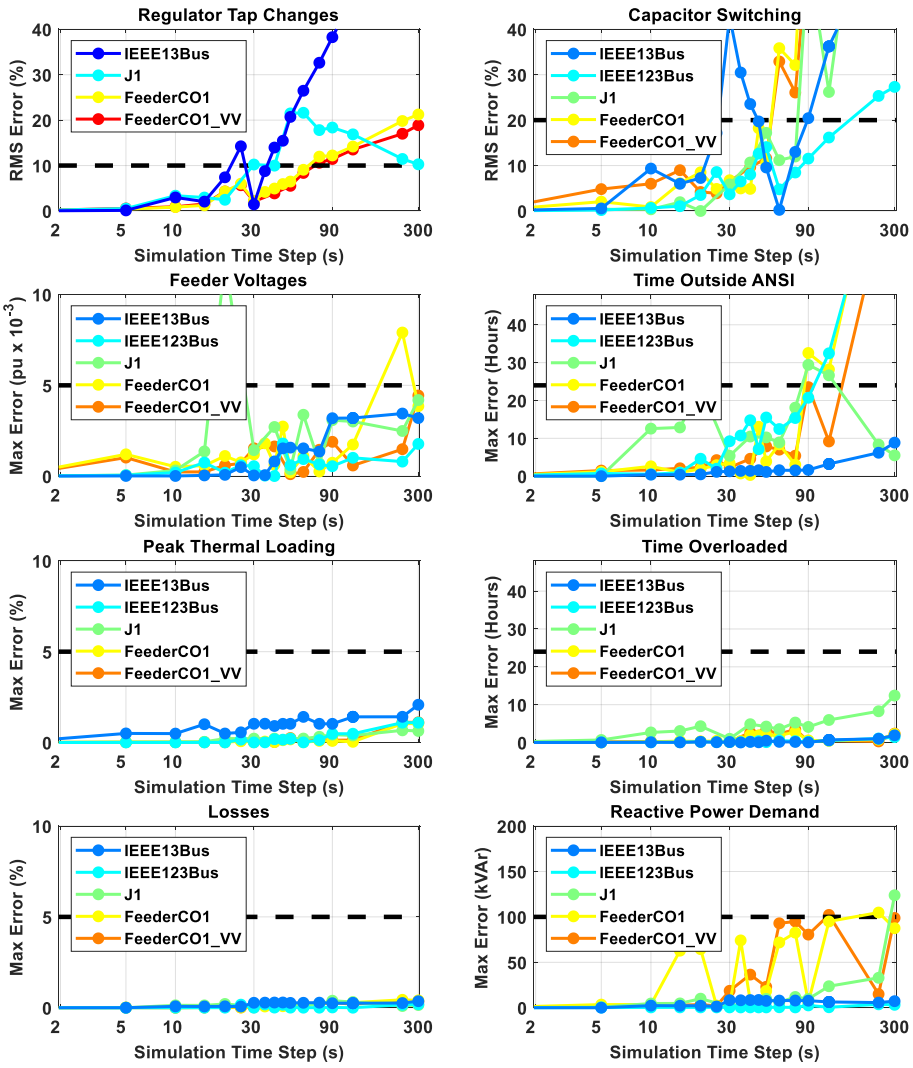


Figure 8. Errors introduced by performing a yearlong QSTS simulation at time step resolutions greater than 1-second of different feeders.

2.4. Summary of QSTS Standards

In general, it is more important to do longer simulations than higher resolution simulations if limited by computational power. This is especially true for estimating worst-case voltages and time outside of ANSI. To demonstrate this, Figure 9 shows the previous error graphs plotted on the same axis of times faster in computational time. For example, a 2-second yearlong simulation (half of the computational time) has 0% error for regulator tap changes, while a 1-second time-step simulation of half the days in the year has a 12% error.

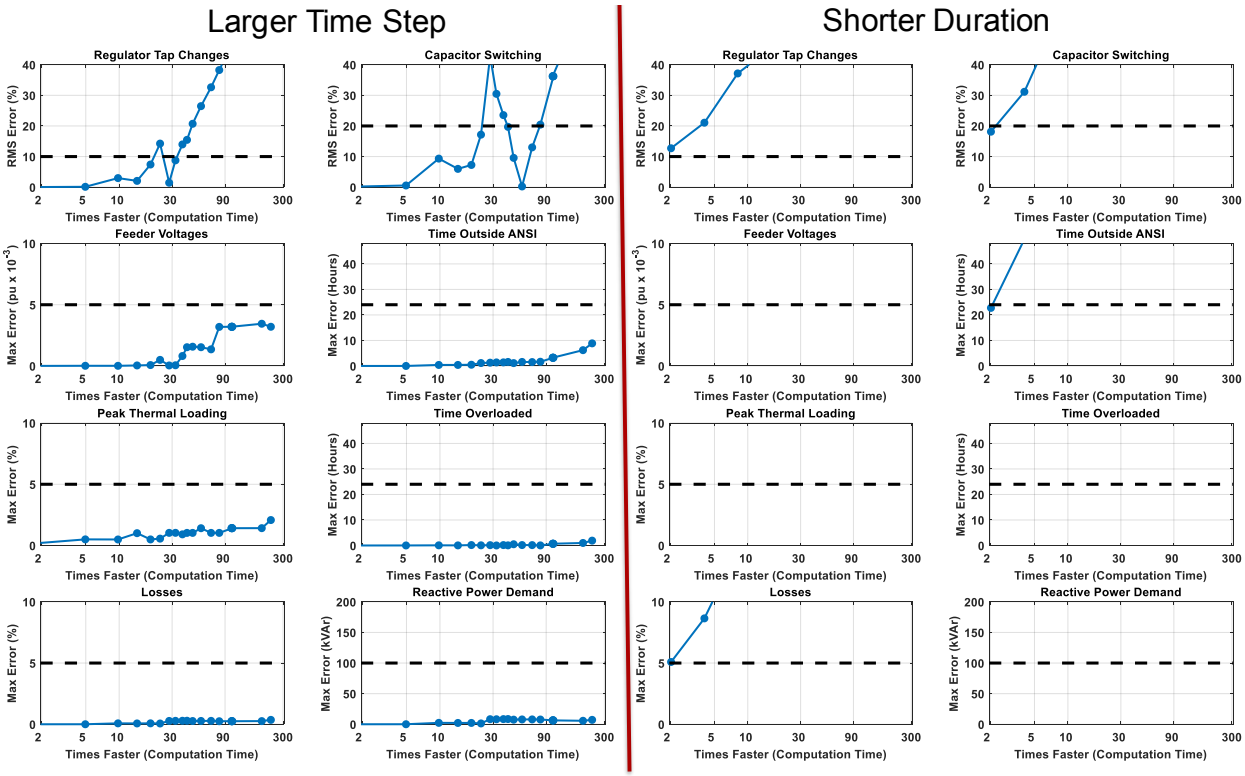


Figure 9. Comparison of error introduced by doing either larger time-steps or fewer days (mean error) than a yearlong 1-second resolution QSTS simulation.

Using the established acceptable error thresholds, Table 4 shows the minimum requirements for QSTS simulation time-step and length of time for each QSTS analysis type. These error results and requirements can vary depending on the distribution system configuration, input time-series data, voltage regulation controls, error thresholds, and other system parameters. Thus, Table 4 should be viewed as rough minimum requirements for having acceptable error for each analysis. Conversely, based on the simulation results, Table 5 shows the requirements for accurate high-resolution QSTS simulations with minimal error in each analysis metric. For this analysis, performing higher-resolution or longer simulations than Table 5 will not provide additional accuracy benefit.

Table 4.QSTS Requirements for Simulation Time-Step and Length of Time to be within an Acceptable Error within the Thresholds

Analysis Metric	Required Time-Step (sec)	Required Length of Time (% of year)
Voltage Regulation Equipment Operations	≤20-second	≥20%
Extreme Max / Min Voltages	~15-minute	≥50%
Time outside ANSI	≤30-second	≥90%
Line Losses	~60-minute	≥10%

Table 5. QSTS Requirements for Simulation Time-Step and Length of Time to have Minimal Error

Analysis Metric	Required Time-Step (sec)	Required Length of Time (% of year)
Voltage Regulation Equipment Operations	≤5-second	≥50%
Extreme Max / Min Voltages	≤1-minute	≥90%
Time outside ANSI	≤5-second	100%
Line Losses	≤5-minute	≥80%

Yearlong high-resolution QSTS analysis is required to adequately model DER impacts on the distribution system. The total line losses for the year can be accurately estimated using shorter low-resolution simulations. However, shortening the simulation to even half of the year can result in extremely high errors for estimating the worst-case voltage magnitudes and hours the feeder will be outside the ANSI voltage range during the year. The interactions and number of actions taken by voltage regulation equipment can be modeled with QSTS simulations using a time-step resolution of up to 5-seconds, but a time-step resolution of 20-seconds or greater begins to induce large errors. In order to be able to capture all distribution system analysis metrics together accurately, a time-step resolution of less than 5-seconds and a time horizon of an entire year is recommended.

3. RAPID QSTS METHODS AND RESULTS

Quasi-static time-series (QSTS) simulations allow for proper modeling of control time delays and voltage regulation operations for analyzing high PV penetration impacts on distribution feeders. However, emerging QSTS requirements introduce new modeling and analysis challenges including new device and controller model development, more profound system details, and more intense data inputs [17]. The use of QSTS analysis by the utilities has been limited by the available simulation software modeling capabilities, the lack of available input data, and the computational intensity of the QSTS analysis. While there are now QSTS capable modeling software options available and more utilities have the necessary data sets for QSTS simulation of PV data impacts, the computational time remains a challenge. Therefore, it is highly desirable to reduce the computational time and complexity of QSTS analysis.

3.1. Challenges to Increasing the Speed of QSTS

In this section, we discuss the challenges in reducing the computational time of QSTS simulations that is described in detail in [71]. This power system research direction differs from the conventional research directions (OPF, unit-commitment, etc.) with its temporal aspect and the consideration of voltage regulating devices. Subsections 1-6 address six major considerations that impact QSTS.

1) The first challenge is the fact that the power flow solution can be very fast, but the difficulty is solving the power flow equations 31 million times for each second of the yearlong QSTS simulation.
2) The second challenge is the aspects of distribution system modeling that make the problem nonlinear, discontinuous, and unpredictable due to the circuit complexity.
3) The third challenge is the time dependence between each power flow solution that requires QSTS simulations to be solved sequentially.
4) The fourth challenge is hysteresis and dead bands in distribution system controls that allow for several independent states.
5) The fifth challenge is the modeling of the interactions between different controllers and the cascading error that is caused by slight variations in models.
6) The sixth challenge is the speed challenge created by logging massive amounts of data during a QSTS simulation and calculating accurate evaluation metrics.

3.1.1. Challenge 1: Number of Power Flows to Solve

Problem Statement: A yearlong QSTS simulation at 1-second granularity solves 31.5 million static power flows. No matter how fast the iterative algorithm is at solving the unbalanced three-phase nonlinear power flow equations, computing it 31.5 million times requires significant computational power.

Discussion:

Iterative power flow solvers have been researched and implemented since the 1950s [72] [73]. They are at the core of numerous power system analyses; consequently, significant effort has been devoted to improving their simulation speed (i.e. reduce the number of iterations and accelerate each iteration) over the decades. Fast iterative algorithms for distribution systems are already implemented in commercial software like CYME and in open-source packages like OpenDSS, or GridLAB-D.

Two directions can be taken in reducing the computational time of power flow solvers: the time it takes to solve the set of equations and the number of iterations it takes to converge. Various iterative techniques such as Newton-Raphson or Fixed-Point method have been applied in the literature and in commercial programs to solve the power flow equations. Different power flow solution techniques may offer better computational time per iteration depending on the feeder characteristics and especially on how they are implemented in a specific coding language. Power flow approximations are a good way to reduce computational time. On the other hand, significant computational time reductions of the iterative solver can be theoretically achieved by reducing the number of iterations, but progress in this direction has already been pursued and solvers have already been optimized to converge with a small number of iterations within the scope of QSTS simulations.

Power flow approximations

In order to reduce the computational time, one alternative is to apply approximations to the power flow equations or their solution methods. Numerous power flow approximation methods are presented in the literature [71]. However, many of these approximations do not work well with multi-phase unbalanced distribution systems (e.g. due to their low X/R line ratio), and all power flow approximations suffer in certain conditions (whether in terms of accuracy or robustness). Moreover, the interdependency between time-steps of the QSTS simulation requires each time step to be solved chronologically which furthers the argument that much more important gains can be obtained by reducing the sheer number of time-steps to be solved as opposed to the CPU time of individual power flow solutions. Therefore, even with fast iterative solvers, solving the unbalanced three-phased power flow equations millions of times requires significant computational power.

3.1.2. Challenge 2: Circuit Complexity

Problem Statement: The set of power flow equations for an unbalanced, 3-phase system is nonlinear by nature. When considering various controller logics, the QSTS simulation becomes a discontinuous nonlinear system that can be very complex. Simplifying this system can be very challenging without having prior knowledge of how it behaves.

Discussion: Size of the distribution feeder

The computational time of the power flow solver is proportional to the number of nodes since each additional node increases the number of equations required to be solved. This correlation can be demonstrated by solving a yearlong QSTS simulation at 1-second resolution in OpenDSS using the KLU solver for three different circuits. The computational time versus the number of nodes is plotted in Figure 10. Since most distribution feeders have 1000+ buses [74], the size of the modeled distribution circuit has a significant impact on the speed of a QSTS simulation. Even with fast iterative solvers, additional work can be done to address the size of the feeder to reduce the computational speed.

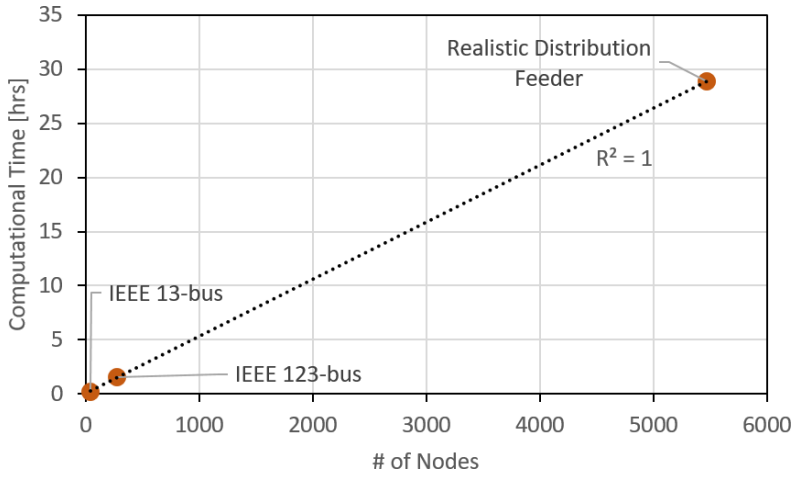


Figure 10. Computational time of a yearlong QSTS simulation at 1-second resolution in OpenDSS as a function of the number of nodes.

Discussion: System unpredictability

The nature of this discontinuous nonlinear system makes it especially challenging to predict how it will behave. For instance, the size of a PV system may or may not impact the operation of various controllers on a feeder. Furthermore, their impact is neither continuous nor linear as shown in Figure 11.

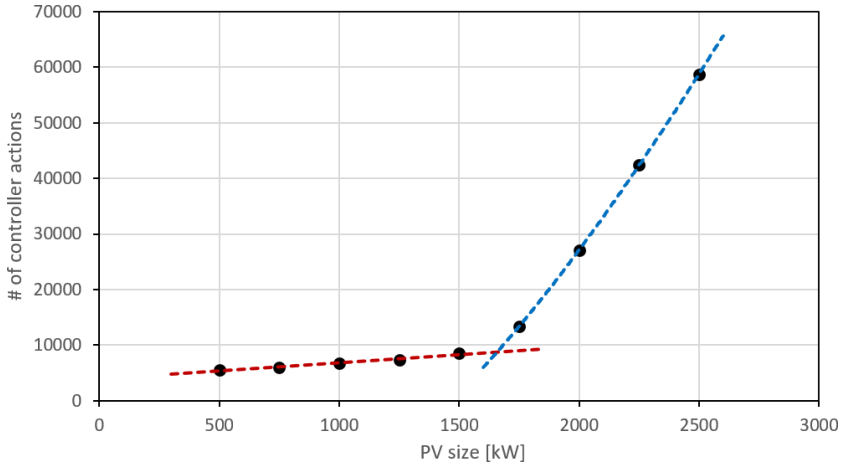


Figure 11.Total number of controller actions on a modified IEEE 13-bus test circuit with a centralized PV system of different sizes.

As shown in the figure above, the correlation between the number of controller actions and PV size is approximately linear only until it reaches ~1500kW. This is a trend that would have been very challenging to foresee based on the characteristics of the feeder.

The size of the PV system is not the only factor that causes unpredictability in the system. Its locations, whether it is distributed or centralized, the controller settings, the location of voltage regulating devices, their interactions, etc. are a small subset of the factors impacting the operation of a distribution feeder. The circuit complexity creates a challenge of unpredictability in the system that makes modeling QSTS simulation without going through each time-step challenging. Therefore, this nonlinear (power flow equations) and discontinuous (control logic) system cannot be easily simplified to study its unpredictable behavior.

3.1.3. Challenge 3: Time Dependence Between Time Steps

Problem Statement: The time dependence in the control logic of certain distribution system devices requires the QSTS simulation to be solved chronologically. For instance, the delays and deadbands in the controllers create a hysteresis in the state of the system. This hysteresis can be a challenge in reducing the computational time of QSTS simulations.

Discussion:

Controller logics in some devices on a distribution feeder can have a time dependence either by design or by nature. Delays and deadbands are often incorporated in controllers (e.g. tap changers or capacitor banks) to ignore any short fluctuation in power flow and avoid oscillation in their operation. Delays can filter out high-frequency variations while deadbands reduce oscillations caused by their own or other devices' operation. In addition to distribution voltage regulating devices, there can be many other devices on the distribution system with time-dependence, such as PV systems with advanced inverter controls and energy storage systems (ESS) state-of-charge (SOC) controls.

The benefit of the QSTS simulation solving power flow chronologically is that the time dependence in the different controllers can be modeled. Delays, deadbands, and SOC can easily be modeled similarly to how they are implemented in the field with if-statements and delay timers. For example, let us use the basic control logic of a capacitor bank on a distribution feeder. As the voltage at the point of interconnection varies, the controller logic monitors the voltage and takes an action if the signal is outside the thresholds once the delay timer has expired. A deadband separates the upper and lower thresholds to avoid fluctuation from the voltage variation created by the action of the capacitor switching.

When solving each time step chronologically, this hysteresis is easily modeled through the modeled logics. As the simulation advances second by second, the time dependence is naturally incorporated with the previous states and any delay timers. This may become a challenge for some computational time reduction approaches if the time steps are no longer solved chronologically. The controller hysteresis may not be accurately modeled or completely ignored which does not realistically represent the operation and state of the system. Therefore, the logic in some controllers introduces a time-dependence between time-steps requiring the QSTS simulation to be solved chronologically, creating a hysteresis in the system states.

3.1.4. Challenge 4: Multiple Valid Power Flow Solutions

Problem Statement: Without the historical information about previous system states, multiple valid power flow solutions exist for given power injections on a feeder due to the deadbands and delays in controller logic. Therefore, correlating these power injections with the states of controllable devices becomes a challenge.

Discussion:

Deadbands are often incorporated within controllers to reduce the oscillation from their own or other devices' operation. As a result, controllable devices on a feeder can have multiple valid states within their controller limits for a given power injection (e.g. for a given demand). For instance, in voltage regulating tap changers, system operators design the voltage deadband to include 3-5 tap positions within the thresholds to avoid oscillation.

A graphical model can be used to illustrate overlapping tap positions for a given demand. Generally speaking, $VregCtrl$ is not linearly correlated with the load, however, in most distribution systems

the error introduced by this linearized model is negligible. When we assume a linear correlation between $VregCtrl$ and system load, we can model each tap position of the regulator as a solid line, as shown in Figure 12

When the load increases from L0 to L1, $VregCtrl$ will drop from V0 to V1; similarly, when the load decreases from L0 to L2, $VregCtrl$ will increase from V0 to V2. In fact, as long as the load maintains within L3 and L4, no tap action will be triggered. However, when the system load moves beyond L3 and L4, a tap action will be triggered, and the tap will move to the adjacent tap position, which corresponds to the adjacent lines in the graphic model.

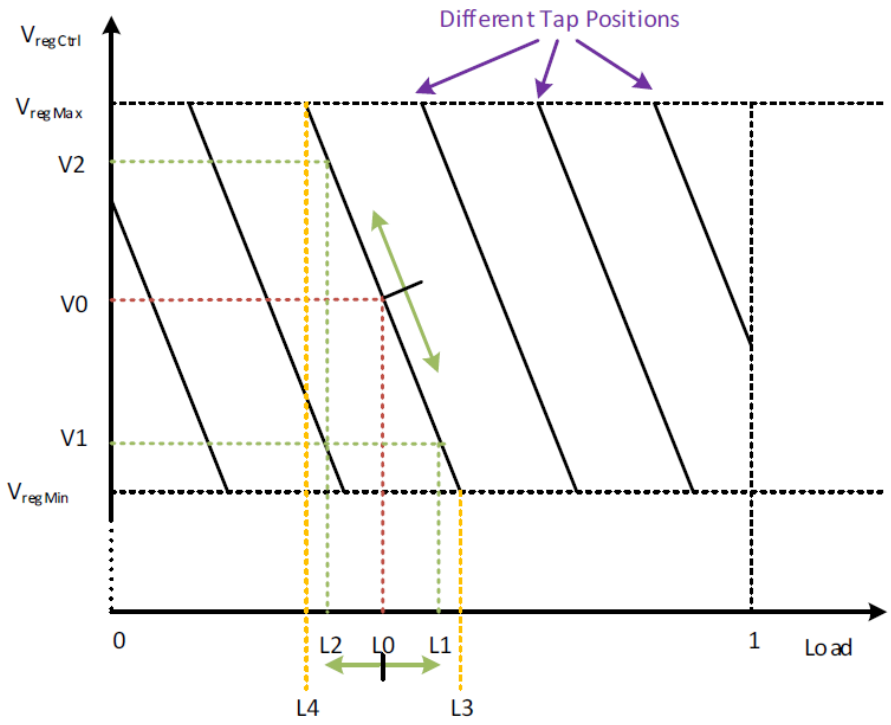


Figure 12. Illustrative representation of regulator control input voltage vs. system load.

When these solid lines overlap each other, one cannot associate a load level to a specific tap position without considering its state at a previous time step. In practice, these solid lines will overlap 3-5 times for a given load. This discontinuity in the relationship between load and system states can become a challenge in approximating control logic models.

Because the deadbands in controllers create a hysteresis in the state of the system, approximating control logic models can become extremely complex without modeling the actual logic of the controllers. Models based on power injections cannot be used when controllers are considered since multiple discrete system states would be valid for the same power injections. This challenge can present a problem for any new QSTS algorithms that do not track the system states through time. The most intuitive solution to eliminate the effect of the multiple valid solutions is to introduce time dependence and time correlation, which itself becomes a new challenge that can be computationally cumbersome to achieve an accurate representation of the operation of the system. Therefore, the deadbands in the controllers can have multiple valid states within their limits, making the correlation between power injections and system states challenging.

3.1.5. Challenge 5: Controllable Element Interactions

Problem Statement: Controllable elements placed on the same phase will interact with one another. Because of their deadbands, an action in one controller caused by a small voltage approximation error in the power flow solution can create false oscillations in other controllers before it can be cleared.

Discussion: Oscillations

Multiple voltage regulating devices can be placed on the same circuit, especially on long radial distribution feeders. Deadbands and delays in the controller of each device are coordinated to avoid continuous oscillations between devices. However, their coordination becomes complex when PV introduces large fluctuations in power injections in the circuit, which can create reverse power flow. This challenge is illustrated with a modified IEEE 13-bus test circuit with 10% and 40% PV penetration Figure 13. Two voltage regulating devices are considered: a voltage regulating tap changer at the substation and a capacitor bank near the PV system.

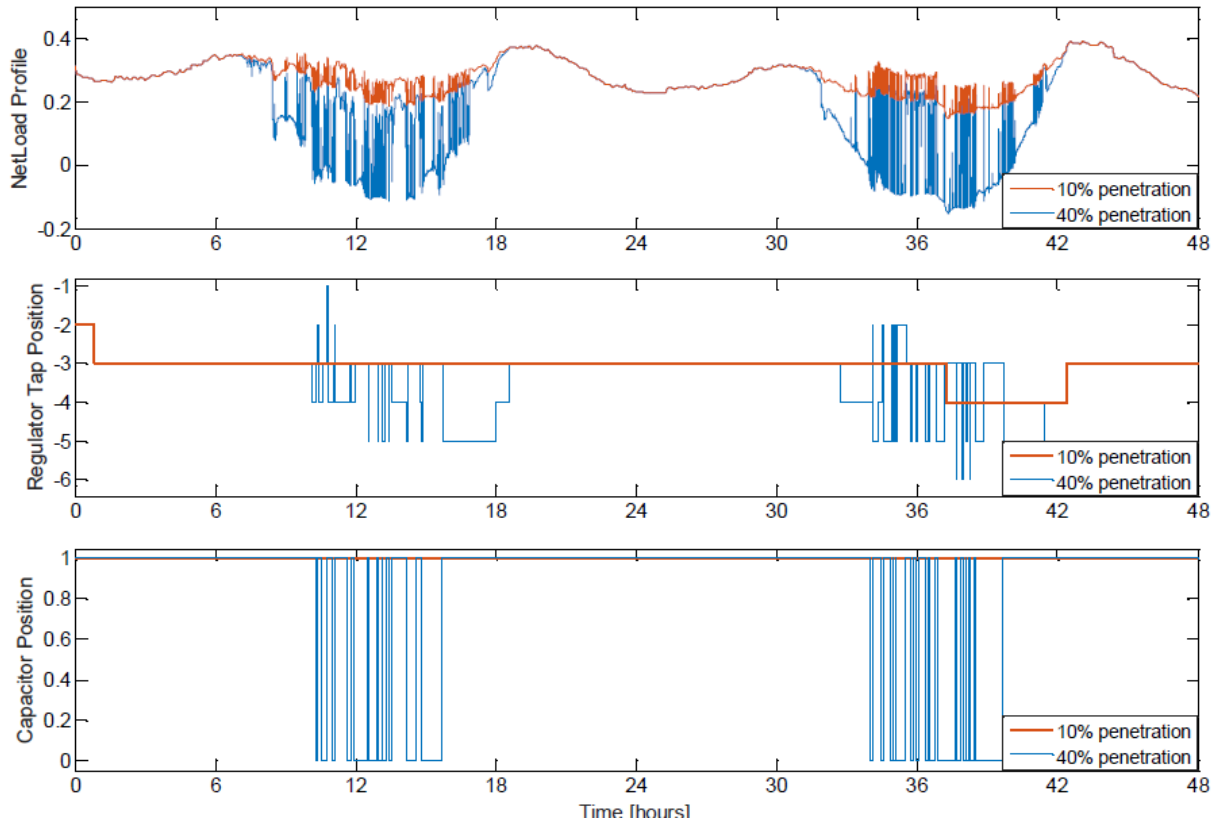


Figure 13. Plot of the net load at the substation (normalized to peak load), regulator tap position and capacitor position over a 2-days period for a system with 10% and 40% penetration of PV. The regulator has 125 tap actions (instead of 3) and the capacitor band.

As expected, the tap changer will regulate the voltage to follow the daily variation of the demand. In the 10% simulation, the controller of the capacitor bank does not operate because of its delay being longer (30 sec.) than that of the tap changer (15 sec.) allowing the voltage to be regulated before the capacitor bank operates. The daily operation of the controllers is very different when a larger PV system is considered. In the 40% simulation, the capacitor bank will operate to regulate the voltage at the end of the feeder. This operation will trigger the tap changer to operate in response to the

capacitor state because of the reactive power injection variation. Since the capacitor bank is more sensitive to the PV system, its operation will increase and consequently increase the operation of the tap changer.

Modeling the interactions between the operations of these voltage regulating devices can be difficult to predict especially when they interact with one another. The deadbands and delays in the controller logic are designed to avoid oscillation under specific conditions. However, new interconnections can disrupt this balance and create emergent behaviors with considerable impacts on the feeder.

Discussion: Cascading error

Another aspect of the challenge with controller interactions is cascading errors. Because of the deadbands in the controllers, the controllable element may trigger a change and remain in that state for an extended period of time. As a result, the operation of other controllers can be significantly impacted by it. For example, the state of a capacitor bank, which is a reactive power injecting device, can impact the operation of an upstream tap changer. Because of the multiple valid power flow solutions discussed earlier, under the same power injection conditions, the operation of the tap changers can increase or decrease dramatically based on the state of the capacitor bank. In Figure 14, a simulation is conducted where a single regulator action is neglected and as a result triggers the capacitor to operate. This single error produces a completely different series of controller events over the following few hours causing the tap changers to record additional actions before returning to identical system states.

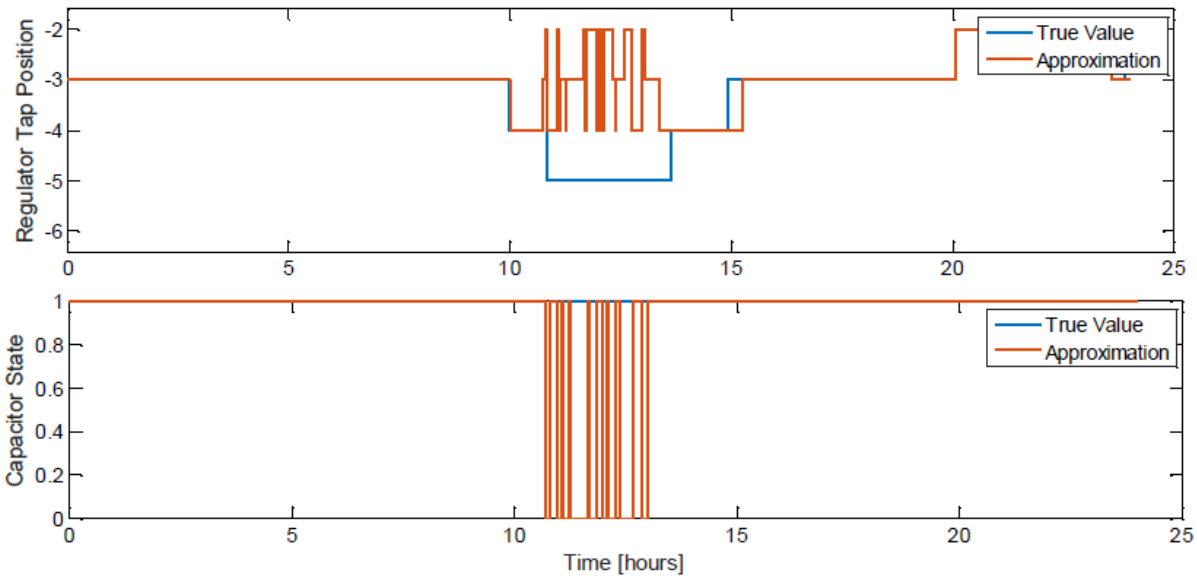


Figure 14. States of voltage regulating devices over a 24-hour period, demonstrating how the interaction between devices can create cascading errors with excess actions.

In this simulation, the cascading error only took approximately four hours to disappear, but it could have easily taken a few simulation days. Speeding the QSTS simulation without going through the controller logic at each time-step is challenging because of these controller interactions. A small approximating error in the power flow solution can create a controller action that can impact how another controller has operated over a period. This is a significant challenge that affects most if not

all computational time reduction approaches. Therefore, because of controllable element interactions, an erratic action in one controller caused by a small voltage approximation error can create oscillations in other controllers before it is cleared.

3.1.6. Challenge 6: Accurate Analysis for Extended Time-Horizon Simulations

Problem Statement: To characterize the impact of a new resource on a feeder, various metrics (e.g. number of tap actions or voltage violations) can be computed posteriori based on the time-series solutions. However, a large amount of data is often required to fully understand its impact. For instance, monitoring voltage violations would require recording voltage quantities for all the nodes in the system for all the time points (e.g. 31 million time points times 10k unbalanced nodes). In addition, the accuracy of each reported metric may be impacted differently based on the approach taken to reduce the computational burden.

Discussion: Data logging requirement

The amount of data to be recorded is dependent on the objective behind the QSTS simulation, whether it is to study the impacts on voltage quality or the operation of controllable devices. The analysis from a QSTS simulation can be categorized into two types of data: discrete metrics or time-series measurements.

Discrete metrics, such as the number of controller actions or total power losses, can be recorded as aggregating values at each time-steps or later computed by recording time-series data to process posteriori. Obviously, there is an advantage for both approaches. Only recording aggregated values does not have a significant memory requirement but will not allow further analysis besides the final discrete metric. On the other hand, recording time-series data requires significant memory but allows post simulation analysis.

Recording data at each time-step can increase the computational time either because of the sheer amount of data (i.e. voltage magnitude) or because of a necessary logic (i.e. tap change if- statement). More specifically, recording time-series voltage measurements for large distribution feeders (500+ buses) may not be possible without running the simulation in sequences. When the power flow solver is contained in its own dynamic-link library (DLL), non-negligible computational overhead may also result from the transfer of large amounts of data at each time- step between the solver and the main application.

The purpose behind running a QSTS simulation is to understand the operation of a distribution feeder under certain conditions. However, it may be challenging to provide a clear understanding of the system without having multiple metrics and/or time-series measurements to analyze. Data management can become a challenge based on the approach taken to reduce the computational time.

Discussion: Metric accuracy

Reducing the computational time of a QSTS simulation can be done in multiple ways: increasing step size, decreasing time-horizon, circuit reduction, etc. However, the accuracy of some reported metrics could be negatively impacted in the process. Each approach used to reduce the computational speed is based on assumptions that may or may not impact the accuracy of a metric. For instance, the increased step-size approach assumes that a single time-step represents multiple other time steps, or the shorter time-horizon approach assumes that a portion of the year is representative of the rest of the yearlong simulation.

Because approaches can compute metrics differently, the accuracy of certain metrics may be impacted in different ways. Some metrics can be time-sensitive, voltage sensitive, or neither. For instance, the total number of controller actions recorded in a yearlong simulation is voltage- and time-sensitive, while power losses in the system are neither. Power losses can be estimated with negligible error without having to solve each time-step chronologically or accurate voltage profiles. Furthermore, the number of controller actions is a discrete metric based on the hysteresis of the controllers, while power losses are not.

Increasing the time-step size is the most apparent approach in reducing computational time. Running the yearlong simulation at a 100-second resolution instead of 1-second will provide a 99% reduction in computational time. However, this approach introduces an error in reported metrics that is time-sensitive. For example, the total number of controller actions by the tap changer will not be accurately reported but power losses will see little impact. (Figure 15)

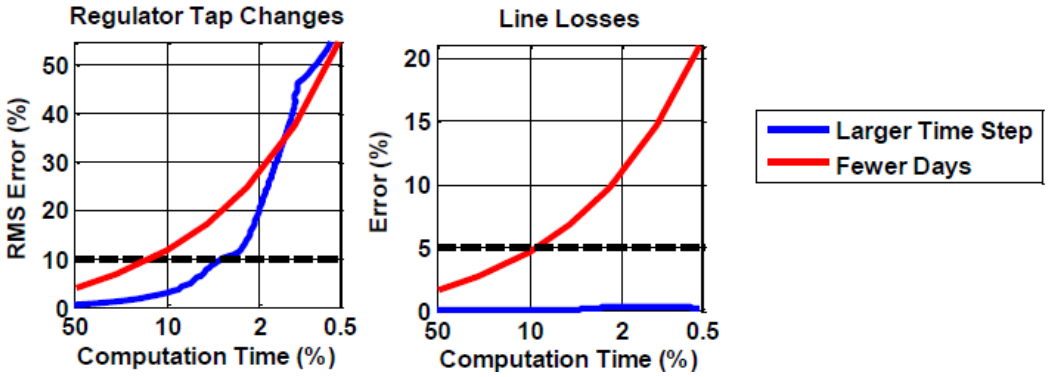


Figure 15. States of voltage regulating device over a 24-hour period demonstrating how the capacitor state can create excess actions by the tap changers

The error associated with the larger time-step size is due to the delays in the controllers. As soon as the time-step size approaches or exceeds the length of the delay, tap actions may or may not be recorded which can accumulate to poor accuracy in reported metrics (See Challenge 5). Reporting various metrics accurately is especially challenging given that this temporal system is a discontinuous and nonlinear system. While considering different approaches to reduce the computational time of QSTS simulations, it is important to retain an acceptable accuracy in the reported metrics. Therefore, the accuracy of each metric reported by the QSTS simulation can be impacted differently based on the approach taken to reduce the computational burden.

This section highlighted the most relevant challenges to reducing the computational speed of QSTS simulation: the number of power flows to solve, circuit complexity, multiple valid power flow solutions, time dependence between time-steps, controllable element interactions, and extensive accurate simulation analysis. The next sections will describe the research efforts in this project to address these challenges.

3.2. Methods

Due to the increasing need for studying the impacts of high penetration PV scenarios and the integration of new smart grid technologies and controls, improving the speed of the QSTS simulation is a major ongoing research area. While there are many challenges to speeding up QSTS simulations, significant improvements have already been demonstrated in prototype cases. Ongoing research in this area includes investigating a variety of approaches such as machine learning, intelligent sampling, variable time-step, event-based simulation, vector quantization, circuit reduction, and temporal and spatial parallelization.

As shown in Figure 16, there are three main ways to decrease the computational time of a QSTS simulation (t_{sol}). The total time can be decreased by reducing the time taken by a single power flow (t_{PF}), reducing the number of power flows solved (N_{PF}), or increasing the computational power (CP) applied to the overall QSTS solution. Figure 17 shows how the proposed rapid QSTS algorithms can be grouped into one of those three categories. The time taken by a single power flow (t_{PF}) can be decreased by circuit reduction or power flow improvements. The number of power flows solved (N_{PF}) can be decreased by using vector quantization, variable time-step, or event-based simulation. Reducing the time-horizon of the required QSTS solution via machine learning or intelligent sampling can also decrease N_{PF}. Finally, the computational power (CP) can be increased by using multiple processors for temporal parallelization or spatial parallelization (Diakoptics).

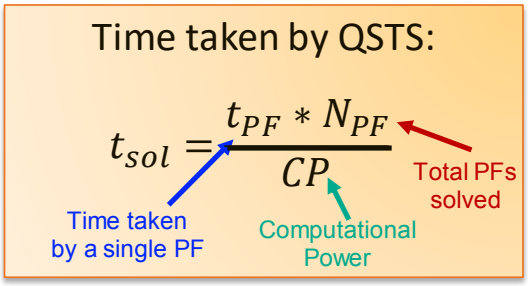


Figure 16. Total QSTS computational time is a combination of the time per power flow (PF) multiplied with the total number of PF, divided by the computational power. Each type of rapid QSTS algorithm is trying to address one of these main pieces of the computational time.

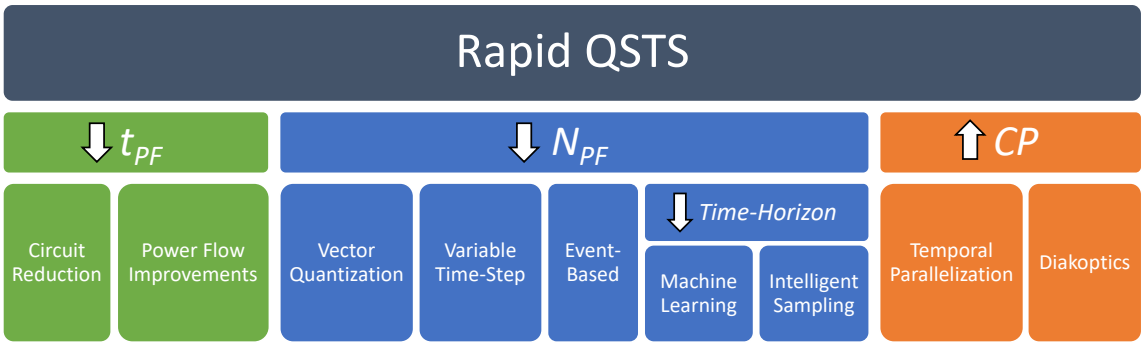


Figure 17. Categories of Rapid QSTS Algorithms.

The following subsections discuss the unique approach that each rapid QSTS method takes to reduce computational time while addressing the challenges highlighted in the previous section.

3.2.1. Variable Time Step

While large time-steps in QSTS simulation can create significant errors, variable time-step algorithms can be used to intelligently modify the time-step size around key periods of interest like periods with high variability or discrete control events. The goal of a variable time-step solver is to focus computational efforts during the periods of the year when the system state is changing rapidly. For example, when the net load demand is fairly constant at nighttime, the QSTS simulation can step forward in time with larger time-steps. For the variable time-step simulation, the input data is preprocessed to define the periods of interest that exceed a deviation threshold, as shown in Figure 18. The backtrack-based solver proceeds at the max time-step until a system state change is detected, at which point it steps backward in time to search at high-resolution for when the event occurred.

Advantages:

- Easy to implement as it does not change the solver, only how the simulation progresses through time
- Works with any type of controls, such as advanced inverters

Limitations:

- May miss the extreme operation time points if the max time step size is too large.
- It is challenging to determine the appropriate thresholds for deviation thresholds, max time step, and how often to backtrack (especially with a large number of input profiles, some of which may impact the circuit more than others).

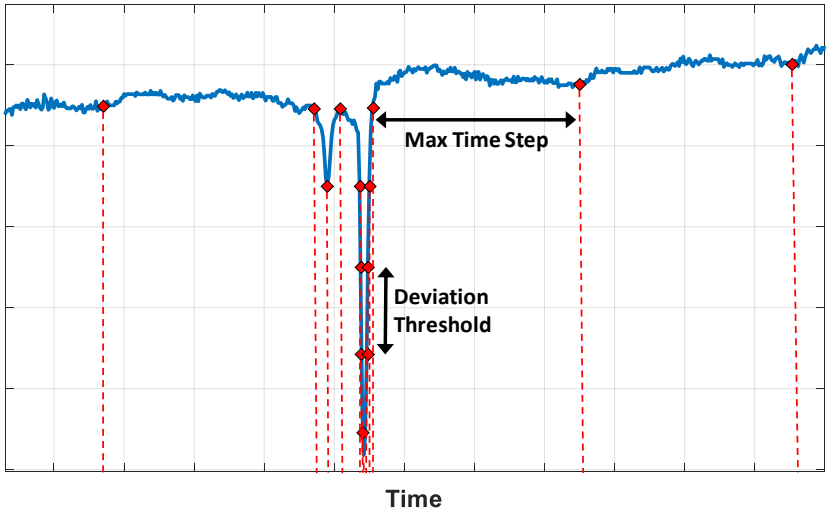


Figure 18. QSTS Variable Time-Step Algorithm

References

- M. Reno and R. Broderick, "Predetermined time-step solver for rapid quasi-static time series (QSTS) of distribution systems," Proc. IEEE PES Innovative Smart Grid Technologies (ISGT) Conf, 2017. [75]
- B. Mather, "Fast Determination of Distribution-Connected PV Impacts Using a Variable Time-Step Quasi-Static Time-Series Approach," in IEEE Photovoltaic Specialists Conference (PVSC), 2017. [76]
- M. J. Reno, J. A. Azzolini, and B. Mather, "Variable Time-Step Implementation for Rapid Quasi-Static Time-Series (QSTS) Simulations of Distributed PV," in 2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC). [77]

3.2.2. Vector Quantization

Vector quantization groups together scenarios throughout the year that produce similar power flow solutions, thereby limiting the total computations of the non-linear AC power flow equations. The input variables are quantized, or discretized, and all power flow solutions with the same input variables are considered the same. So, when a familiar scenario is encountered, the solution can be pulled from a look-up table rather than recalculating the nonlinear power flow solution. Figure 19 shows a heat map of the number of time steps in a year with similar load and PV multiplier values when quantized in 101 clusters.

Advantages:

- Very fast for long-term QSTS simulations, and the longer the duration of the simulation, the algorithm is faster because it is more likely to have previously encountered the solution and saved it in the lookup table.
- Better performance on larger circuits because the lookup table speed is constant but brute force power flow computational time increases with circuit complexity.

Limitations:

- The accuracy is based on the discretization level and determining the optimal resolution for speed vs. accuracy ahead of time is difficult with large numbers of input time-series and regulating equipment.
- The lookup table size increases with more input variables, so the method may not scale well for a large number of input time-series, such as AMI data for thousands of customers

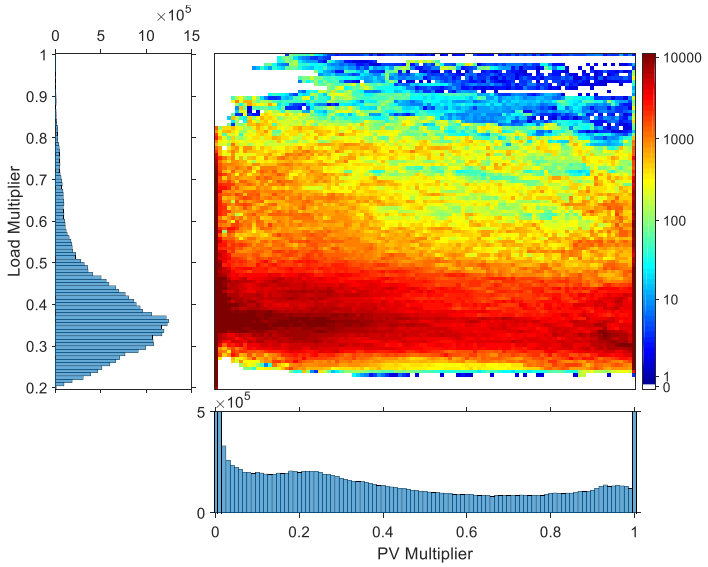


Figure 19. Vector Quantization QSTS Algorithm

References

- J. Deboever, S. Grijalva, M. J. Reno, and R. J. Broderick, "Fast Quasi-Static Time-Series (QSTS) for Yearlong PV Impact Studies using Vector Quantization," Solar Energy, 2018. [78]
- J. Deboever, S. Grijalva, M. J. Reno, X. Zhang, and R. J. Broderick, "Scalability of the Vector Quantization Approach for Fast QSTS Simulation for PV Impact Studies," IEEE Photovoltaic Specialists Conference, 2017. [79]
- J. Deboever, S. Grijalva, M. J. Reno, and R. J. Broderick, "Algorithms to Effectively Quantize Scenarios for PV Impact Analysis using QSTS Simulation," IEEE Photovoltaic Specialists Conference (PVSC), 2018. [80]
- W. Li, B. Mather, J. Deboever, M. J. Reno, "Fast QSTS for Distributed PV Impact Studies using Vector Quantization and Variable Time-Steps," IEEE Innovative Smart Grid Technologies (ISGT), 2018. [81]

3.2.3. *Event-Based*

Event-based QSTS simulations use discrete local voltage sensitivity models to detect and predict when the next state changes, such as a voltage regulator tap change, will occur. Because QSTS simulations are nonlinear (power flow equations) and discontinuous (controller logic), a discontinuous linear approximation can be utilized by creating a separate linear model for each topology and discrete state of the distribution system, as shown in Figure 20. While the time-dependent nature requires the simulation to be solved in sequential order, a combination of the linear approximations can be used to solve the time-series via discrete event-based simulation.

Advantages:

- Very fast using matrix multiplication of the voltage sensitivity model instead of matrix inversion of the impedance matrix. The matrix is also generally smaller since it is based on the number of input variables (load and PV profiles) instead of the number of buses.
- System states and the timing of state changes are predicted very accurately for all devices throughout the year by jumping from event to event. Other quantities can be determined by post-processing with different time steps depending on the desired accuracy.
- Works well with all types of equipment and controllers, including advanced inverters.

Limitations:

- Must re-linearize after every state transition (regulator tap or smart inverter mode change).
- Nonlinear losses are hard to estimate using linear regression, so the present solution is summing the differences in power on either end of lines.
- The sensitivity matrix size increases with more input variables, so the method may not scale well for a large number of input time-series profiles, such as AMI data for thousands of loads.

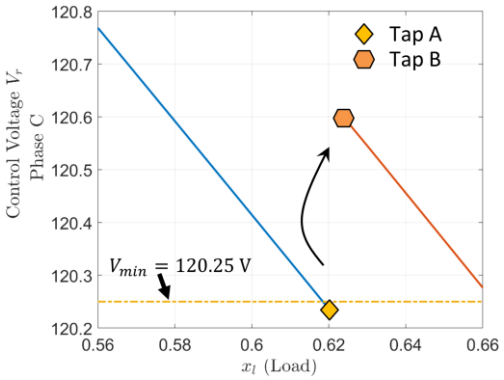


Figure 20. Event-Based QSTS Algorithm

References

- X. Zhang, S. Grijalva, M. J. Reno, J. Deboever, and R. J. Broderick, "A Fast Quasi-Static Time Series (QSTS) Simulation Method for PV Impact Studies Using Voltage Sensitivities of Controllable Elements," IEEE Photovoltaic Specialists Conference (PVSC), 2017. [82]
- M. U. Qureshi, S. Grijalva, and M. J. Reno, "A Fast Quasi-Static Time Series Simulation Method for PV Smart Inverters with Var Control using Linear Sensitivity Model," IEEE Photovoltaic Specialists Conference, 2018. [83]
- M. U. Qureshi, S. Grijalva, M. J. Reno, J. Deboever, X. Zhang, and R J. Broderick, "A Fast Scalable Quasi-Static Time Series Analysis Method for PV Impact Studies using Linear Sensitivity Model," IEEE Transactions on Sustainable Energy, 2019. [84]
- M. U. Qureshi, S. Grijalva, M. J. Reno, "A Rapid Quasi-Static Time Series Method for Evaluating Current-Related Distributed PV Impacts including Feeder Loading and Line Losses," IEEE PES General Meeting, 2019. [85]
- M. U. Qureshi, "A Fast Quasi-Static Time Series Simulation Method Using Sensitivity Analysis to Evaluate Distributed PV Impacts," Georgia Institute of Technology, 2019. [86]

3.2.4. Intelligent Sample Selection

Intelligent Sampling uses representative samples from input time-series data to perform a QSTS analysis. This method first decomposes the year into smaller 6-hour time periods, then characterizes them using the solar variability index and mean load values. The underlying assumption is that similar time periods will have closely correlated QSTS results. After running one period, all other periods are assumed to have the same results, e.g. number of tap changes as shown in Figure 21. So, by only solving the representative samples, the results can be extrapolated for the whole year.

Advantages:

- Intelligently selects pieces of the year to solve and reduces the computational burden by not having to solve the entire year.
- Requires less of the year to be solved compared to random sampling (>85% of the year).

Limitations:

- There is significant variation in the power flows (seasonal and daily) so a large amount of the year must still be run using QSTS to see all possible combinations.
- To maintain accuracy, it was found at least 40% of the year must be solved, with only 2.5x speed improvement.
- Selecting the periods of the year to run is critical and using a small percent of the time can lead to large percent errors.
- Each period of the QSTS is dependent on the previous periods, so sampling and running specific time periods in the year adds error because each period may not be initialized correctly.

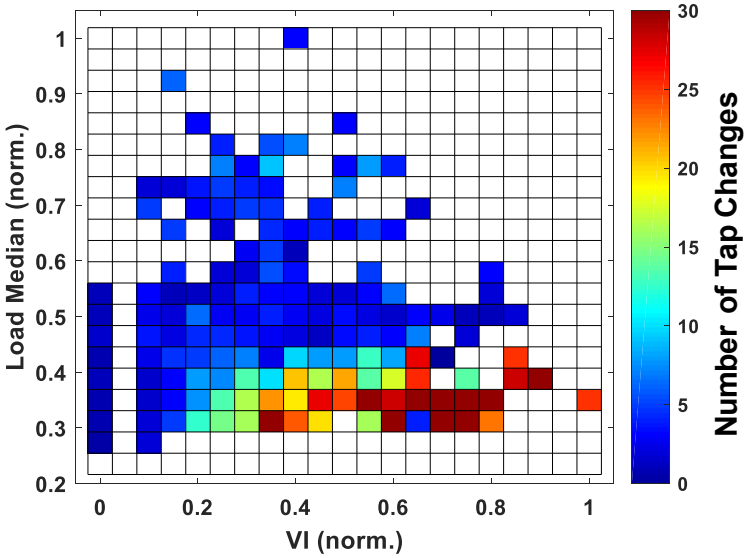


Figure 21. QSTS Intelligent Sample Selection Algorithm

References

- J. Galtieri, M. J. Reno, "Intelligent Sampling of Periods for Reduced Computational Time of Time Series Analysis of PV Impacts on the Distribution System," IEEE Photovoltaic Specialists Conference (PVSC), 2017. [67]

3.2.5. Machine Learning

Machine learning algorithms can be trained from the results from a partial yearlong QSTS simulation to learn the correlation between inputs (load and PV time-series) and the outputs (line losses, number of tap changes, etc.) as outlined in Figure 22. This approach has produced promising results (such as in [68, 87, 88]). Several types of machine learning, including ensembles of multiple types, have been investigated to utilize the brute-force QSTS to initially train the machine learning, which can then be used to accurately model the distribution system for other times during the year. However, the training of the NN is feeder specific and requires at least 21% of the yearlong brute-force QSTS to run.

Advantages:

- Intelligently selects pieces of the year to solve and reduces the computational burden by not having to solve the entire year.
- Significantly reduces the amount of the year to be solved (only 20% required) compared to random sampling (>85% of the year) or intelligent sampling (40% of the year) by learning the correlations.

Limitations:

- There is significant variation in the power flows (seasonal and daily) so a large amount of the year must still be run using QSTS to generate the training data. To maintain accuracy, it was found at least 20% of the year must be solved, with only 5x speed improvement.
- Selecting the periods of the year to run is critical. Depending on the training data, there are sometimes large percentage errors.
- Each period of the QSTS is dependent on the previous periods, so sampling and running specific time periods in the year adds error because each period may not be initialized correctly.
- A separate machine learning algorithm must be trained to generate each metric, e.g. tap changes vs. line losses.

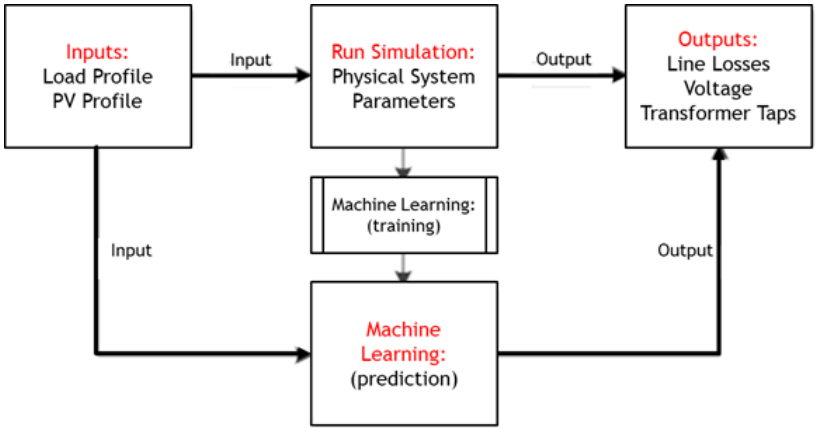


Figure 22. QSTS Machine Learning Algorithm

References

- M. J. Reno, R. J. Broderick, and L. Blakely, "Machine Learning for Rapid QSTS Simulations Using Neural Networks," in 2017 IEEE 44th Photovoltaic Specialist Conference (PVSC), 2017. [68]
- L. Blakely, M. J. Reno, and R. J. Broderick, "Decision tree ensemble machine learning for rapid QSTS simulations," in 2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), 2018. [69]
- L. Blakely, M. J. Reno, and R. J. Broderick, "Evaluation and Comparison of Machine Learning Techniques for Rapid QSTS Simulations," Sandia National Laboratories, SAND2018-8018, 2018. [70]

3.2.6. Diakoptics

Diakoptics is a technique for tearing large physical circuits into smaller geographical subnetworks to reduce the modeling complexity, i.e. spatial parallelization [89]. Figure 23 shows an example of a circuit spatially divided into four different subnetworks. Using modern multi-core computers, available cores will independently solve a separate piece of the circuit in parallel to find a total solution. This technique accelerates QSTS simulations by simplifying the power flow problem to achieve a faster solution at each simulation step. Consequently, the total time reduction when performing QSTS simulations with Diakoptics will be evident at each time step.

Advantages:

- Most computers have multiple cores, so there is an automatic speed improvement.
- If integrated into the power flow solver, every power flow is faster, not only time-series power flows.

Limitations:

- Parallel processors must communicate to reach a converged solution for each power flow which is more complicated to implement.
- Speed improvement is partially limited by the parallel processing implementation and communication between cores.
- Determining the locations to tear the circuit to separate between processors is challenging; if each sub-circuit is not the same size as each other, the speed improvement is reduced.

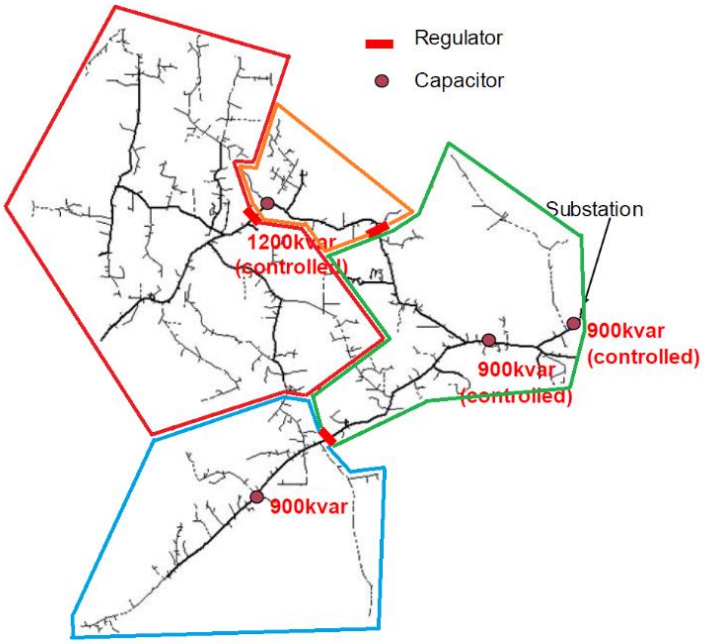


Figure 23. Diakoptics Algorithm

References

- D. Montenegro, G. A. Ramos, and S. Bacha, "A-Diakoptics for the Multicore Sequential-Time Simulation of Microgrids Within Large Distribution Systems," IEEE Transactions on Smart Grid, 2017. [89]
- D. Montenegro, G. A. Ramos, and S. Bacha, "An Iterative Method for Detecting and Localizing Islands Within Sparse Matrixes Using DSSim-RT," IEEE Transactions on Industry Applications, 2018. [90]

3.2.7. Temporal Parallelization

The temporal parallelization method increases the computational power by dividing the total time horizon of the simulation into different sections and solving those sections simultaneously on multiple cores of a single computer. Figure 24 shows a visual representation of a yearlong QSTS simulation divided into four equal parts to be solved concurrently on four separate processors. Since QSTS simulations contain both "parallelizable" components (e.g. power flow solutions) and "serial" components (e.g. assigning a time-series multiplier to a load), the relationship between speed improvement and degree of parallelization is dependent upon both the characteristics of the circuit and the available computing resources. Due to the time-dependence of QSTS simulations and the existence of multiple valid power flow solutions, the unknown initial conditions at each temporal division may introduce errors that can persist through the simulation. The likelihood of these errors increases with the degree of parallelization because each additional temporal division introduces another unknown state into the simulation.

Advantages:

- Most computers have multiple cores, so there the potential for a speed improvement without requiring specialized hardware.
- No modifications to the power flow engine are needed since it is still solving part of the year in the same way.

Limitations:

- Each power flow in a QSTS simulation is based on the previous power flow, so initializing the QSTS in the middle of the year without solving the previous periods introduces error.
- Data management collecting results and combining from multiple simulations can be challenging to implement.
- Some aspects of the simulation, such as I/O, will always be sequential, so speed does not increase linearly with the number of processors (5 times speed improvement seen with 8 cores).

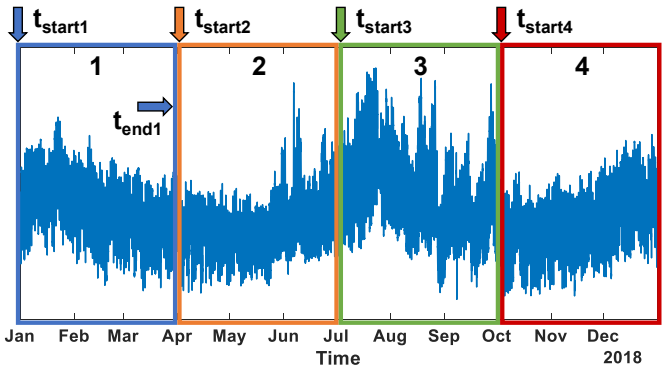


Figure 24. Temporal Parallelization Algorithm

References

- R. Hunsberger and B. Mather, "Temporal Decompostion of Distribution System Quasi-Static Time-Series Simulation," in IEEE PES General Meeting, 2017. [91]
- A. Latif and B. Mather, "Monte Carlo Based Method for Parallelizing Quasi-Static Time-Series Power System Simulations," in 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), 2018. [92]
- D. Montenegro, R. C. Dugan, and M. J. Reno, "Open Source Tools for High Performance Quasi-Static-Time-Series Simulation Using Parallel Processing," in 2017 IEEE 44th Photovoltaic Specialist Conference (PVSC), 2017. [93]
- J. A. Azzolini, M. J. Reno, and D. Montenegro, "Implementation of Temporal Parallelization for Rapid Quasi-Static Time-Series (QSTS) Simulations," in IEEE Photovoltaic Specialists Conference (PVSC), 2019. [94]

3.2.8. *Circuit Reduction*

Circuit reduction creates a physics-based electrically equivalent reduced-order model with fewer buses, thus reducing the total time taken by the solver at each time step. Key buses of interest (BOI) and voltage regulation equipment can remain in the reduced model to preserve simulation accuracy while the total number of buses is reduced. BOI can be user-selected, critical components (voltage regulators, PV with advanced inverters, switching capacitors, etc.), or kept because they are required by the algorithm (source bus, topology buses). However, detecting specific buses of interest algorithmically can be a challenging task given that the lowest and highest voltage extremes can occur anywhere in the feeder. Figure 25 shows an example distribution circuit before and after it is reduced.

Advantages:

- Reduces both the computational time (fewer buses in the power flow solution) and the memory requirements (fewer elements to record).
- Circuit reduction can be applied before implementing other rapid QSTS algorithms.
- Does not introduce error at the buses of interest.

Limitations:

- There is a reduced insight into buses and components that are removed in the reduced-order circuit. For example, if an extreme voltage occurs at a bus that was not selected as a BOI, the extreme voltage will not be seen.
- Advanced inverters with nonlinear volt-var curves (such as a deadband), cannot be aggregated into a fully equivalent reduced order model, meaning all advanced inverters may have to remain in the circuit.

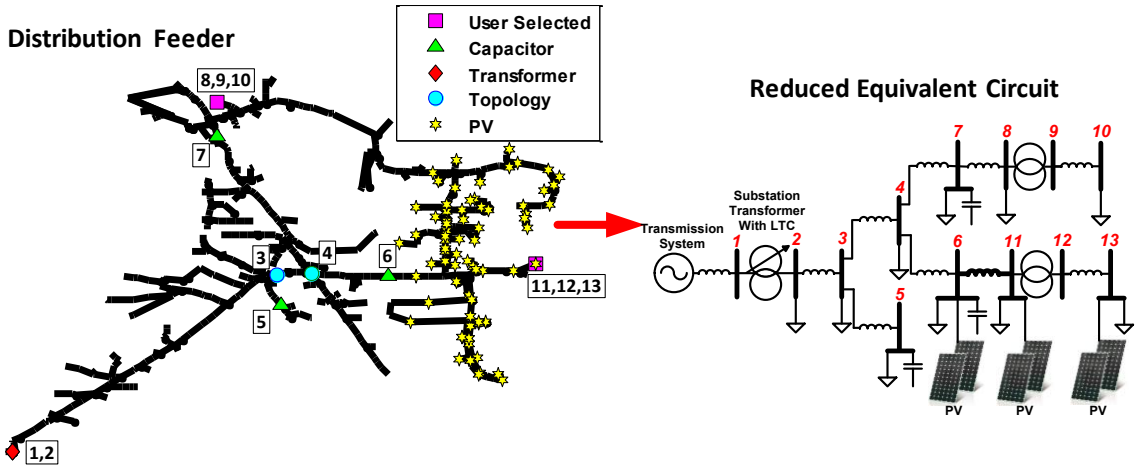


Figure 25. Circuit Reduction Algorithm

References

- M. J. Reno, R. J. Broderick, and S. Grijalva, "Formulating a Simplified Equivalent Representation of Distribution Circuits for PV Impact Studies," Sandia National Laboratories, SAND2013-2831, 2013. [95]
- M. J. Reno, K. Coogan, R. J. Broderick, and S. Grijalva, "Reduction of Distribution Feeders for Simplified PV Impact Studies," in IEEE Photovoltaic Specialists Conference, 2013. [7]
- Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl, "Multiphase Distribution Feeder Reduction," IEEE Transactions on Power Systems, 2017. [96]
- Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl, "Inversion Reduction Method for Real and Complex Distribution Feeder Models," IEEE Transactions on Power Systems, vol. 34, no. 2, pp. 1161-1170, 2019, doi: 10.1109/TPWRS.2018.2872747. [97]

3.3. Test Circuits

Five test circuits were developed for the project to test the QSTS algorithms. Each circuit contains a yearlong simulation at 1-second resolution with high penetrations of PV.

Table 6. Description of the five QSTS test circuits.

Circuit	Buses, Nodes	kV	Peak Load (MW)	Length of Feeder	Secondary System Modelled	Regulating Devices	PV Penetration	Types of PV Systems	Number of Timeseries Profiles
IEEE 13	16, 41	4	4.3	1.5 km	No	3 LVRs with LDC, 1 switching CAP bank	40%	1 centralized	2
IEEE 123	132, 278	4	3.6	6.2 km	No	1 LTC and 6 LVRs with LDC	50%	1 centralized	5
CO1	2969, 5466	12	6.4	21.4 km	Yes	1 LTC and 3 LVRs, 5 switching CAP banks	62%	2 centralized, 142 distributed	6
CO1 VV	2969, 5466	12	6.4	21.4 km	Yes	1 LTC and 3 LVRs, 5 switching CAP banks	62%	2 centralized, 142 distributed	6
J1	3433, 4242	12	6.3	18.1 km	Yes	1 LTC and 8 LVRs, 3 switching CAP banks	28%	2 centralized, 5 distributed	10

Additional Comments:

LVRs (or VREGs) are single-phase line voltage regulators.

LTCs are 3-phase on-load tap changers installed at the substation transformer. CAP stands for capacitor.

LVRs and LTCs have 33 taps (except for Feeder J1 where the LTC has 17 taps). Each time series profile is 1-second, yearlong.

Centralized PV are 3-phase systems rated above 100 kW.

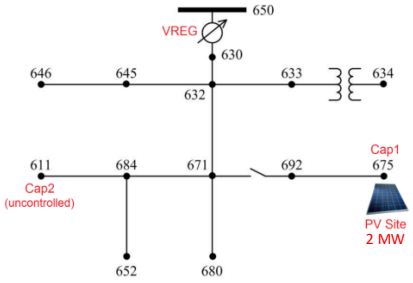
Distributed PV are single-phase systems rated below 100 kW.

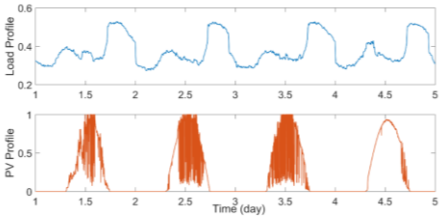
All regulating devices have delays and deadbands.

Time series data is synthesized from SCADA and on-field irradiance sensors. IEEE 123-bus reconfigures during the summer months using sectionalizing switches. Feeder CO1 and J1 are actual utility feeders located in the U.S.

3.3.1. IEEE 13 Bus Test Circuit

- Features:
  - Peak Load: 4.15 MW
  - Rated PV: Single 3ϕ, 2 MW system (unity power factor)
  - Three 1ϕ VREGs (with LDC) at Substation
  - One 3ϕ switching capacitor bank (600 kVAR)
  - One 1ϕ capacitor bank fixed (100 kVAR)
  - 1 Load profile from actual SCADA data
  - 1 PV profile based on irradiance data (NREL Oahu)
- Unique Aspects:
  - Fast oscillations between VREG and Cap1
  - PV profile with significant variability

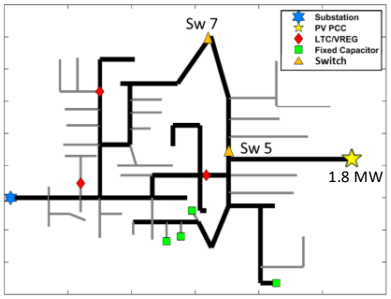


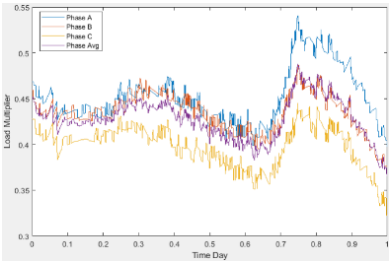


Brute-force QSTS takes around 15 minutes! (1-sec, yearlong)

3.3.2. IEEE 123 Bus Test Circuit

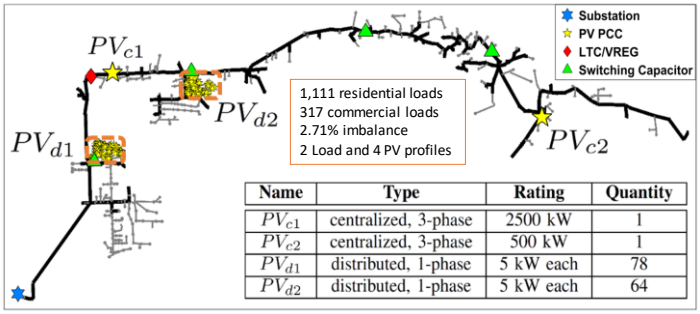
- Features:
  - Peak Load: 3.49 MW
  - Rated PV: Single 3ϕ, 1.8 MW system (-0.98 Power Factor)
  - Six 1ϕ VREGs (with LDC), One 3ϕ Substation LTC (with LDC)
  - 4 fixed capacitor banks (non controllable)
  - 4 Load profiles (by phase) from actual SCADA data
  - 1 PV profile based on irradiance data (NREL Oahu)
- Unique Aspects:
  - Loads on each phase have a unique profile
  - Multi-phase loads have a unique profile as well
  - PV system at non unity power factor
  - Feeder reconfiguration, twice a year



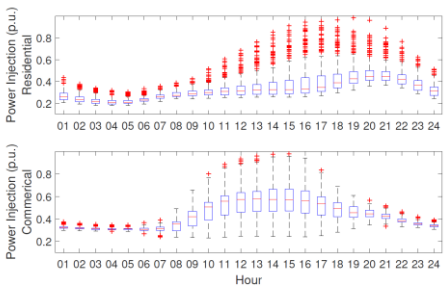


Brute-force QSTS takes around 1.5 hours! (1-sec, yearlong)

3.3.3. CO1 Test Circuit



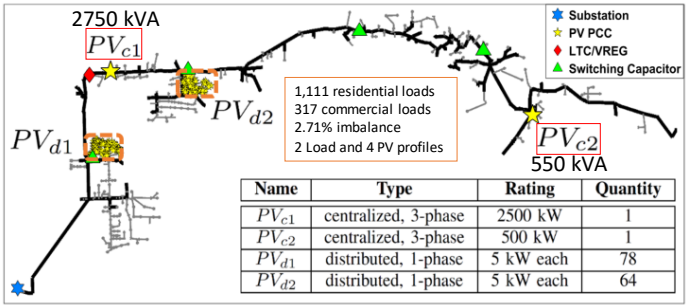
Name	Type	Rating	Quantity
PV_{c1}	centralized, 3-phase	2500 kW	1
PV_{c2}	centralized, 3-phase	500 kW	1
PV_{d1}	distributed, 1-phase	5 kW each	78
PV_{d2}	distributed, 1-phase	5 kW each	64



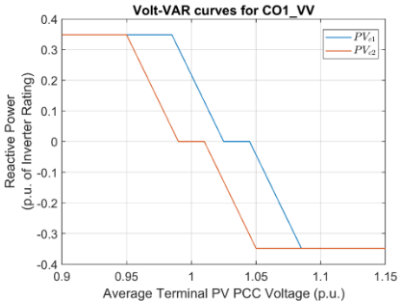
- Features:
  - 21.7 km actual utility distribution feeder with 5469 nodes
  - Peak Load: 6 MW
  - Rated PV: 3 MW (both 3ϕ and 1ϕ systems, 144 total)
  - Three 1ϕ VREGs, One 3ϕ Substation LTC
  - 5 switching capacitor banks (both 3ϕ and 1ϕ)
  - 4 PV profiles (each region) and 2 Load profiles (by customer class)
- Unique Aspects:
  - Size of feeder
  - Total number of profiles
  - Significant voltage imbalance
  - LV secondary system modeled
  - PV profiles based on location

Brute-force QSTS takes around 29 hours! (1-sec, yearlong)

3.3.4. CO1_VV Test Circuit



Name	Type	Rating	Quantity
PV_{c1}	centralized, 3-phase	2500 kW	1
PV_{c2}	centralized, 3-phase	500 kW	1
PV_{d1}	distributed, 1-phase	5 kW each	78
PV_{d2}	distributed, 1-phase	5 kW each	64



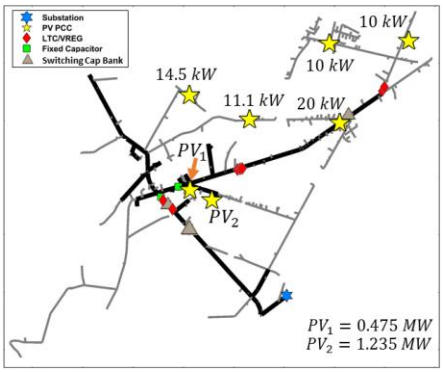
- Features:
  - 21.7 km actual utility distribution feeder with 5469 nodes
  - Peak Load: 6 MW
  - Rated PV: 3 MW (both 3ϕ and 1ϕ systems, 144 total)
  - Inverters for PV_{c1}, PV_{c2} are in Volt-VAR (VV) mode
  - Inverter turns off at less than 0.1% real power output
  - Both PV_{c1}, PV_{c2} are following a unique VV curve
- Unique Aspects:
  - Size of feeder
  - Total number of profiles
  - Significant voltage imbalance
  - LV secondary system modeled
  - VV mode for PV_{c1}, PV_{c2}
  - PV upstream/downstream of VREG

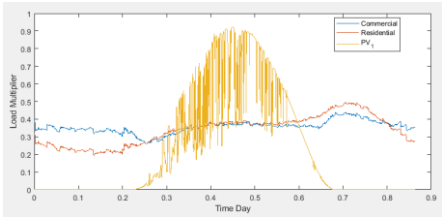
Brute-force QSTS takes around 67.5 hours! (1-sec, yearlong)

3.3.5. *EPRI J1 Test Circuit*

- Features:
  - Peak Load: 6.3 MW
  - Rated PV: 1.77 MW (both 3ϕ and 1ϕ PV system, 10 total)
  - Eight 1ϕ VREGs, One 3ϕ Substation LTC
  - 3 switching and 2 fixed capacitor banks
  - 3 Load profiles (by customer type)
  - 7 PV profile based on irradiance data
  - Actual utility distribution feeder
- Unique Aspects:
  - Size of feeder (3433 buses), LV secondary system modeled
  - Each PV system has its own unique profile
  - Multiple VREGs in series
  - Substation LTC has larger tap distance

Brute-force QSTS takes around 24.3 hours! (1-sec, yearlong)





3.4. Results

Each rapid QSTS algorithm was run on the test circuits described in the previous section. A summary of the 5 test circuits is shown in Table 7. For each circuit, a baseline simulation was run in OpenDSS at 1-second resolution for 1-year, and all QSTS accuracy metrics from Section 2.2 such as tap change and line losses were recorded.

Table 7. Summary of the QSTS test circuits.

	IEEE 13 Bus	IEEE 123 Bus	Feeder CO1	Feeder CO1_VV	J1
Buses	13	123	2969	2969	3433
PV systems	Single 3-ph (unity PF)	Single 3-ph (-0.98 PF)	144 systems (3-ph + 1-ph)	144 systems (3-ph + 1-ph)	10 systems (3-ph + 1-ph)
Total Profiles	1 Load, 1 PV	4 Load, 1 PV	2 Load, 4 PV	2 Load, 4 PV	3 Load, 7 PV
Regulating Devices	3 LVRs with LDC, 1 switching CAP bank	1 LTC and 6 LVRs with LDC	1 LTC and 3 LVRs, 5 switching CAP banks	1 LTC and 3 LVRs, 5 switching CAP banks	1 LTC and 8 LVRs, 3 switching CAP banks
Challenges	Fast regulator and cap interactions	Feeder reconfiguration, Phase profiles, non-unity PF	Size, Total profiles, LV modeled	Volt VAR for 3-ph PV systems	Size, multiple VREGs in series, LV secondary

After running the baseline brute-force simulation on each test circuit, the rapid QSTS algorithms were run to compare computational time and accuracy. Table 8 shows the speed improvement results of the rapid algorithms on each of the test circuits. If the rapid QSTS algorithm did not meet the accuracy thresholds for a given circuit, the results are not reported. Note that Table 8 does not

include results for Intelligent Sample Selection or the Machine Learning approaches. Those methods did not provide significant speed improvement and were unable to validate all QSTS metrics, so development was discontinued.

Table 8. Speed improvement (times faster than 1-second yearlong brute force simulation) for each method on different test circuits.

Algorithm	IEEE 13 Bus	IEEE 123 Bus	Feeder CO1	Feeder CO1_VV	J1
Event-Based	100x	288x	300x	111x	115x
Variable Time-Step	41x	-	53x	56x	42x
Vector Quantization	243x	-	169x	243x	-
Temporal Parallelization in OpenDSS with 10 Cores	1.5x	5.4x	6.5x	6.2x	5.7x

The results in Table 8 show that the Event-Based and Vector Quantization algorithms had the greatest speed improvements on each of the test circuits while maintaining accuracy within the predetermined limits. The results also reveal that the Variable Time-Step and Temporal Parallelization algorithms had a more consistent performance across the test circuits compared to Event-Based and Vector Quantization. This behavior is expected as each algorithm has its own advantages and limitations—some of which were showcased by the challenges encompassed in the diversity of test circuits. Specifically, the algorithms each perform differently on Feeder CO1 compared to Feeder CO1_VV; some performed better in the presence of the advanced inverter controls in the latter test circuit, while others performed worse. In the case of Vector Quantization, the volt-var inverter control results in individual power flow times taking longer, so there is higher speed improvement of bypassing the power flow with the vector quantization lookup table. Conversely, in the Event-Based simulation, the advanced inverter control actions caused the algorithm to re-linearize more often with the increased number of inverter states, reducing the simulation speed. Similarly, the IEEE 13 Bus circuit has fast interactions between regulating devices, which led to analogous effects on the performance of the Vector Quantization and Event-Based algorithms.

The performance of the Variable Time-Step algorithm is affected in a slightly different manner across the test circuits. The backtracking solver in this algorithm is forced to rewind the simulation and solve the time points in between at a higher resolution whenever the control queue is populated by a pending action from a regulating device. Therefore, the speed of Variable Time-Step is reduced as the number of backtracking events increases. In OpenDSS, control actions taken by an advanced inverter do not appear in the control queue as there are no time delays associated with them, so those actions would not initiate a backtracking event. Furthermore, since the advanced inverters are regulating voltage at a faster rate, some control actions from other regulating devices like LVRs or switching capacitor banks may be avoided. It also follows that the likelihood of a backtracking event increases with the number of regulating devices in the circuit. Considering the behavior of the

backtracking solver, it makes sense that the performance of Variable Time-Step is improved from Feeder CO1 to Feeder CO1_VV and reduced from Feeder CO1 to IEEE 13 and J1.

For Temporal Parallelization, the speed improvement is dependent upon both the characteristics of the circuit and the available computing resources. QSTS simulations contain both "parallelizable" components (e.g. power flow solutions) and "serial" components (e.g. assigning a time-series multiplier to a load). Parallelizable components can be independently handled by multiple processors, whereas the serial components require access to shared resources like memory or caches. Therefore, the serial components of QSTS simulations can cause bottlenecks that ultimately affect the speed of the parallelized simulation. For similar reasons, the size and complexity of the circuit also have a significant impact on the performance of parallelization. Larger circuits with longer computation times for power flow solutions typically have better speed improvements because the processors are spending a higher percentage of time performing computations compared to the time spent waiting for shared resources. In other words, larger circuits typically have a smaller serial component, which is more suitable for parallelization. Table 8 shows that Temporal Parallelization performed better on Feeder CO1 than on J1, even though the latter circuit has more buses. While the results are close, one possible explanation is that J1 has 10 time-series profiles compared to the 6 in FeederCO1, leading to a larger serial component.

While all results are within the predetermined error limits in Section 2.2, it should be noted that each algorithm has a different range of errors produced. For example, variable time-step and vector quantization algorithms contain variables to adjust thresholds for improved speed vs. accuracy. The results above for variable time-step have been tuned to produce the maximum speed just within the error thresholds. On the other hand, event-based simulation and temporal parallelization produce very consistent results with little error. See the references under each algorithm for exact errors expected for the algorithms on the test circuits.

Table 9. Expected speed improvement (times faster than 1-second yearlong brute force simulation) for circuit reduction and Diakoptics on different test circuits

Test Circuit	OpenDSS Reduction	GridPV Reduction	Diakoptics (15 Sub-circuits)
IEEE 13 Bus	1.2x	2.0x	-
IEEE 123 Bus	1.8x	6.5x	6.46x
Feeder CO1	1.3x	296.9x	9.23x
Feeder CO1_VV	1.3x	296.9x	9.23x
J1	7.6x	137.3x	8.37x

Due to the challenge of accurately capturing all necessary simulation metrics when using circuit reduction and Diakoptics, these algorithms were omitted from Table 8. However, Table 9 provides an estimate of speed improvements for circuit reduction and Diakoptics on each test circuit. In Figure 10, it was shown that the computational time of a QSTS simulation is proportional to the number of nodes in the circuit model. Therefore, we can estimate the speed improvement by comparing the number of nodes before and after each algorithm is implemented. For Diakoptics, the sub-circuit with the largest number of nodes was used for the calculation.

For the OpenDSS reduction column in Table 9, the *reduce* function was used with both the *laterals* and *stubs* options and was called repeatedly until no reduction was observed. While performing circuit reduction directly in OpenDSS is fast and convenient, the level of reduction is rather limited compared to the GridPV implementation. For Diakoptics, the results show that while there is room for further improvement (in theory, up to 15x faster with 15 sub-circuits), the circuit tearing algorithm was successful and can bring about considerable speed improvements. In general, the results in Table 9 indicate that both circuit reduction and Diakoptics are better suited to larger circuits (e.g. Diakoptics did not work on the IEEE 13 Bus circuit).

3.5. Combination of Methods

In general, the total solution time for a QSTS simulation has three main components: the computational time for a single power flow solution (t_{PF}), the total number of power flows solutions (N_{PF}), and the computational power (CP). Figure 26 describes the formula for the QSTS time and Figure 27 summarizes the relationship between these components and the total QSTS solution time while categorizing each rapid QSTS method based on the component it aims to address. Being that each rapid QSTS method takes a unique approach to reducing the total computational time, it is possible that further speed improvements could be realized through a combination of methods either within or across the categories. The ideal outcome of combining any two rapid QSTS methods would be a multiplication of their individual speed improvements and for any compounding errors to remain within the predetermined thresholds for all metrics.

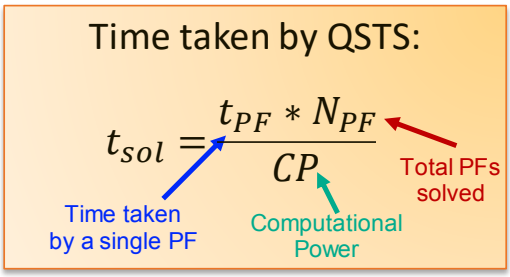


Figure 26. (Same as Figure 16—repeated for easy reference) Total QSTS computational time is a combination of the time per power flow (PF) multiplied with the total number of PF, divided by the computational power. Each type of rapid QSTS algorithm is trying to address one of these main pieces of the computational time.

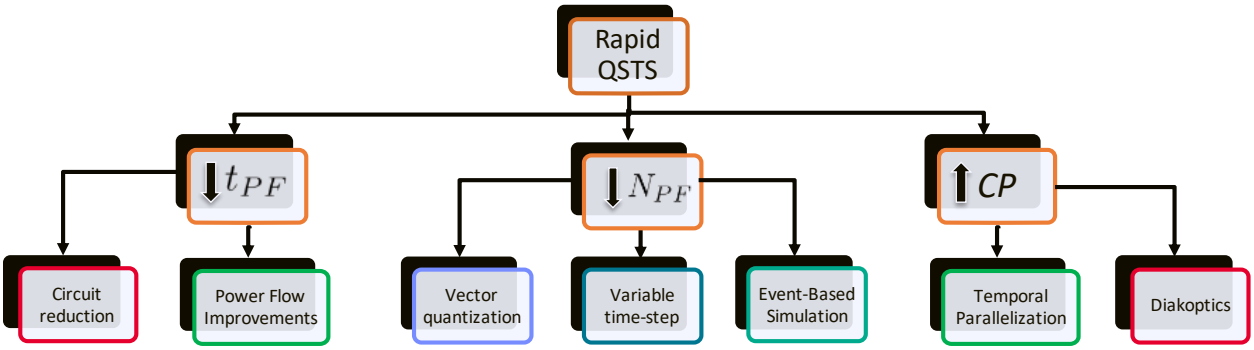


Figure 27. Categories of Rapid QSTS Algorithms (same categorization as Figure 17 but presented in a slightly different format).

One possible combination within a single category is between the Vector Quantization and Variable Time Step algorithms. This combination was explored in [81]. First, Vector Quantization was implemented in combination with the predetermined time step (PT) solver [98], which would allow

the algorithm to fast-forward in time until the load or PV value had significant variation. However, the total number of computed power flows over the time horizon was identical to that of the Vector Quantization algorithm alone, providing only a negligible reduction in the computational time of the matrix indexing logic. Vector Quantization was also implemented in combination with the back-tracking variable time step algorithm (VTS-BD), where large time-steps are taken until a system state change (i.e. pending control action) is detected. The solver then backtracks to the previous large time-step instance and proceeds with a small time-step for a period of time until pending control actions are complete and no pending action is detected, so the large time-step is resumed [76]. The Vector Quantization subroutine was utilized in place of solving power flows in both the coarse simulation and the backtracking processes. Table 10 shows the detailed results of this combination on runtime reduction and accuracy of capturing control events.

Table 10. Runtime Reduction and Accuracy for Combination of VQ and VTS-BD

VTS Large Time-step	VQ # of Cluster	VTS Small Time-step	# of Unique PF Solutions	# of PF Reduction	# of Tap Changes Reg1 / Reg2 / Reg3	# of Cap Switching	MAPE
	Base Case		31536000	0.00%	7056 / 7233 / 8460	2506	0.00%
15	-	1	1810835	94.26%	0.00% / 0.00% / 0.00%	0.00%	0.00%
-	201	-	212469	99.33%	-0.74% / -0.57% / -0.85%	-0.72%	0.72%
15	201	1	160316	99.49%	-0.74% / -0.57% / -0.85%	-0.72%	0.72%
15	201	2	150582	99.52%	0.68% / 0.37% / 0.09%	0.80%	0.49%
15	201	4	142737	99.55%	3.29% / 3.14% / 3.19%	6.38%	4.00%

Overall, the combined solution approach of VQ and VTS-BD provided a speed improvement of about 33% and an error increase of 3.28% compared to VQ alone. So, while these results do reveal some potential for combining rapid QSTS methods from within a single category, the magnitude of improvement is far from optimal compared to multiplying the speed improvements of both methods together.

Similar results are expected from other combinations within a single category, like in the case of increasing computational power (*CP*). The Diakoptics algorithm works by "tearing" a circuit into multiple sections, solving the power flow of each section on a separate processor, and stitching the results back together. For Diakoptics to even be considered, the circuit must be fairly large for the tearing algorithm to yield proper results. Assuming the circuit is large enough, there also exists an optimal number of torn sections that results in the best speed improvement. For Temporal Parallelization, speed improvement generally increases with the number of processors being used, but the magnitude of improvement diminishes with each additional processor [94]. Therefore, it is conceivable that, given enough processors, some combination of Diakoptics and Temporal Parallelization exists that outperforms each individual algorithm using that same number of processors. However, the speed improvement in this scenario is still likely to fall short of an ideal combination.

The other possibility is to explore the combination of methods from across the categories in Figure 27. Suitable combinations can be narrowed down by qualitatively analyzing the potential interactions between any two algorithms. For instance, we know that the Diakoptics algorithm requires a large circuit to work properly, so Temporal Parallelization would be a better fit when combined with Circuit Reduction. Following this approach and considering the degree of difficulty in implementation, six distinct combinations were selected as candidates with a potential for drastic speed improvements. Figure 28 shows each of these combinations and their corresponding categories, where each combination is represented by a unique color. For example, the green line

represents the combination of the Circuit Reduction and Event-Based algorithms, whereas the red line represents the combination of the Event-Based and Diakoptics algorithms.

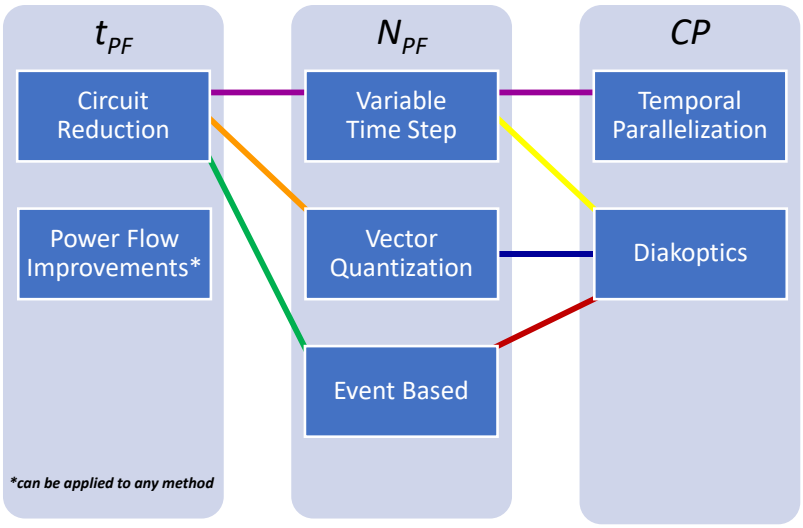


Figure 28. Potential combinations of rapid QSTS methods

In Figure 28, it can be observed that Variable Time Step is the only method from the N_{PF} category that is combined with Temporal Parallelization. As such, Variable Time Step is also the only method shown to be combined with an algorithm from each category. The reason for this is that the Vector Quantization and Event-Based algorithms both rely, in part, on a look-up table of familiar system states to reduce the number of power flow solutions or other computations. As time progresses, the probability of encountering a familiar state increases. When combined with Temporal Parallelization, each divided section must have the ability to read and write to the look-up table for the benefits of VQ and EB to be preserved. While this feature can be incorporated by adjusting the algorithms to communicate with a centralized database, adding this capability increases the difficulty of implementation. In contrast, Variable Time Step can be directly implemented with Temporal Parallelization. Being that Circuit Reduction is generally more compatible with Temporal Parallelization than Diakoptics, both VQ and EB are limited to a combination with either Circuit Reduction or Diakoptics, but not both simultaneously.

To reiterate, Figure 28 does not represent an exhaustive comparison, but rather contains a handful of suggested combinations expected to provide a significant speed improvement based on a qualitative analysis of potential interactions between multiple algorithms. Also, the suggested combinations are based on the current applications of QSTS simulations, which are typically performed on a single circuit out of a single substation. However, if a model contains a network of substations with multiple circuits and hundreds of thousands of buses, the combination of Circuit Reduction and Diakoptics becomes a more viable option. Therefore, combinations outside of the ones shown in Figure 28 should not be permanently ruled out.

The performance of any individual rapid QSTS method, in terms of runtime and accuracy of all simulation metrics, is also affected by the number of buses, the number of controllable devices, and the number of time-series profiles in the circuit model. So, aside from the challenge of combining multiple methods from an algorithmic standpoint, other impacts must also be considered, like the increased difficulty of quantifying simulation metrics and/or the compounding of errors introduced by different methods. The degree to which these issues would inhibit adopting a combinatorial approach depends on both the underlying circuit and the application of the QSTS simulation.

4. GENERATING DATA FOR QSTS

Previous sections have discussed methods to increase QSTS simulation speed while maintaining simulation results that closely match full-fidelity (long-runtime) simulations. The accuracy of the resulting simulations, though, is additionally dependent on the accuracy of the input data.

4.1. Introduction

While distribution system components (lines, transformers, etc.) are typically well-defined in utility models and generally are static (i.e., line and transformer settings do not change with time), customer loads and PV generation are both temporally dependent. This means that load and PV introduce variability into the system that must be considered. As discussed below, load is often measured at low temporal resolution and only at the feeder head. PV production is often not measured at all by the utility. Thus, to ensure accurate QSTS studies, there is a need for accurate load and PV data generation with high spatial and temporal resolution.

4.2. Load

In the past, for QSTS simulation, the load at every node in a distribution system would likely be populated with the same load shape as the substation, with just magnitude differences introduced by distribution factors. However, the load profiles for the nodes in a distribution system are typically diversified with different patterns and variabilities as the load profiles are driven by customer use patterns and the specific end-use loads present in each building. Therefore, representing all the nodes in a distribution system with only one substation load shape cannot effectively capture the operation characteristics of the system through the use of QSTS simulation techniques. With the advent of smart meters, some of the utilities have some measurement data at the single customer level. However, measured smart meter data typically cannot be directly used for simulation, testing, and algorithm development due to the relatively low resolution of the smart meter data (typically 15-minute energy usage data at best) and concerns about customer privacy. Thus, in order to enable accurate QSTS simulations load models were developed that transformed low temporal resolution data into higher resolution data via a load variability model. Further, a load diversity model was developed to tune the diversity of loads on a distribution feeder as these load diversity metrics are typically tracked by utilities for other planning purposes. These two models are described below. Full descriptions of the techniques used are available in [99] [100].

4.2.1. Variability and diversity modeling approach

The overall modeling approach is summarized in Figure 29. The modeled load profile on each node is composed of four major parts: 1) the diversity model extracted from the diversity library, 2) the variability model extracted from variability library, 3) the substation load profile, 4) the distribution factor for the particular node. Descriptions of how the variability and diversity libraries are developed are given below.

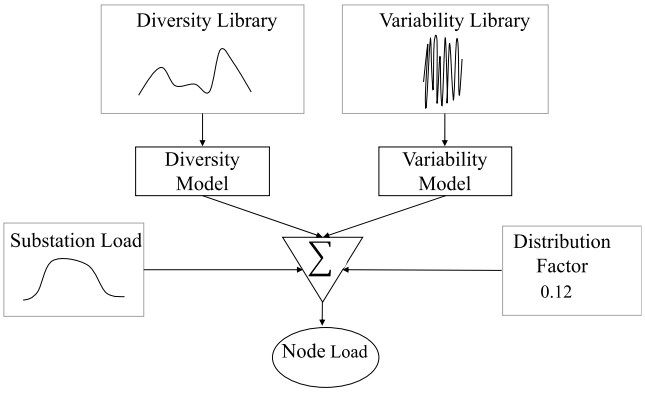


Figure 29. Overview of load variability and diversity modeling approach.

As shown in Table 11, the modeling flow is described as three steps: first, the substation load profile is scaled by the distribution factor for the node (i.e. how much overall load is represented by that particular node); second, the extracted diversity model will be applied onto the scaled substation load profile, now the scaled substation load profile will have the certain load pattern defined by the diversity model. Finally, the variability model extracted from the variability library will be employed to create the detailed load model for this node. [101]

Table 11. Steps for Variability and Diversity Load Models.

Step 1	Scale substation load profile by distribution factor
Step 2	Extract diversity model from library and apply to scaled substation load profile
Step 3	Extract variability model from library and apply to the profile with diversity applied

4.2.2. Load variability modeling results

Multi-resolution discrete wavelet transforms (DWT) are used to extract the variability from actual, field-collected load data. The idea here is that utilities would collect limited amounts of high-resolution load data (e.g. 1-second load monitoring) and use this data to populate a variability library that can then be used for the modeling of entire classes of service within their service territory. The high-frequency variabilities of the data are extracted out and used to build the variability library. Details of the construction of the variability library are introduced in [101].

The general workflow of extracting the variability is described as follows: As shown in Figure 30, input data will be decomposed into detailed and approximate coefficients. a1, a2, and a3 represent approximate coefficients, which are obtained by low pass filter and contain the low-frequency signal. d1, d2, and d3 represent detailed coefficients, which are obtained by a high pass filter and contain the high-frequency signal. As shown in (1), the summation of all the detailed coefficients and the approximate coefficient from the last level of decomposition will reconstruct the original signal. The coefficients which contain the high-frequency variability will be retained and be used to build the variability library. Lower frequency detailed coefficients represent lower speed variability components which are effectively modeled via other means such as the inclusion of 15-minute average energy usage data from smart meters.

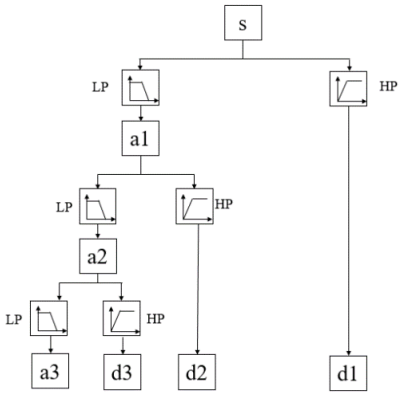


Figure 30. DWT separation of input load data into wavelets.

The variability library can be used for the creation of an infinite amount of synthetic load variability load profiles by simply shifting individual, or sets of, wavelet coefficients in time. The variability library is setup to provide different variability depending on the time of day and the customer class as these appear to be the main drivers of the expected load variability present at a distribution transformer level. Further, the obfuscation of the actual load data, which provides the data necessary to populate the variability library, through the DWT and time-shifting makes the synthetic data non-sensitive from a privacy standpoint but still allows a reasonably accurate data-driven model.

An example visual representation of the DWT coefficients resulting from the variability modeling is shown in Figure 31. The original measured load profile is shown on the top line and the decomposition of the signal into the various relevant frequency ranges is shown. Lower number coefficients are used to model load variability (e.g. the natural turning on and off of an air conditioner) whereas the higher coefficients can be modeled instead by simple noise models. This reduction of the DWT coefficients necessary to implement the load variability model greatly reduced the data requirements for implementing the model and sped synthetic load generation time.

Example synthetic load data generated using the load variability model are shown in Figure 32 with the original dataset, used to develop the load variability library, shown for reference. In this case, there is no expectation that the two load profiles match. In fact, they should not match as one of the features of the load variability model is the ability to use measured data to populate the variability library while also not restricting the use, due to privacy concerns, of the synthetic load profiles generated. As can be observed, the variability during these example 4-hour periods of time during the Spring, Fall, Summer, and Winter look reasonable. To validate the synthetic time-series load models an investigation into the distribution of load ramp rates (up and down) was completed. Figure 33 shows the ramp rate distributions for both the original, measured, load data used to populate the load variability library as well as the ramp rate distributions for the synthetic load profiles generated with the load variability model. The distributions match quite well providing confidence that the load variability model is generating load profiles that have a realistic amount of load variability.

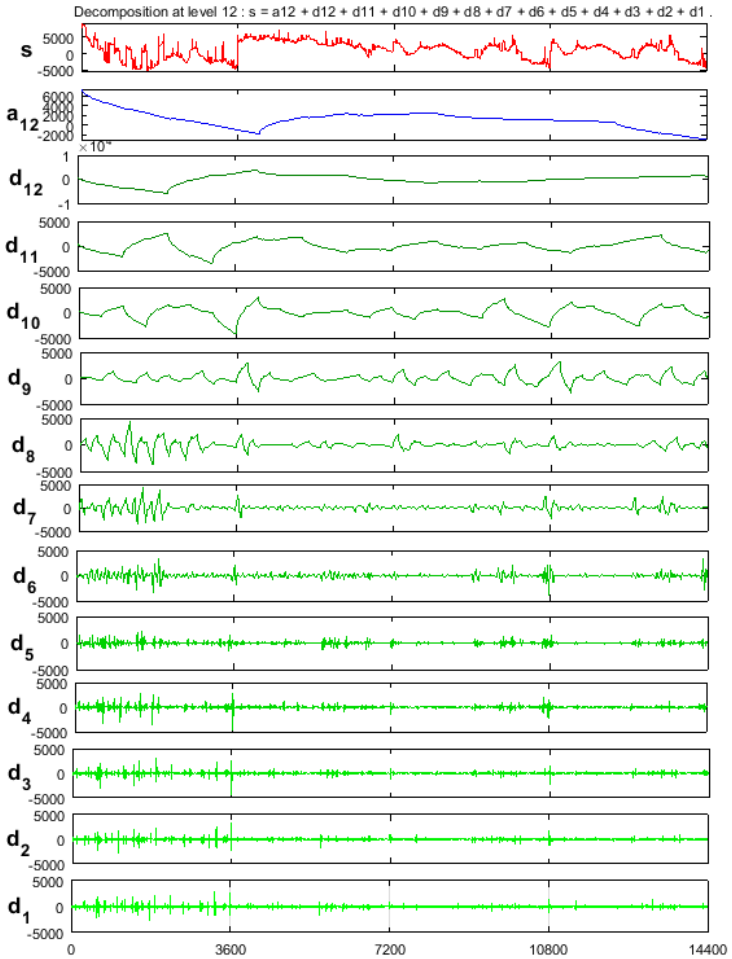


Figure 31. Example original load profile decomposed into DWT coefficients (4 hours in total).

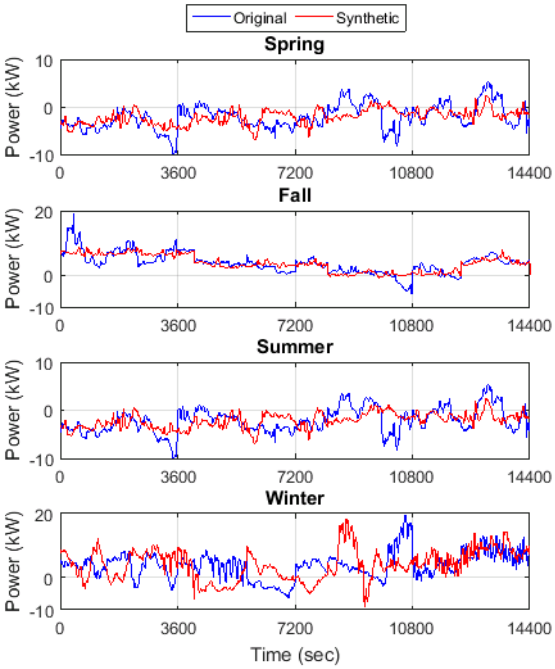


Figure 32. Example original and synthetic data set for the four seasons developed.

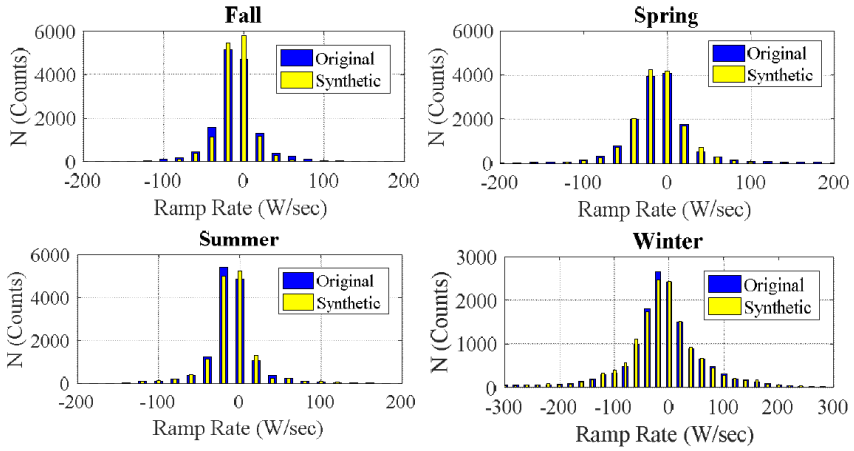


Figure 33. Histogram comparison of the original measured load data and synthetically developed load variability data.

4.2.3. Load diversity modeling results

A certain amount of load diversity naturally resulted from the load variability modeling approach described above but for instances where 15-minute AMI/smart meter data was not available the typical modeling approach of scaling a substation load profile by load factor to each individual load on a circuit resulted in load profiles that were too similar (i.e. low diversity). Thus, a load diversity modeling approach was developed to add load diversity to such load profiles in addition to load variability via the modeling approach described above.

Figure 34 shows the basic steps taken to extract load patterns, used to populate the load diversity library, from the field-measured data. The data, in this case, needs to be collected concurrently across multiple loads on the same circuit (or at least within the same customer class and in the same general region). Hour-long periods of load data are clustered, using k-means techniques, in order to develop a relatively small set of potential load diversity profiles or lower frequency load patterns. The center profile from each cluster is selected as being representative of that load pattern cluster and used to build the load diversity library.

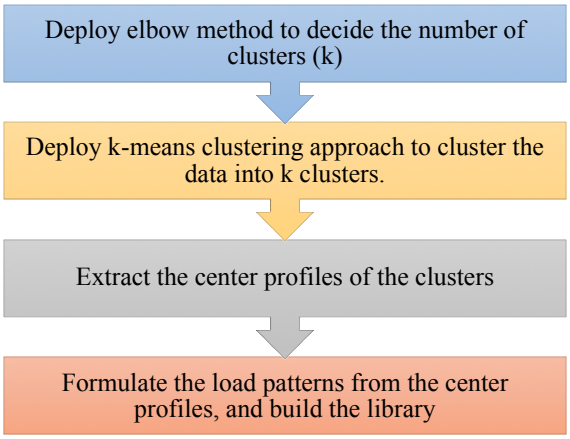


Figure 34. Load pattern extraction for the load diversity library.

Figure 35 shows three example load patterns that belong to the same cluster. Even though these real load profiles are from different loads it is easy to see why they were clustered together as they have the same general shape.

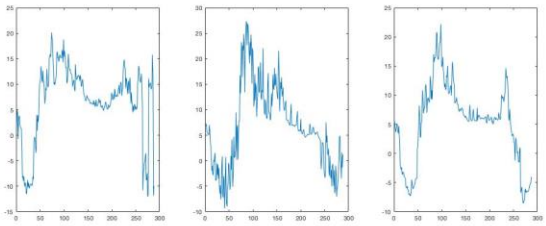


Figure 35. Examples of load patterns in a single cluster.

After clustering the center of the cluster load patterns is formulated into a load pattern. The concept is that only load diversity is modeled and the addition of load variability at this step is undesirable. Thus, as is shown in Figure 35 the load pattern is generated by heavy quantization of the original load profile.

a) b)

Figure 36. Load patterns: a) original load profile, b) the generated load pattern for the diversity library.

The results of the combined load diversity and load variability models applied to three loads is shown in Figure 37. In all cases and at every point in time, the sum of the load profiles is equal to the originally measured substation load profile. Even with this constraint, the load profiles with both diversity and variability modeled appear more realistic than when loads are modeled using just the substation load profile.

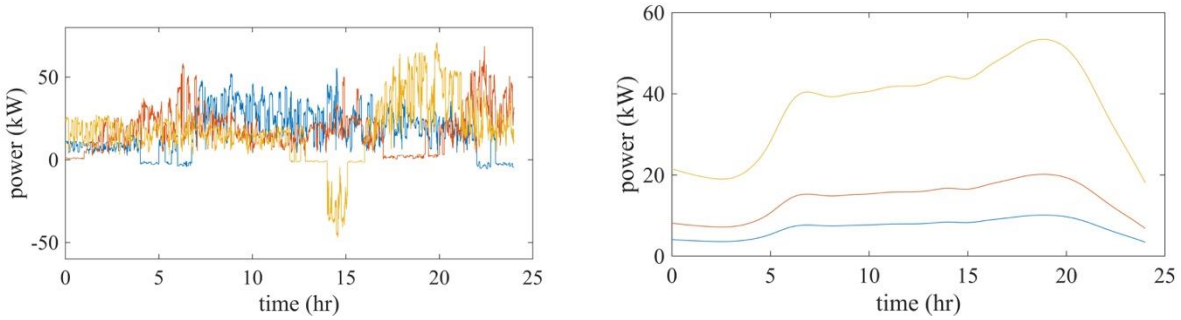


Figure 37. Example load profiles a) with diversity and load variability modeled and b) the same load profiles if they are scaled versions of the substation load profile.

An investigation into the improved QSTS modeling capability was undertaken to provide a baseline for how the diversity and variability load models might impact typically DER interconnection planning studies. Table 12 shows the maximum and minimum voltages experienced on the IEEE 123-bus Test Feeder for a month of temporal analysis using various QSTS temporal resolutions (time-steps) and when modeling diversified loads vs. plain loads (meaning loads allocated via scaling of the substation load profile. These results indicate that at lower QSTS resolutions the study differences between using diversified loads and plain loads are relatively small. This should be expected because lower temporal resolution QSTS is effectively filtering the diversified loads profile, via low-resolution sampling, thus only the load diversity effects, not the load variability model effects, remain. Even at these lower QSTS temporal resolutions, there are some differences because each load on the circuit does not have the same load profile thus the highest and lowest voltages present on the circuit increase and decrease respectively. At lower QSTS temporal resolution (e.g. 1-second resolution) the differences in study results are considerable with differences in voltage being in the range of 1.5-2% p.u. This sort of difference is significant considering the entire standard voltage range is ± 5% p.u.

Table 12. Max/min voltage comparison using diversified loads vs. plain loads for multiple QSTS time resolutions using the IEEE 123-bus test feeder.

		Maximum Voltage (p.u.)	Minimum Voltage (p.u.)	Maximum Ramp (p.u./ Δt)
1-second	Diversified Loads	1.0338	0.9379	0.0258
	Plain Loads	1.0183	0.9720	2.9802×10^{-6}
1-minute	Diversified Loads	1.0331	0.9398	0.0258
	Plain Loads	1.0182	0.9720	1.4764×10^{-4}
10-minute	Diversified Loads	1.0299	0.9479	0.0246
	Plain Loads	1.0180	0.9720	0.0015
30-minute	Diversified Loads	1.0286	0.9495	0.0151
	Plain Loads	1.0179	0.9720	0.0042

4.3. Irradiance

High-frequency solar variability is an important input to accurate distribution grid integration studies. Using low-frequency solar variability results in an underestimation of the impact of solar photovoltaics (PV) on distribution grid operations, as the low resolution of the PV hides its high-frequency variability. Underestimations of voltage regulator tap change operations of up 20-70% were found when using 15-minute solar variability instead of 30-second solar variability [66].

However, measurements of high-frequency solar variability are scarce. Sandia has collected a database of 10 high-frequency (30-seconds or better) irradiance measurements, mostly in the western United States. Separately, NREL has developed satellite-derived irradiance variability samples with resolution up to 4-seconds [102] and availability across the United States.

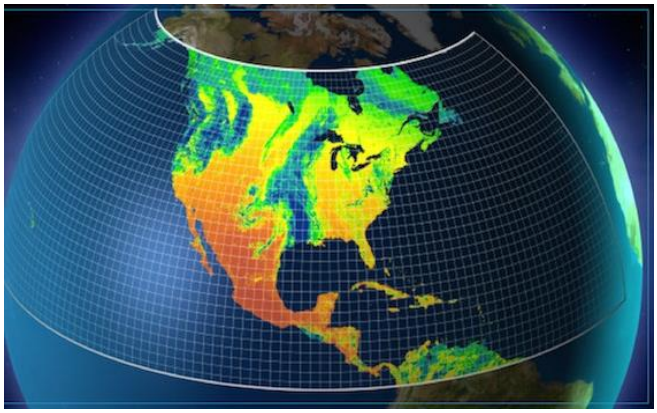


Figure 38: NSRDB coverage [103].

Work on irradiance modeling within this project had two components: 1) the temporal downscaling of nationally available data to provide single point irradiance relevant to distribution-level studies and 2) to develop a model for the spatial relationship of multiple PV systems on the same distribution circuit which was accomplished using cloud field modeling techniques.

4.3.1. Temporal downscaling

The NREL-developed high-resolution irradiance algorithm (HRIA) is capable of producing irradiance samples at up to 4-second resolution [102]. 4-second samples are produced based on low-frequency, satellite-derived irradiance available through the National Solar Radiation Database (NSRDB) [103]. As can be seen in Figure 38, NSRDB measurements, and, hence HRIA 4-second samples, are available for most of North America. NSRDB irradiance is resolved on a 4 by 4-km grid.

The HRIA predicts the temporal variability for a satellite-derived irradiance pixel using two steps. The first step is called the SIND Method followed by Further Downscaling as described below.

4.3.1.1. SIND Method (30-minute to 1-minute)

First, the method used for the Solar Integration National Database (termed the "SIND" method) [104] is used to downscale satellite-derived irradiance to 1-minute resolution. This is done using a spatial "patch" of satellite data points: both the pixel containing the location of interest plus several surrounding pixels are used to determine the "Class" of solar variability. Classes roughly range from low variability to high variability but can also have features related to changing sky conditions (e.g., clear to cloudy). Once a class is selected, it is used to model the 1-minute solar variability. Figure 39 gives an overview of the process used to create 1-minute resolution samples.

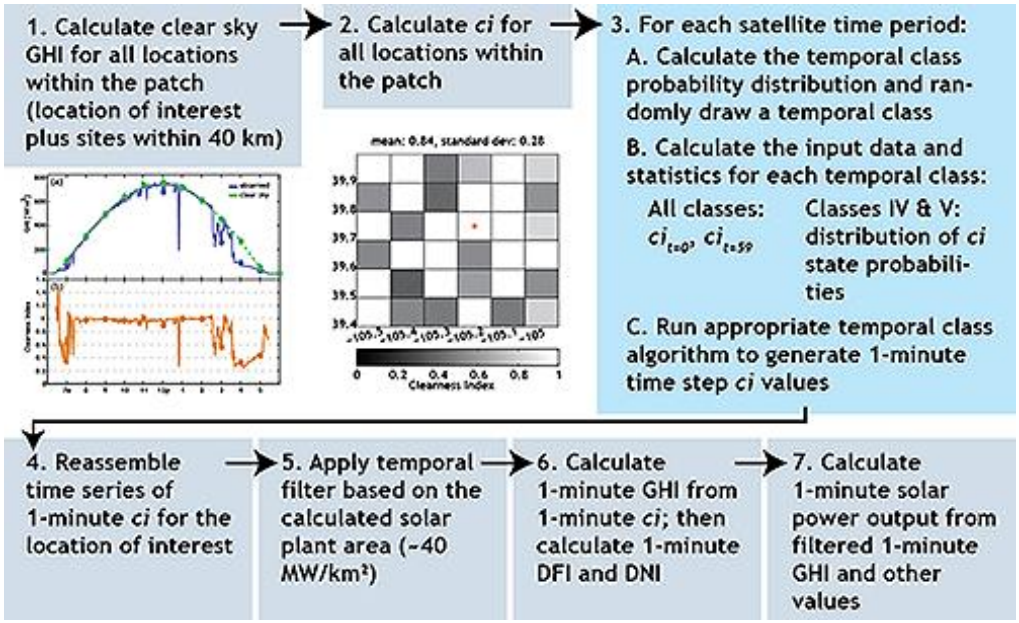


Figure 39: SIND method flowchart, showing how 30-minute satellite data is downscaled to 1-minute resolution [104]. The method uses the cloud index (ci) to classify the irradiance in each satellite pixel.

4.3.1.2. Further Downscaling (1-minute to 4 second)

Second, the 1-minute samples are further downscaled to 4-seconds using an extension of the Fourier transform [105], as shown in Figure 40. For each class of sky conditions (as defined in the SIND method), a library was assembled of 1-second ground measurements from Oahu, Hawaii [106]. The average Fourier power content was found for each class and then used to fill in the Fourier spectrum in the 1-minute to 4-second range. 4-second HRIA samples were then created using an inverse Fourier transform.

As considerable effort had been expended developing the HRIA, including manual classification of 1000s of hours of irradiance data into different classes of variability, the first work to potentially leverage the existing methods to use at the distribution scale was to validate the existing method.

4.3.1.3. Validation

Due to the method used to create them, the satellite-derived high-frequency irradiance samples are not expected to exactly match the ground measured irradiances. Specifically, the timing of clouds may not match between satellite-downscaled and ground-measured irradiance variability. Instead, it is important that the overall irradiance variability statistics are captured by the HRIA model. Thus, direct comparisons using traditional evaluation metrics (e.g., RMSE) which compare measurements at the exact same timestamp are not appropriate.

Since the variability samples are most likely to be used to understand the relative impact of solar variability on electric grid operations, comparison metrics that evaluate the variability over a longer period of time such as a day, month, or year, are more appropriate [107].

One such metric that will be used for comparison is the daily variability score [66]. The variability score is a way to quantify solar variability: variability scores are low (0 to 10) for clear conditions that have low variability and large (>100) for highly variable conditions. Two data samples that have the

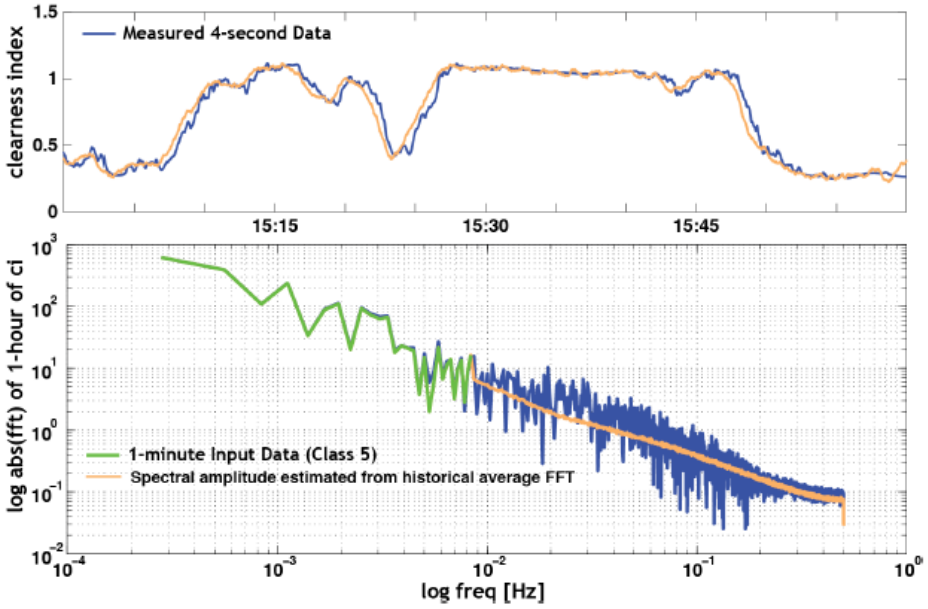


Figure 40: [Top] Clearness index samples: blue measured, orange modeled with HRIA. [Bottom] Fourier transforms: green 1-minute SIND data, blue 1-second measured, orange modeled with HRIA based on average of library samples [105].

same variability score have similar solar variability. The variability score is the maximum value of the quantity ramp rate magnitude (RR_0, expressed in % of 1000 Wm^{-2}) times ramp rate probability, multiplied by 100 to give an easier to interpret number:

$$VS_{RRdist} = 100 \times \max\left[RR_0 \times P(|RR| > RR_0)\right] \quad (1)$$

Here, all comparisons are done at 30-second resolution. Since some of the samples in Sandia's database were collected at 3-second resolution and so do not have identical timestamps as the HRIA data, 30-second averages are a fairer comparison. Temporal sensitivity analysis [65] shows at most around 3% errors in distribution grid simulations when using 30-second irradiance data instead of higher-frequency, so 30-second comparisons are sufficient for this analysis.

In the initial comparison, days were separated into clear and cloudy with separate analysis of each. Based on previous experience with the VS [66], values VS<10 are typically clear days. Thus, days when VS<10 were classified as clear and days when VS>10 were classified as cloudy.

In general, on clear days the HRIA variability score (VS) is similar to the ground VS. As seen in Figure 41, on clear days the HRIA produces VS values that are close to the ground VS values. In other words, the HRIA does not produce extremely variable days when the ground measurements indicated clear days.

However, a trend is seen in Figure 41 whereby the HRIA VS almost always exceeds the ground VS (i.e., there are more points above the 1:1 line than below).

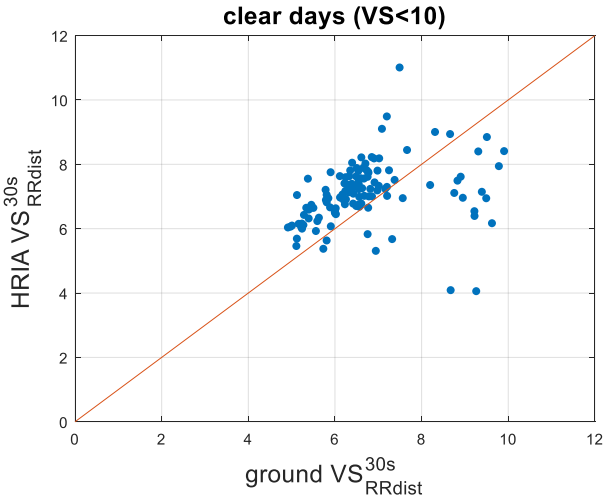


Figure 41: Scatter plot of HRIA 30-second variability score (y-axis) versus ground 30-second variability score (x-axis) on clear days during the year 2013 in Albuquerque, NM.

Figure 42 shows ground measurements and HRIA simulations on a clear day. Even though the HRIA follows the general clear-sky shape, it has some variability that is not reflected in the ground measurements. This is the reason why HRIA VS values are slightly higher on clear days than ground VS values: the HRIA is adding a small amount of variability, even on fully clear days. This is likely caused by the use of the average Fourier transform for each class, as described in Section 0

A second observation from Figure 42 is that the HRIA simulated irradiance exceeds the clear-sky values at certain times (e.g., around 13:00). Irradiance should only exceed clear-sky values when nearby clouds provide reflections, termed cloud enhancement. On fully clear days such as the one shown in Figure 42, there will be no clouds nearby and so clear-sky values should not be exceeded by as much as they are in the HRIA simulation.

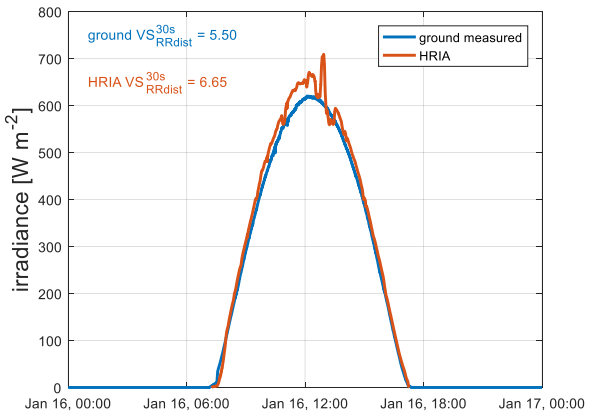


Figure 42: Plot of ground measurements (blue) and HRIA simulated (red) time-series on a clear day: January 16th, 2013 in Albuquerque, NM. 30-second variability scores are also included in the top left.

Because of both the slight overestimation of variability during clear conditions and the exceedance of clear-sky values, a possible modification to the HRIA would be to simply assume a clear-sky model when the HRIA predicts a fully clear day. However, care should be exercised to make sure

this method does not then underestimate the variability. Mostly clear days with short variable periods might be predicted to be fully clear, and hence the variability is underestimated. A mixed statistical approach whereby e.g., 90% of clear days are fully clear and assigned clear-sky values while 10% of clear days are created using the current HRIA method with small amounts of variability added could also be investigated.

On some cloudy days, the HRIA was found to underestimate the high-frequency irradiance variability. A highly variable day is shown in the top plot of Figure 43. On this day, while the HRIA captured the basic trends in the ground data (e.g., the reduced irradiance around 08:00), but it did not match the many high-frequency up and down ramps seen in the ground data. Thus, the HRIA VS (10) did not match the large ground VS (136).

A partial explanation for this variability underestimation may be the spatial averaging implicit in the HRIA samples. The training library used for the 4-second algorithms is based on the average of 18 point sensors in Oahu. Thus, there was inherent spatial variability smoothing. The SIND method may also suffer from this inherent smoothing,

To show the impact of spatial smoothing, the bottom plot in Figure 43 compares the HRIA sample to a smoothed version of the ground sample that was smoothed using the wavelet variability model [62] to represent the spatial average over the area covered by the 18 point sensors in Oahu. The VS of this smoothed sample (31) is closer to that of the HRIA sample, but the HRIA sample still underestimates the variability when compared to this smoothed sample. Thus, spatial smoothing is likely only a partial explanation of the underestimation.

Since the HRIA is based on two methods of downscaling the satellite irradiance – the SIND method from 30-minutes to 1-minute and the further Fourier downscaling to 4-seconds – the variability underestimation could be caused by one or both methods. For example, if the 1-minute data does not have sufficient variability, the 4-second data will also not have sufficient variability, regardless of the ability of the 4-second algorithm to accurately downscale from 1-minute to 4-seconds. Conversely, if the 1-minute data does have sufficient variability, errors may be in the 4-second algorithm.

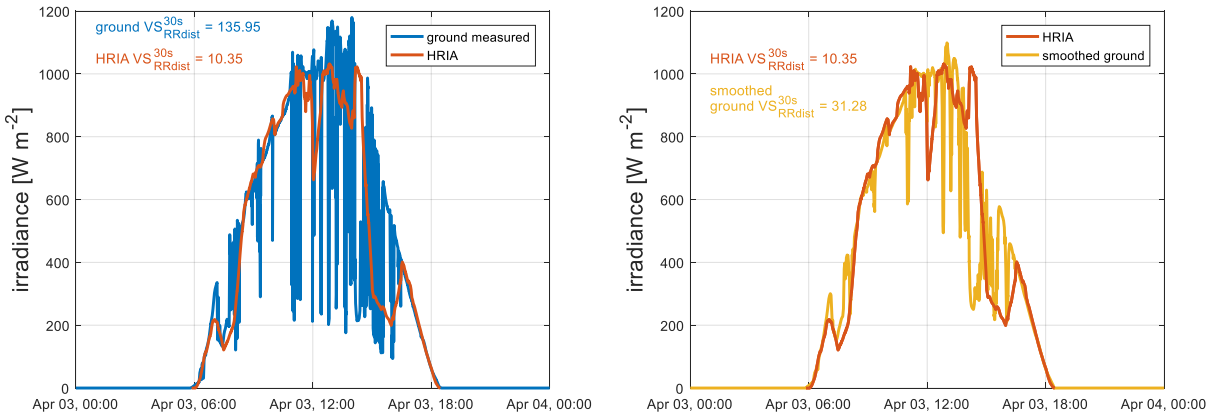


Figure 43: [Top] Plot of ground measurements (blue) and HRIA simulated (red) time-series on a cloudy day: April 3rd, 2013 in Albuquerque, NM. 30-second variability scores are also included in the top left. [Bottom] Same HRIA sample (red), compared to ground measurements smoothed over the area covered by the 18 pyranometers in the NREL Oahu sensor network (yellow).

To help understand the performance at each timescale on this highly variable day, we used a wavelet decomposition. The wavelet decomposition allows for the resolution of variability at a variety of timescales [108]. For example, small clouds may cause variability at short timescales (e.g., 30-seconds), while longer-term weather trends will lead to long-term variability (e.g., 1-hour): these are resolved as fluctuations in short or long wavelet timescales. Figure 44 shows the wavelet decomposition of the clear-sky index for both the ground, HRIA, and smoothed ground samples on April 3rd, 2013 (the day shown in Figure 43). The smoothing applied to create the smoothed ground sample can be seen to reduce the ground variability at shorter timescales (i.e., the 32s wavelet timescale).

The ~1h timescale HRIA variability matches well (at least in magnitude) with both the ground and the smoothed ground variability. Matches vary at other timescales, but in general, the HRIA appears to underestimate the variability on this day at all timescales less than 30-minutes, even when compared to the smoothed ground sample. Specifically, the variability underestimation in the ~1m to ~17m range suggests that the SIND method is largely responsible for the underestimation of variability on this day.

Possible improvements to better match high-frequency variability on a cloudy day include adding more data to the library of lookup samples for both the SIND and the 4-second algorithms and making sure that the library measurements match the spatial diversity of the ground measurements they are meant to represent.

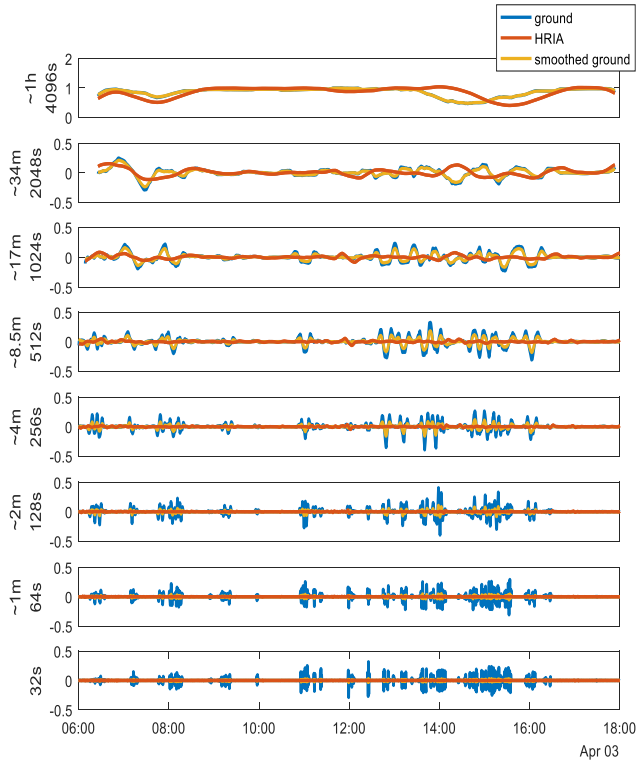


Figure 44: Wavelet decomposition showing variability performance at timescales ranging from 32s to ~1h. Irradiance samples were divided by a clear-sky model to create clear-sky index values before performing the wavelet decomposition.

4.3.1.4. Development of improved techniques

Following the initial validation, it was clear that the existing HRIA was not performing well during periods of high irradiance variation (i.e. partly cloudy days). Many variables within SIND and the further downscaling methods were evaluated for sensitivity for generating more accurate, high-temporal resolution and high-spatial-resolution downscaled data from the NSRDB. For instance, SIND looked at the clearness index of multiple adjacent 2 by 2-km pixels in order to make a determination of the "variability class" or "irradiance variability library" which was relevant to the current atmospheric conditions. As a distribution-system is relatively small the effect of these calculated "patches" was removed from the SIND algorithm, but no marked improvement was detected. Further similar investigations were completed such as the attempted correlation of irradiance variability with other variables available in the NSRDB in order to provide more data for the decision of what variability conditions were predominant over a 30-minute period. A variable indicating the "cloud class" was available and appeared to be a potentially promising way of applying variability libraries for ramp rate generation but further investigation showed that the cloud classes, while helpful, did not obviously predict all variability conditions. In this work, it was also determined that the originally defined variability classes (also known as ramp rate libraries) were simply too smooth. Even though they had been generated from actual 1-second irradiance data from a region of the U.S. expected to have high levels of irradiance variability, as shown in Figure 45, even the most variable ramp rate library (denoted as r3), irradiance data from NREL's SRRL could not possibly be fully modeled as all the ramp rate libraries ramp rate densities were too low in the positive and negative ramp tails of the distribution.

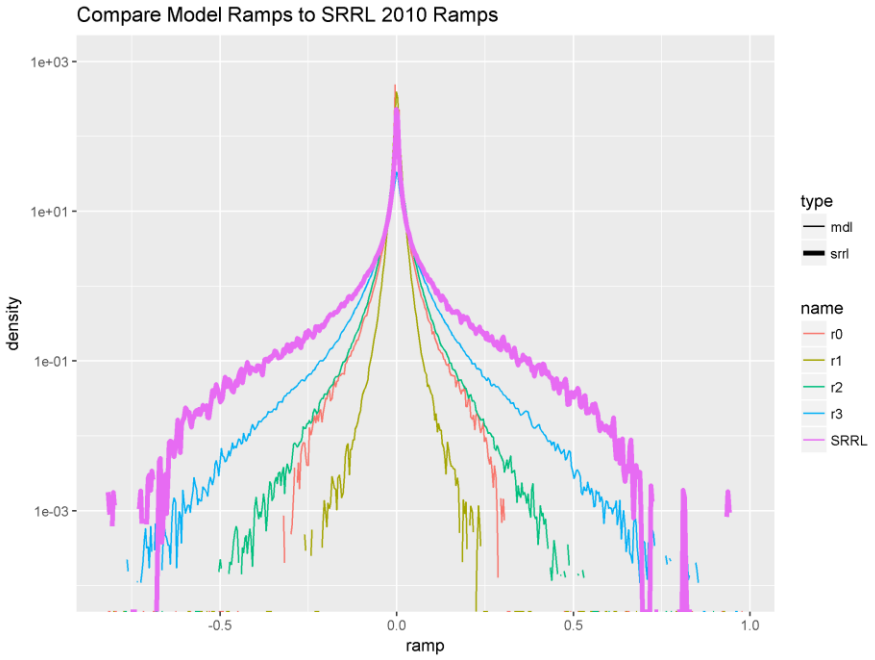


Figure 45. Ramp rate density distribution for measured data (SRRL) and the HRIA ramp rate classes (r0-r3).

As it was clear that the previously developed ramp rate libraries could not be leveraged the evaluation of three alternative methods was undertaken. These methods include 1) a direct ramp rate library generation method using nearby collected irradiance data 2) a statistical approach again using site-specific data to train a "jump process" for variability modeling and 3) a fully statistical approach using a mixture method and only using NSRDB data as an input.

4.3.1.4.1. Site-Specific Ramp Rate Libraries

The concept of this approach was to develop ramp rate libraries that could then be used within the HRIA modeling framework but that correctly binned time periods of irradiance variability appropriately into variability classes. Similar to some previous tests, the cloud type or cloud class was used as an indicator of potentially different irradiance variability regimes. Each 30-minute period of nearby site data (i.e. data from the nearest publicly available high-quality irradiance measurement) was divided up by the cloud class reported by NSRDB for the same 30-minute period. Each ramp rate library was then generated by taking the minute-to-minute ramp rates of the entire dataset as divided by cloud type. Figure 46 shows an example result of this technique. There are 10 cloud types (0-9) and each is plotted with a different color. While there is some separation of the ramp rate density by cloud type there are also many cloud types that largely overlap. Some example modeling using this technique was completed but either the transportability or the lack of a clear correlation between the amount of irradiance variability with clout type resulted in a not much-improved implementation of the HRIA, particularly with the additional difficulty of needing a nearby ground data set to build the variability libraries.

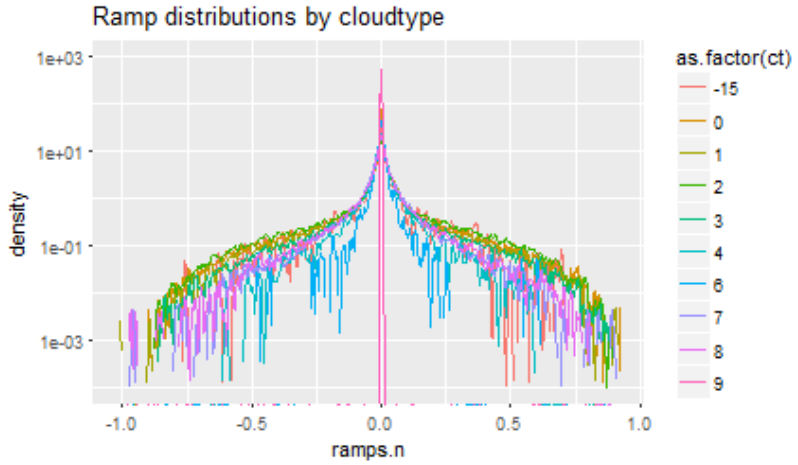


Figure 46. Ramp rate distribution developed by cloud type and using measured irradiance data from site relatively near areas being modeled using the HRIA.

4.3.1.4.2. Non-Gaussian Mixture Model with Jump Process (based on local ground data)

A method that used a non-gaussian mixture model for the solar irradiance, along with a jump process, was evaluated for a method capable of downscaling irradiance data with a reasonable level of expected variability. The full description of this technique is given in [109]. The approach uses added variability the amount of which is determined by the empirical history of ramps or excursions for expected clear sky conditions (driven by nearby irradiance ground data) and a detection method of when extra excursions from clear sky conditions should take place by calculating the variance between reported NSRDB data and clear sky expected conditions as shown in Figure 47. When the two match well, no extra "jumps" are added to the modeled irradiance but when the two depart, and by how much, "jumps" are added, the number and size again depending on the empirically measured amount from a local irradiance measurement.

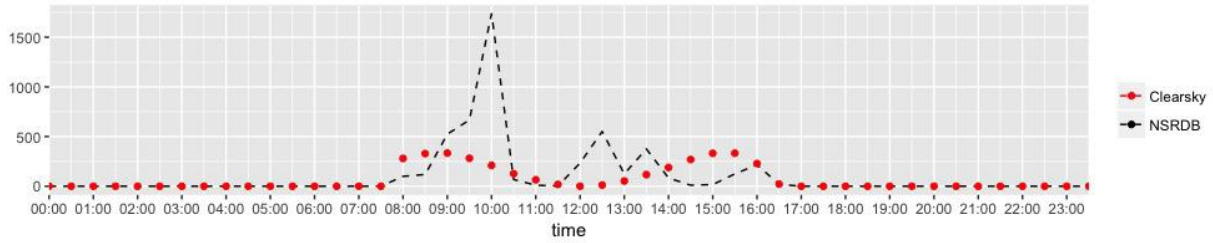


Figure 47. Variance comparison between NSRDB data and clear sky data for a location in Oregon.

This method produced relatively good results when the modeled data locations were close to, and the non-gaussian mixture model was tuned to, local irradiance measurement sites. Figure 48 shows the results of the model (shown in red) when generating synthetic down-sampled irradiance data for Eugene, OR. Each day shows considerable irradiance variability and the non-gaussian model with the jump process also shows considerable variability.

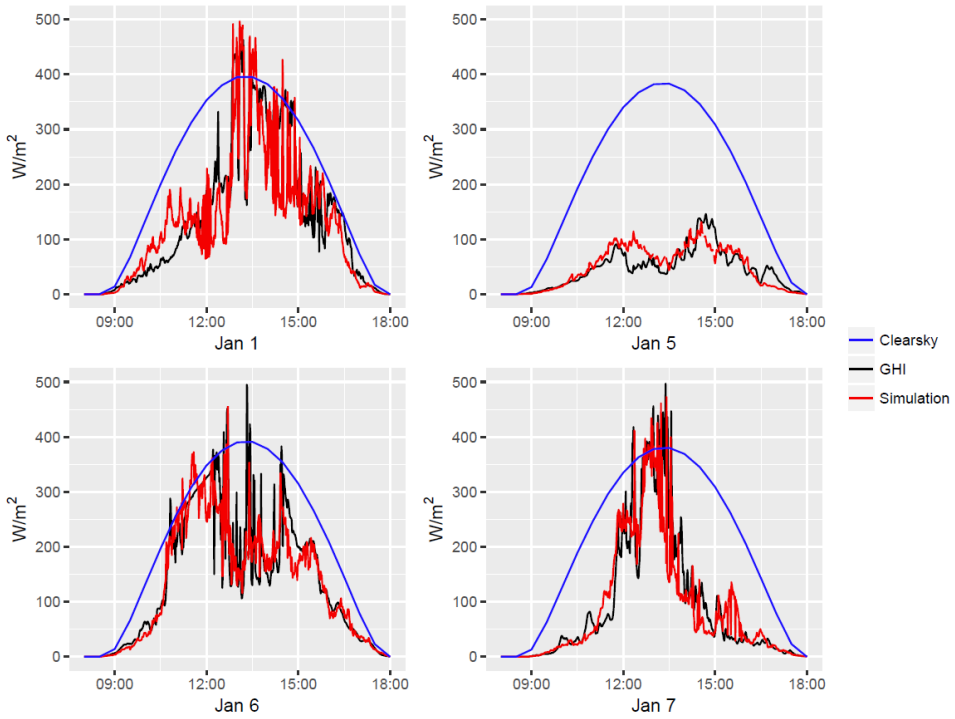


Figure 48. Example simulated/down-sampled irradiance data compared to measured global horizontal irradiance and calculated clear sky irradiance.

However, the transportability of this model from locations not near the measured irradiance sites or for regions of highly variable conditions over relatively small geographic distances proved to be a limiting factor. Thus, a third method was investigated that used only NSRDB data, which is available nationwide, for irradiance modeling purposes.

4.3.1.4.3. Generalized Linear Framework

A method that uses non-gaussian mixture models and stochastic estimators, instead of the empirical jump process, was developed and evaluated to try and model high-temporal irradiance data via downscaling techniques using only NSRDB as a model input. A full description of this irradiance modeling work is available in [110]. This full non-gaussian mixture model approach, along with the

non-gaussian mixture model with the "jump process" method described above was evaluated for modeling the irradiance variability, via down-scaling, for a location on Lanai, HI where high-resolution irradiance data was available to be used to both train the model as well as compare the model results. Figure 49 shows the results of such a comparison. Three time-scales are shown in three different colors (1-minute in blue, 30-minute in red, and hourly in orange). Plotting these different time scales (representing temporal averaging) helps understand what timescale of variability the various models presented over- or under-model. The results for both the jump process and the full mixture model generally appear very similar. The jump process does seem to have better modeled some of the extra irradiance variability expected on the minute-to-minute timescale. However, the mixture model performs well considering that no training of a nearby high-temporal-resolution irradiance measurement was needed. Still, both the jump process and mixture models fall short of modeling the relatively extreme variability visible in the measure Lanai, HI data.

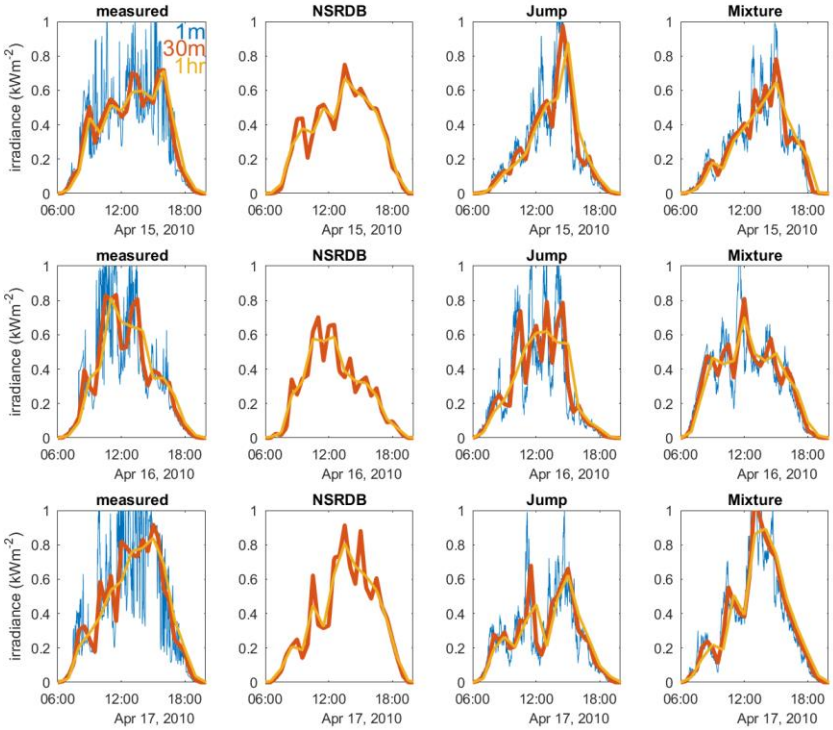


Figure 49. Comparisons of modeled vs. measured irradiance for 3 separate days for Lanai, HI, showing measured data, NSRDB data, "jump process" data and the full non-gaussian mixture model data.

4.3.2. Cloud Fields for Unique Irradiance Profiles

High-frequency solar variability with unique inputs for different interconnection points on distribution feeders is important to accurate QSTS studies. Using a single PV profile for all interconnection points results in an overestimation of the PV impact due to the spatial diversity present in distributed PV (i.e., not all PV systems are covered by the same cloud). In this section, we demonstrate how synthetic cloud fields that match high-frequency irradiance statistics can be integrated into distribution grid QSTS simulations. These synthetic cloud fields allow for unique PV samples at each interconnection point, samples that accurately represent the high-frequency solar variability statistics while also capturing the spatial decorrelation among distributed PV.

4.3.2.1. Generation of cloud fields

The synthetic cloud fields method begins by creating random noise at different spatial scales. Next, each scale of random noise is linearly interpolated to a grid the same size as the finest grid. This results in a smooth field for the larger scales while retaining the more variable field at the smaller scales. These interpolated fields are added together to create a cloud field, as shown in Figure 50. Different weights are applied to the different interpolated fields. These weights are related to the solar variability at each timescale: longer-term variability (e.g., large, slow passing clouds) will lead to higher weighting on the coarsely interpolated scales, while shorter-term variability (e.g., small clouds leading to short fluctuations) will mean higher weighting of the fine scales.

However, this initial cloud field does not look like actual sky conditions: values range from fully clear to fully cloudy without distinct cloud shapes, as seen in the bottom right "cloud field" in Figure 50.

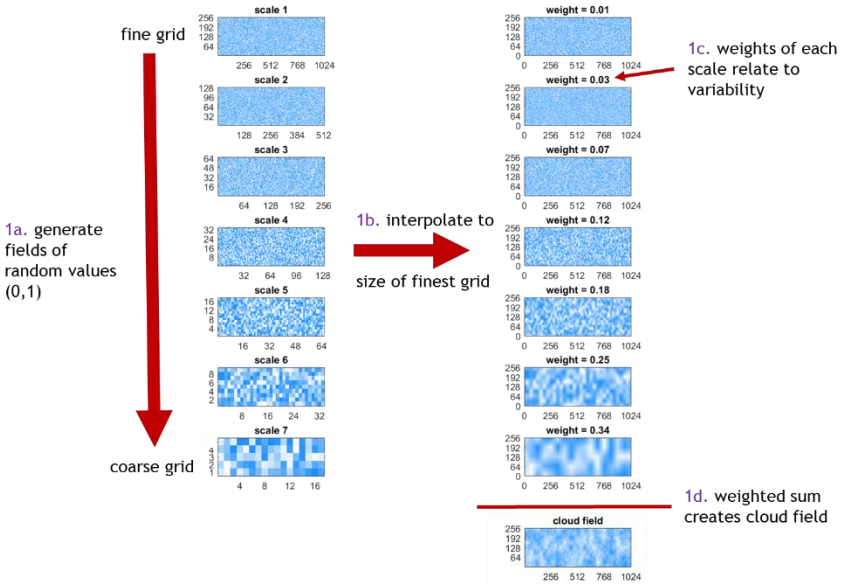


Figure 50: Initial cloud field created by summing all the interpolated fields.

To obtain more distinct clouds, we create a cloud mask, which is based on the expected fraction of the sky covered by clouds, as seen in Figure 51. This mask is applied by setting all "clear" areas (green color in top of Figure 51) to 1 (fully clear). Areas in the "cloudy" region of the mask initially retain their previous values (this was later improved upon – see section 4.3.2.1.3). The resulting cloud field values range from 0 to 1 and are analogous to clear-sky index samples. They can be converted to GHI by multiplying by a GHI clear-sky model (e.g., [111]).

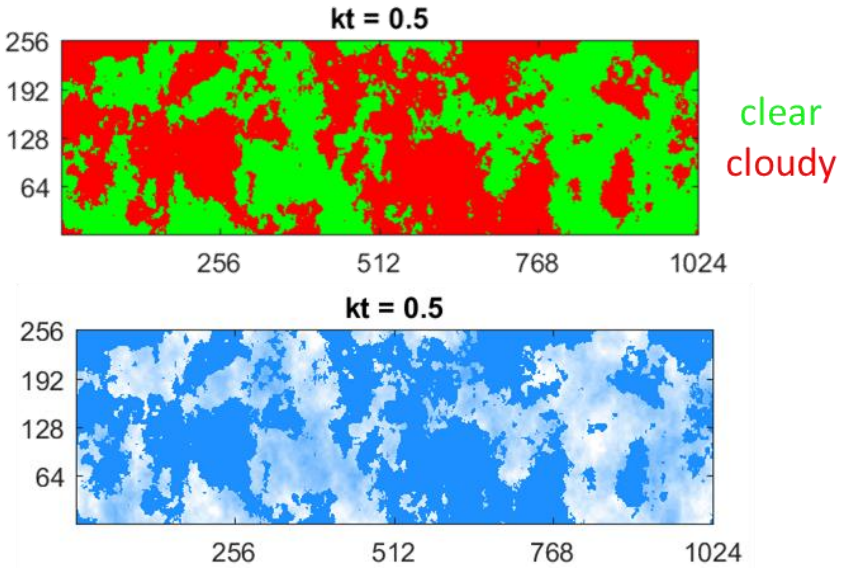


Figure 51: [Top] Cloud mask. [Bottom] Resulting cloud field after the mask is applied.

Overall, the main factors which affect cloud field creation are:

A) The weighting of fine versus coarse scales – this impacts the variability of the cloud field.
B) The percentage of the area covered by clouds – this determines how many areas are clear blue sky.
C) The average intensity of clouds – clouds may tend to be opaque or more translucent.

These main factors are best derived from a single high-frequency irradiance sample. In this way, the single measurement is extended through the cloud field to create unique PV samples at each point over an area, such as the area of a distribution feeder. In this section, we discuss several improvements made to the cloud field implementation through the project, which incorporate high-frequency (e.g., 1-second) irradiance inputs and result in more accurate irradiance statistics across the synthetic cloud fields. Additional details on the previous ("old") implementation are available in [112].

4.3.2.1.1. Scale Weighting

Under the previous implementation, the weighting of fine versus coarse cloud field scales (factor A) was a monotonic function of the variability score. Scaling weights increased significantly at higher-order (coarser) timescales, and thus often resulted in large cloud features dominating leading to sharp cloud edges and higher than expected solar variability. Additionally, under the previous implementation only 8 (coarse to smooth) cloud field scales were considered. Improvements implemented spatial scale weighting based on a wavelet transform of the measurements from a high-frequency point irradiance sensor and increased the cloud field scales to 12 to be consistent with the temporal scales computed based on the wavelet transform [108] of the input high-frequency irradiance. This results in cloud field scale weighting that is not necessarily monotonically increasing and can better capture variation among the timescales, as seen in Figure 52.

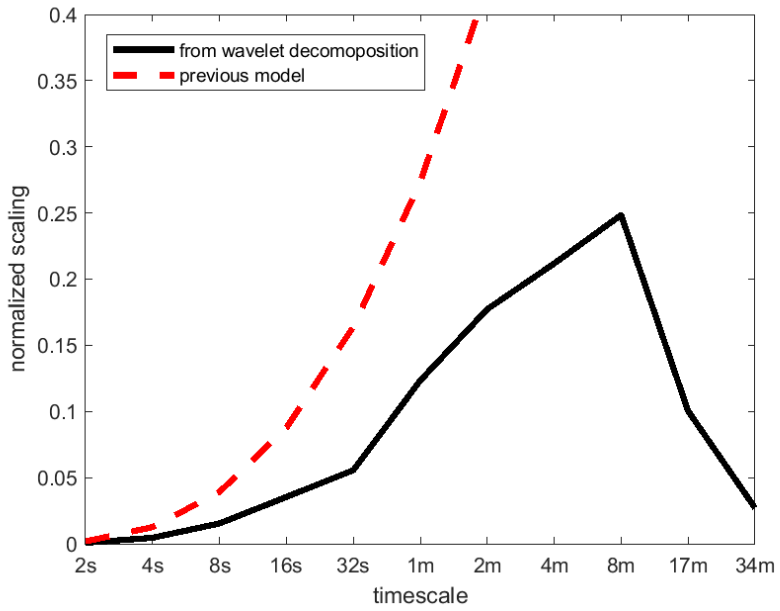


Figure 52: Normalized scaling of cloud fields using a wavelet decomposition (black line), showing the enhanced detail versus the previous simple model (red dashed line).

4.3.2.1.2. Clear-Sky Fraction

The fraction of clear-sky areas (factor B) was previously computed from the average clear-sky index of the irradiance sample. This often resulted in an over-prediction of the areas of clear-sky. For example, a fully cloudy period of varying cloud opacities (say from 0.3 to 0.7) may result in an average clear-sky index of 0.5. Although this period was fully cloudy, the previous method assumed that 50% of the spatial area was clear-sky. Improvements were implemented to instead account for clear periods based on the percentage of values in the measured time-series which had a clear-sky index greater than 0.9. In this way, the percentage of clear-sky pixels was decoupled from the average clear-sky index.

4.3.2.1.3. Cloud Opacity

Two additional enhancements were made to cloud intensities. First, the new implementation allowed for cloud enhancement – values greater than 1. Cloud enhancement was programmed to only occur at cloud edges – the interface between cloudy areas and clear areas (edges of the cloud mask). Second, the intensity of clouds was scaled to match the average clear-sky index.

Figure 53 compares the ramp rates of the clear-sky index for the improved ("new") and previous ("old") methods to the measured clear-sky index from a high-frequency irradiance sensor in Oahu, Hawaii. It also shows that the new method outperforms the old method, especially during periods of large clouds such as hours 10 and 12. Not shown in Figure 53 is that the improved method also accurately matches ramp rate distributions during very (but not fully) clear conditions during hours 16 and 17 (see Figure 54), while the old method had assumed fully clear conditions and hence that all ramps had zero magnitude.

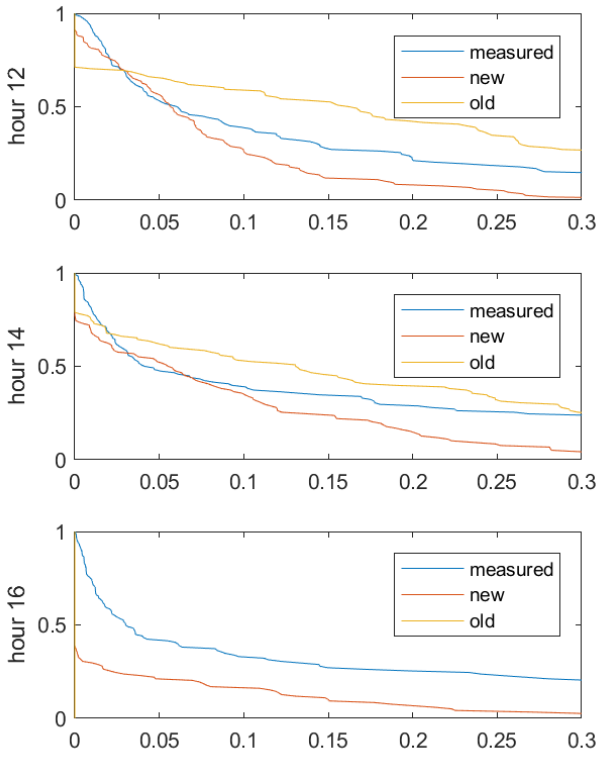


Figure 53: Ramp rate statistics for the measured clear-sky index, and for the method with improvements ("new") and the previous method ("old").

Figure 54 shows the cloud fields created with the previous method versus the updated method. Also included are clear-sky indices derived from the cloud fields compared to the measured clear-sky index. The new method better matches the clear-sky index trends of the measured data. The updated method is much better at capturing the long, continuous cloud features such as those present at hour 6 and hour 10.

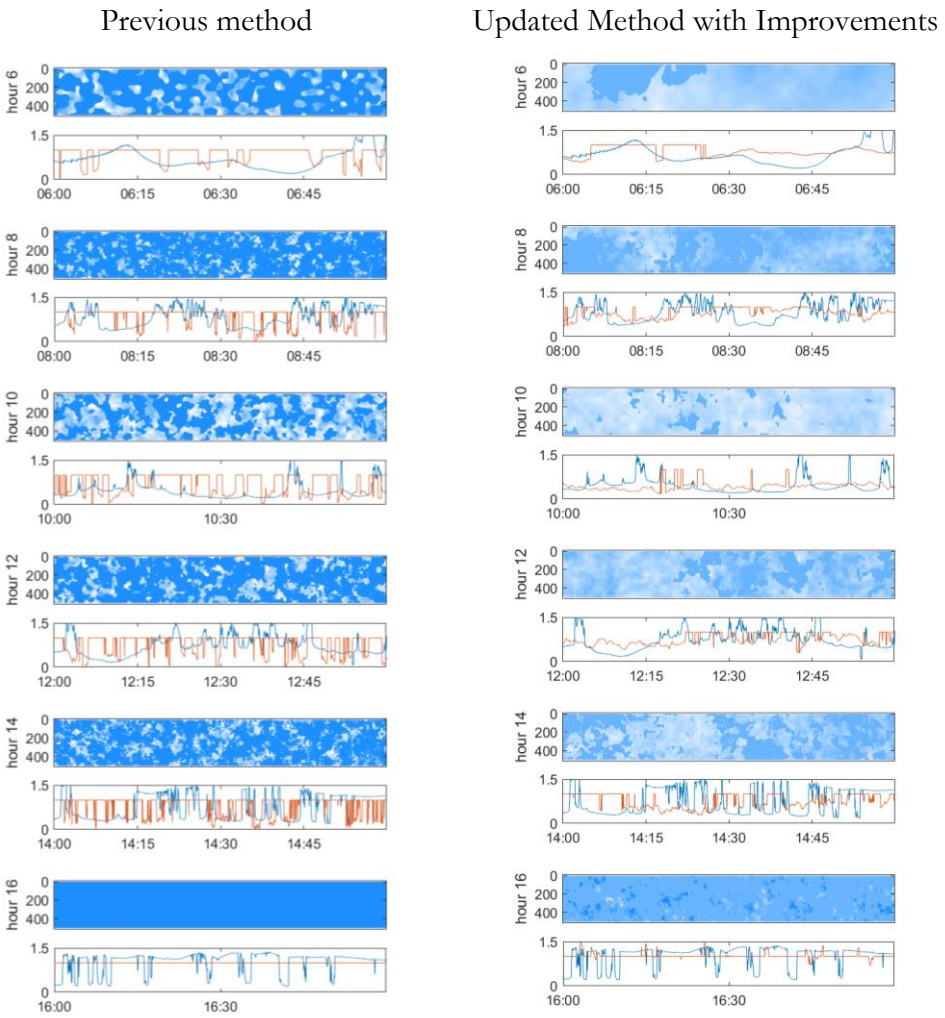


Figure 54: Comparison of cloud fields and clear-sky index samples for the old method to the new method. The blue line is the measured clear-sky index and the red line is the modeled clear-sky index.

4.3.2.2. Implementation and Evaluation

Cloud fields using the method which includes all the improvements described in Sections 4.3.2.1.1, 4.3.2.1.2, and 4.3.2.1.3 were applied to produce unique PV power output profiles for use in QSTS simulations, as outlined in the flow chart in Figure 55. First, the PV locations on the feeder to be simulated were mapped to the cloud field. Next, the cloud field was advected through time-based on the cloud speed – a constant cloud speed was assumed for the entire cloud field, and for simplicity cloud shapes were static (i.e., the is no deformation or creation of clouds). For each PV location, the passing clear-sky index values were recorded as a time-series. These clear-sky index time-series were then translated to create a simulated irradiance time-series by multiplying by a clear-sky irradiance model. Finally, the simulated irradiance time-series were passed through DC (i.e., PV module characteristics) and AC (i.e., inverter characteristics) power models to create simulated AC power output at each PV interconnection location.

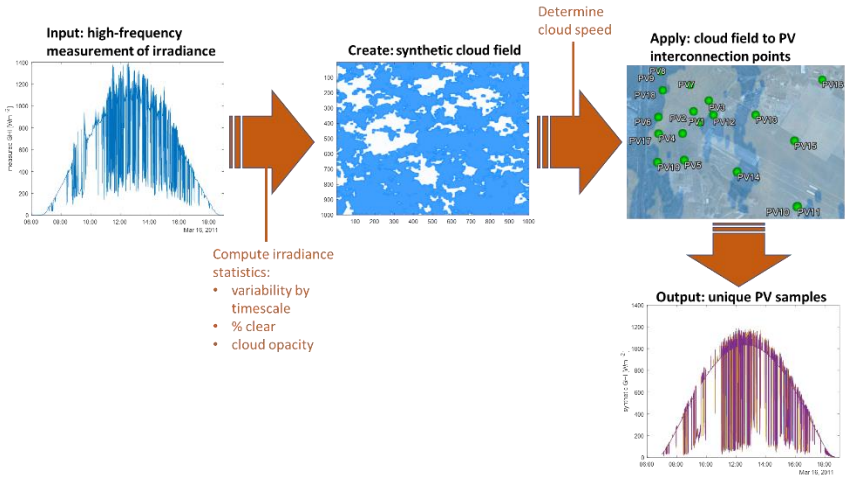


Figure 55: Flow chart showing process to use cloud fields to make unique PV samples across a distribution feeder.

These unique PV power output profiles were applied to distribution grid simulations using the modeling software OpenDSS. We chose an agricultural distribution feeder with 265 different transformers with PV interconnections as our test feeder, as shown in Figure 56. As seen in Figure 56, the feeder has a voltage regulator tap changer roughly halfway down its main line, allowing for direct quantification of the impact of various PV power profiles on voltage fluctuations. So that the voltage regulator is a better indicator of the PV impact, all PV was located downstream of the voltage regulator. A total of 2.8MW of PV was connected over the 265 different transformers, corresponding to roughly two 5kW PV systems installed per transformer. Without PV, the maximum load through the voltage regulator was 3.6MW. Load from the peak load week was used for all simulations.

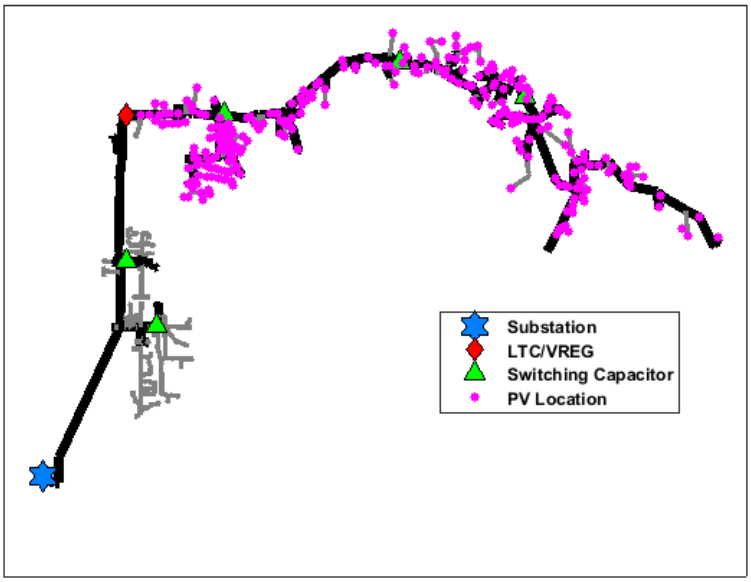


Figure 56: Test feeder layout showing substation (blue star), voltage regulator (red diamond), and the 265 PV interconnection locations (magenta dots).

Two test cases were considered: (a) a comparison of simulations with cloud fields to simulations using a network of irradiance measurements, and (b) a comparison of cloud field simulations to simulations based on only a single irradiance measurement. In each case, cloud fields were paired with QSTS simulations to determine tap change operations, as shown in the flow diagram in Figure 57.

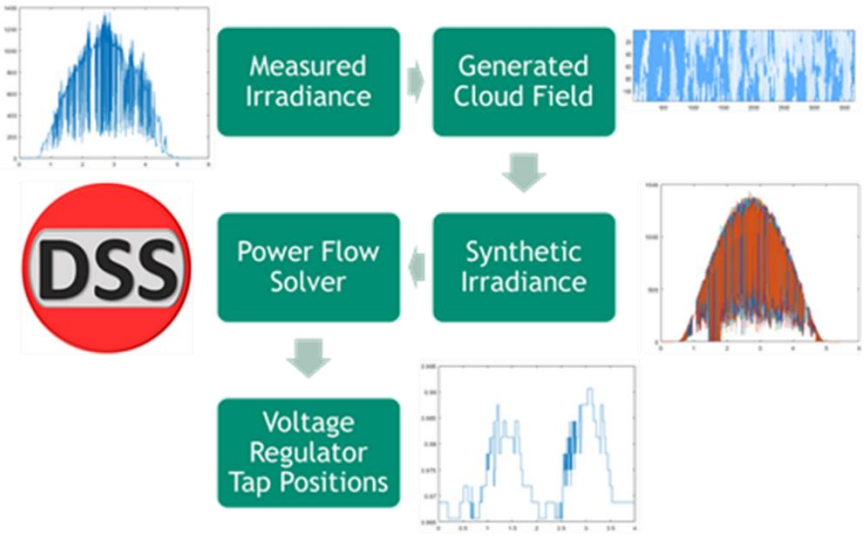


Figure 57: Flow chart showing process to determine tap change operations using QSTS simulations incorporating cloud-field generated irradiance.

4.3.2.2.1. Comparison to Measured Sensor Network

To compare the results of the cloud fields to actual measured irradiance, the measurements from the irradiance network in Oahu, HI were used [4]. There are 19 irradiance sensors in the Oahu network. However, two of the sensors were found to often have bad data and so were eliminated from the analysis. The remaining 17 sensors were used and assigned to one of the PV interconnection points on the feeder. This resulted in many duplicate PV profiles – as there were only 17 unique profiles but 265 interconnection points – but is the best approximation that can be done with only 17 sensors. To mimic this setup, only 17 synthetic irradiance samples derived from cloud fields were used.

Results of this simulation for one week are shown in Figure 58. The synthetic irradiance simulation does a good job of capturing the actual variability of the measured irradiance. The total number of tap change operations over the week is consistent to within 10% between the measured and the synthetic irradiance samples, showing good agreement.

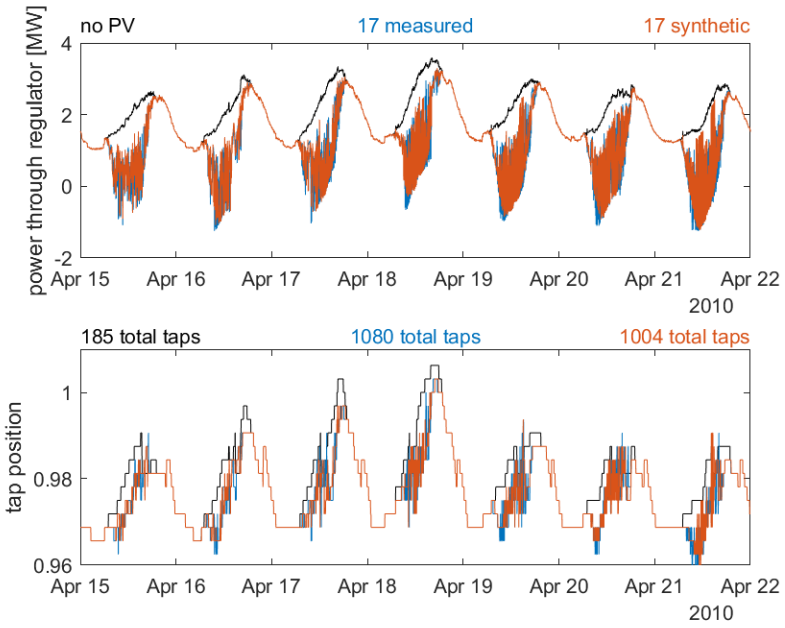


Figure 58: Simulation results when using 17 measured (blue) and 17 synthetic (red) PV inputs. Also included for reference is the no PV case (black).

4.3.2.2.2. Cloud Fields vs. Single Sensor

Cloud fields are created on the presumption that a single irradiance sensor, even at a high temporal frequency, does not capture the spatial diversity across a distribution feeder. In other words, a single irradiance sensor applied as the irradiance profile at all transformers will overestimate the variability on the feeder. The realization of this assumption is shown in Figure 59, where one irradiance measurement applied to all transformers is compared to synthetic profiles which are unique at each transformer.

The difference between the single sensor and the 265 unique profile is clear: the single sensor applied to all transformers results in nearly double the number of tap change operations as the 265 unique profiles. The 265 unique profiles, although synthetic, are a more realistic representation of the actual PV impact since they account for the spatial smoothing.

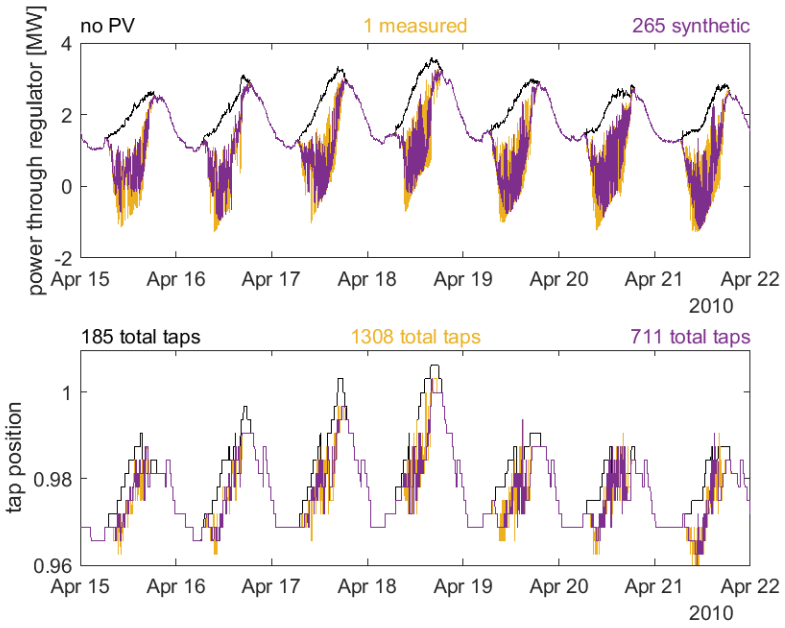


Figure 59: Simulation results when using 1 measured (yellow) and 265 synthetic (magenta) PV inputs. Also included for reference is the no PV case (black).

It is also notable how many fewer tap change operations resulted in the 265 unique profile case (Figure 59) compared to the 17 unique profile case (Figure 58). Although there was smoothing (compared to the single sensor) for the 17 profile case, there was significant additional smoothing for the 265 profiles. This emphasizes how important it is to accurately model the number of unique PV locations – even a relatively dense sensor network (as in the 17 sensor Oahu network) can still significantly underestimate the spatial smoothing.

4.3.2.2.3. Visualization of Spatial Impacts

While the results in sections 4.3.2.2.2 and 4.3.2.2.3 quantitatively demonstrate the value of synthetic cloud fields, it is difficult to convey the spatial variation visually across the 265 different PV locations. In this section, we visually show the impacts on a smaller feeder and across only seven PV interconnection locations.

The feeder used is a typical rural feeder with many branches. Along some of these branches, PV is connected. This PV is varied based on cloud field simulations. This PV variation is fed into QSTS simulations to see the impact on voltage along the feeder. This is shown in Figure 60, where cloud fields, instantaneous PV power, and instantaneous voltage are shown for two different instants in time 1-minute apart. By comparing the two instants, we see that changes in the clouds result in changes in the PV power output, which, in turn, result in changes to the voltage profile along the feeder. This visually demonstrates the importance of spatially varied PV profiles – this spatial voltage variation would not be representatively simulated without the cloud field representation of solar irradiance.

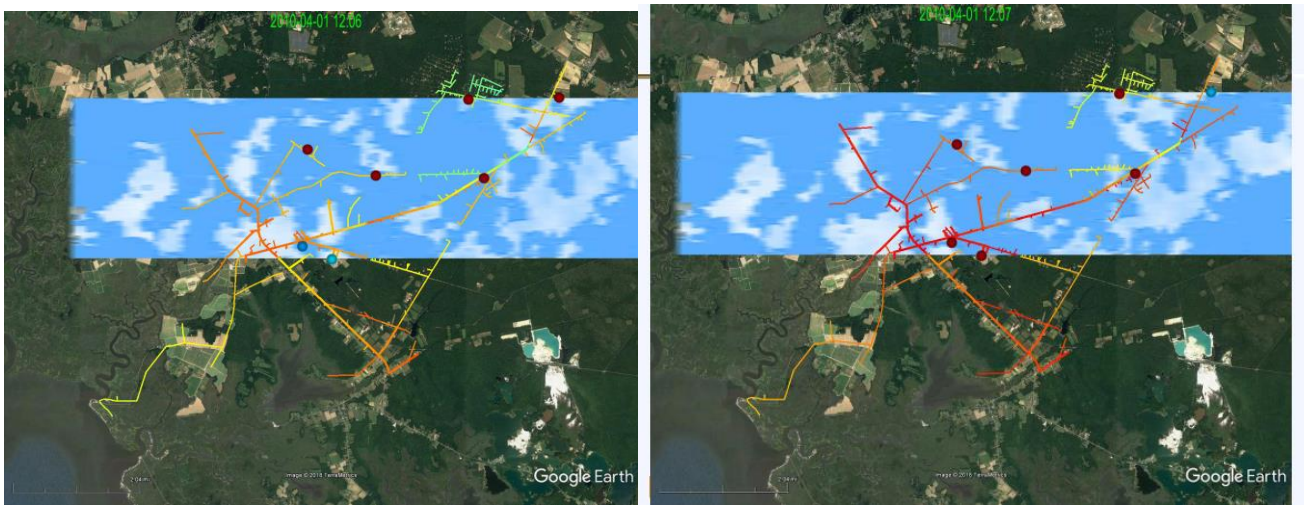


Figure 60: Synthetic cloud fields, simulated PV output, and feeder voltage. The synthetic clouds fields are colored blue (clear) to white (opaque). The PV locations are dots colored blue (low power output) to red (high power output), based on the cloud fields: PV systems under cloudy areas output less power. The feeder lines are colored by voltage from dark red (voltage 1.05 per unit) to dark blue (voltage 0.95 per unit). The left image shows 12:06PM while the right images shows 12:07PM, 1-minute later. Changes in voltage between the two images are due to changes in the cloud fields/PV profiles.

4.4. Summary

The methods presented in this section describe ways to create representative high-frequency inputs for all load and PV nodes on a feeder. The value of these methods over the simple assumption of perfectly correlated load or PV profiles across the feeder can be clearly seen in Figures 37 and 59: by using these methods variability is more granularly and more accurately represented. However, due to limited high-frequency validation data, these methods are still being developed and models or coefficients may be altered slightly as additional measured data is collected. For load modeling, additional high-frequency measurements of customer load profiles will expand the variability library and increase the accuracy of the load chosen to represent a feeder. For PV modeling, (a) additional high-frequency irradiance measurement grids co-located with satellite-derived irradiance will increase the accuracy of the temporal downscaling, and (b) additional high-frequency irradiance or PV measurements across a feeder will allow for further validation of variability profiles produced by synthetic cloud fields. With or without additional measurements, the methods presented here enable load and PV inputs with representative variability, allowing for significantly more accurate QSTS simulations than those using simple load and PV assumptions.

5. IMPLEMENTATION OF RAPID QSTS

The Rapid QSTS algorithms were implemented into three main software environments. First is an open-source release of the algorithms in MATLAB. This allows other researchers to easily test and modify the code for any applications. The second is the implementation of the algorithms into OpenDSS. By integrating the algorithms in OpenDSS, the academic and research community that uses OpenDSS can immediately see the benefits of faster QSTS simulations without having to run any external code. The third is implementation into CYME, which allows utilities to have access to more powerful algorithms in the software environment that they are used to.

Open Source Matlab Release

- Variable Time-Step
- Vector Quantization
- Event-based Simulation
- Detailed Equivalent Circuit Reduction
- Irradiance Modelling

OpenDSS Integrated

- Faster Power Flow Solver
- Multi-Rate Time-Step
- Reduction of Switches and Lateral
- Temporal Parallelization
- Diakoptics

CYME Integrated

- QSTS Long-Term High-Resolution Study Capable
- Faster Power Flow Solver
- Circuit Reduction, including GUI
- Detailed Variable Time-Step
- Dynamic Data Pull

	MATLAB	OpenDSS	CYME
Variable Time-Step	✓	✓	✓
Vector Quantization	✓		
Event-based Simulation	✓		
Circuit Reduction	✓	Simple	✓
Temporal Parallelization	✓	✓	✓
Diakoptics		✓	

5.1. MATLAB (Open-source Code)

An open-source MATLAB toolbox was created for interacting with OpenDSS and modeling PV on the distribution system. This toolbox is available online, and the full documentation is provided in [113]. The majority of the functions are useful for interfacing OpenDSS and MATLAB, and they are of generic use for commanding OpenDSS from MATLAB and retrieving information from simulations. A set of functions is also included for modeling PV plant output and setting up the PV plant in the OpenDSS simulation. The toolbox contains functions for modeling the OpenDSS distribution feeder on satellite images with GPS coordinates. Finally, example simulation functions are included to show potential uses of the toolbox functions. Each function in the toolbox is documented with the function use syntax, full description, function input list, function output list, example use, and example output.

OpenDSS (from the Electric Power Research Institute (EPRI) [58]) is used to model the distribution system with MATLAB providing the frontend user interface through a COM interface. OpenDSS is designed for distribution system analysis and is very good at time-series analysis with changing variables and dynamic control. OpenDSS is command based and has limited visualization capabilities. By bringing control of OpenDSS to MATLAB, the functionality of OpenDSS is utilized while adding the looping, advanced analysis, and visualization abilities of MATLAB.

GridPV Toolbox is a well-documented tool for MATLAB that can be used to build distribution grid performance models using OpenDSS. Simulations with this tool can be used to evaluate the impact of solar energy on the distribution system [114, 115].

The toolbox can be categorized into five main sections:

1. General QSTS Structures and Subfunctions
2. Rapid QSTS Methods
3. Circuit Reduction
4. Irradiance Modeling
5. Load Modeling
6. Visualization

5.1.1. General QSTS Structures and Subfunctions

QSTS simulations can be utilized for a variety of distribution circuit analyses. Therefore, the output data required from a particular QSTS simulation is likely to be dependent on the application at hand. For example, a QSTS study focusing on the impacts of adding a distributed PV system to an existing circuit would likely need the voltage and loading information of the circuit elements near the point of interconnection but wouldn't necessarily need time-series data from every transformer, line, and bus in the system. Furthermore, if a QSTS study were to be run multiple times to account for multiple potential scenarios, the results of each study should be output in a consistent format to simplify the process of analyzing the results. To account for the inherent flexibility of QSTS simulations and to improve the usability of the toolbox, standardized input and output structures have been developed and implemented in each of the QSTS simulation functions. Before a QSTS simulation can begin, a circuit containing time-series profiles must be compiled in OpenDSS and the proper settings must be applied. The MATLAB toolbox also contains subfunctions to address each of these concerns.

5.1.1.1. Input and Output Structures

Each QSTS simulation function can accept a "DataLogging" structure as an optional input. This structure dictates what types of data to record as the simulation progresses through the desired time horizon. Table 13 provides an overview of the options and default values for the "DataLogging" input structure. It should be noted that any fields in this structure not entered by the user are initialized to default values automatically. Thus, the user can set as few or as many fields as desired.

Table 13. Description and default values for the "DataLogging" input structure

Field Name	Default Value	Description
DataLogging.Voltage	True	Logs voltage metrics such as feeder/node max, min, avg values, and time spent outside voltage limits
DataLogging.Controllers	True	Logs total number of tap changes per regulator and total number of switching events per switching capacitor
DataLogging.Thermal	False	Logs max loading on each monitored PDElement
DataLogging.Power	True	Logs power metrics such as feeder max, min, avg values, DER powers, and losses
DataLogging.VoltageTimeseries	False	Logs timeseries voltage metrics for each monitored node
DataLogging.ControllerTimeseries	False	Logs timeseries of tap positions and capacitor states for all regulators and switching capacitors
DataLogging.ThermalTimeseries	False	Logs timeseries thermal metrics for each monitored PDElement
DataLogging.PowerTimeSeries	False	Logs timeseries power metrics for the feeder measurements and DER powers
DataLogging.MonitoredNodes	Empty	Column cell array of strings containing the names of each node to be monitored. If empty, voltage timeseries will return min, max, and avg values for each node
DataLogging.MonitoredPDElements	Empty	Column cell array of strings containing the names of each PDElement to be monitored. If empty, defaults to all transformers and lines
DataLogging.MonitoredDER	Empty	Column cell array of strings containing the names of each DER to be monitored. If empty, defaults to all PV, generators, and energy storage devices
DataLogging.OverVoltageThreshold	1.05 pu	Must be a voltage (pu) threshold between 0.5 and 1.5
DataLogging.UnderVoltageThreshold	0.95 pu	Must be a voltage (pu) threshold between 0.5 and 1.5
DataLogging.OverloadedThreshold	100%	Must be a percentage threshold between 25 and 300%

A standardized output data structure, labeled "QSTS," has been developed and implemented in each of the QSTS simulation functions as well. This nested output structure, depicted in Figure 61, contains all pertinent information relevant to the completed QSTS simulation. Specifically, it contains information about the circuit in OpenDSS, the system on which the simulation was executed (e.g. MATLAB version, OpenDSS version, computer type, etc.), simulation settings, and simulation results. The results section contains sub-structures for different types of results (such as voltage, power, loading, and controller data), a copy of the "DataLogging" input structure, and an array of time points that were solved during the simulation. The voltage, power, loading, and controller subsections each contain both time-series and summarized data (e.g. minimum/maximum values or total state changes for controllable devices). To ensure continuity, each field in Figure 61 will appear in the "QSTS" output struct regardless of whether it contains data or not. Overall, the structure is organized such that the results from any given simulation can be easily identified and analyzed.

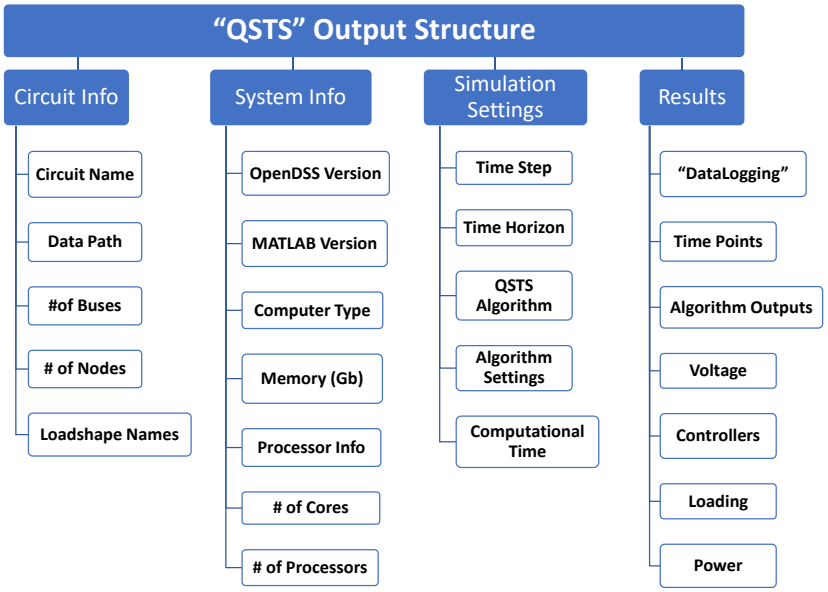


Figure 61. Hierarchy diagram of the standardized "QSTS" output function

5.1.1.2. QSTS Subfunctions

In QSTS simulations, time-varying circuit components like load and PV generation each have profiles associated with them, consisting of a series of multiplier values. These time-series profiles, or loadshapes, are typically created from actual measured data and dictate how the components change through time. Before a QSTS simulation begins, it is important to ensure that each loadshape has the same time-step resolution and time horizon. If not, the loadshapes will be interpolated to meet the requirements of the simulation by the "**getLoadshapes( )**" subfunction. For example, if the user wants to run a QSTS simulation at a 1-second time-resolution, each loadshape will be interpolated from their current time-step resolution down to 1-second. Similarly, if the length of a given loadshape is less than the time horizon of the simulation, that loadshape will be repeated until its length is the same. By default, the loadshapes will each be interpolated down to the minimum time-step resolution of all the loadshapes and extended to the same length as the longest time horizon of all the loadshapes. While this process is similar to how OpenDSS would handle the loadshapes, implementing it as a pre-processing step saves computational time during the QSTS simulation.

The "getLoadshapes( )" subfunction also returns warnings and error messages when necessary. First, the subfunction checks that every load and PC Element has a loadshape assigned to it and displays an error message for any elements that do not. Likewise, if a loadshape contains any NaN values, an error message will appear as OpenDSS will not accept NaN values. Upon completion, all loadshapes will be consistent according to the desired (or default) time-step resolution and time horizon.

Once the loadshapes have been addressed, the QSTS initialization process can begin. QSTS simulations require specific settings in OpenDSS for proper execution; this process is handled by the "**initQSTS( )**" subfunction. First, the simulation is initialized to the beginning of time by setting "hour=0" and "sec=0." Next, the solution mode is set to "duty" mode, which tells OpenDSS to follow the time-series multipliers stored in the loadshapes as time progresses.

Some circuits may be prone to oscillations at the beginning of a QSTS simulation as controllable devices interact with one another (e.g. an LTC changing its tap position causing a capacitor bank to

open). To alleviate these potential artifacts, the "initQSTS( )" subfunction will first perform a power flow solution using the "static" control mode, where time does not advance and all control actions are taken until there are none left in the control queue. Once the devices have settled to their appropriate states, the control mode is set to "time," where control actions are only taken when the time for the pending action is reached or surpassed, i.e. when the device's time delay has expired. At this point, the "reset" command is issued to clear any data associated with the initialization process and the circuit is ready for a QSTS simulation.

In addition to applying the appropriate settings, the "initQSTS( )" subfunction also contains several other safeguards. First, this subfunction analyzes the results from "getLoadshapes( )" to ensure that the time-step and time horizon are within acceptable ranges for QSTS simulations, according to [53]. Lastly, "initQSTS( )" parses the optional "DataLogging" input structure to assign default values where necessary and initializes the "QSTS" output structure.

5.1.2. Rapid QSTS Methods

The following subsections provide details about each of the rapid QSTS algorithms that have been implemented in the MATLAB toolbox. Specifically, the available algorithms include variable time-step, vector quantization, event-based, and temporal parallelization.

5.1.2.1. Variable Time-Step

The function "QSTS_VTS( )" runs a rapid QSTS simulation using the variable time-step algorithm [77]. First, a sensitivity analysis is conducted to see which loadshapes have the most impact on the controllable devices (normalized to each controller's deadband). The variable *dbPercent* sets the deviation threshold for pre-determining time points in terms of "percent of deadband." In other words, this variable tells the algorithm how sensitive to be when pre-determining time points. Acceptable values are between 0-100%, while typical values are in the range of 5-50%. For example, if *dbPercent*=10%, the algorithm will analyze the loadshapes for any time periods that would cause a 10% change in the control voltage of a controllable device. Therefore, this variable is inversely correlated with solution time and accuracy. The variable *maxJump* sets the largest time-step (in seconds) the algorithm will take when moving through time. Typical values range from 60s to 300s. This variable is also inversely correlated with solution time and accuracy. The results from the sensitivity analysis and the *maxJump* variable are used to pre-process the loadshapes and output time points of interest. The algorithm then begins the QSTS simulation by solving these time points sequentially until the control queue is populated, which indicates the presence of a pending control action. The algorithm then backtracks to the last solution and proceeds forward at a finer time-step resolution. The advantage of the variable time-step algorithm is that it can move quickly through periods of time that have low variability and move slowly through time periods of high variability. Overall, this algorithm is roughly 50x faster than the brute-force QSTS simulations while keeping errors below the acceptable limits [77].

To run a rapid QSTS simulation with the "QSTS_VTS( )" function, the user must pass in a compiled OpenDSS circuit object, *DSSCircObj*, and settings for the *dbPercent* and *maxJump* variables. The function also accepts the optional *DataLogging* structure and *timeEnd* variable, which is used to set the desired time horizon of the simulation. Upon completion, the function will output the *QSTS* structure containing the results from the simulation.

5.1.2.2. Vector Quantization

The function "QSTS_VQ( )" runs a rapid QSTS simulation that bypasses most of the time-consuming non-linear 3-phase unbalanced AC power flows by directly assigning the power flow solution based on similar power flows that occurred previously in the simulation, based on the algorithm described in [78]. Because each time step is solved chronologically, controllable elements with hysteresis can still be accurately modeled. First, the function performs a sensitivity analysis like the "QSTS_VTS( )" function. Then, the function discretizes the loadshapes based on the results of the sensitivity analysis. As the precision of the profiles is reduced through vector quantization, the number of unique power flow solutions computed will decrease. The proposed quantization algorithm provides real-time data compression for memory management by only storing unique solutions in the solution space matrix and the solution index as a time-series vector. The time-series matrix can easily be reconstructed ex-situ without having to run the QSTS simulation again to compute any metrics of interest.

To run a rapid QSTS simulation with the "QSTS_VTS( )" function, the user must pass in a compiled OpenDSS circuit object, *DSSCircObj*, and setting for the *dbPercent* variable. The function also accepts the optional *DataLogging* structure, *timeStep,* and *timeEnd* variables, which are used to set the desired time step and time horizon of the simulation. The user can also choose from three different quantization methods, using the optional *VQmethod* parameter. Upon completion, the function will output the *QSTS* structure containing the results from the simulation.

5.1.2.3. Event-Based

The function "QSTS_EB( )" runs a rapid QSTS simulation by exploiting the state-based linear sensitivity model described in [84]. The algorithm relies on creating a local linearization of the power flow manifold to quickly estimate the nodal voltages and branch currents in the circuit. The local linearization is done using the ordinary least squares estimator. In order to get an accurate estimate of the manifold, the algorithm first performs a sensitivity analysis to estimate the size of the neighborhood around the linearization point, using the same type of sensitivity analysis as the "QSTS_VTS( )" function. Once the sensitivity analysis is complete, the algorithm starts estimating models for each node in the circuit. Using these models, the algorithm can estimate the control voltages of different controllers and their corresponding states at each time instant. In case of an event (such as a tap change or capacitor bank connection/disconnection), the algorithm updates the models for all the nodes either by creating a new local linearization or if the state has been observed before, extracting model parameters from a lookup table. Once the models are estimated for each observed state for the given time horizon, the algorithm then reconstructs the voltage and current profiles. This resolution of reconstruction is defined by the user via the *reconstructionTimeStep* variable, which can range from 1 second up to 15 minutes. The advantage of the Event-Based algorithm is the significant speed reduction it can achieve while maintaining high accuracy to estimate the QSTS impact metrics. Overall, this algorithm is roughly 110x faster than the brute-force QSTS simulations [84].

To run a rapid QSTS simulation with the "QSTS_EB( )" function, the user must pass in a compiled OpenDSS circuit object, *DSSCircObj*, and settings for the *dbPercent* and *reconstructionTimeStep* variables. The function also accepts the optional *DataLogging* structure, *timeStep,* and *timeEnd* variables, which are used to set the desired time step and time horizon of the simulation. Upon completion, the function will output the *QSTS* structure containing the results from the simulation.

5.1.2.4. Temporal Parallelization

The function "QSTS_Parallel( )" runs a rapid QSTS simulation by dividing the total time horizon of the simulation into multiple sections and solving those sections simultaneously on multiple processors of a single computer. First, the compiled circuit object is copied onto multiple "actors" using the "clone" function in OpenDSS-PM (EPRI's more modular, flexible, and scalable version of OpenDSS for parallel processing), based on the required input *numActors*. This variable determines how many logical processors will be utilized for the simulation. Each actor created by OpenDSS-PM runs on a separate processor (if possible) using separate threads and has its own assigned core and priority. The total number of actors available for a parallelized QSTS simulation is limited by the number of logical processors available on the computer. However, to prevent the computer from becoming unresponsive, it is recommended that at least one processor is left unused by the simulation. Once the actors have been established, the "QSTS_Parallel( )" function initializes each of them for a QSTS simulation by setting their start and end times accordingly and adds monitors to various circuit elements based on the optional *DataLogging* input structure. Upon completion, the function gathers the data from each monitor and adds it to their corresponding locations in the *QSTS* output structure. The speed improvement from this algorithm is based on the size of the circuit and the number of processors being used. Overall, temporally parallelized QSTS simulations were shown to have a speed improvement of up to 8.156x faster when using 16 cores [94].

To run a rapid QSTS simulation with the "QSTS_Parallel( )" function, the user must pass in a compiled OpenDSS circuit object, *DSSCircObj*, and setting for the *numActors* variable. The function also accepts the optional *DataLogging* structure, *timeStep,* and *timeEnd* variables, which are used to set the desired time step and time horizon of the simulation. Upon completion, the function will output the *QSTS* structure containing the results from the simulation.

5.1.3. Circuit Reduction

The simulation time of power flow solvers is dependent on the size of the circuit's incidence matrix and each bus entry adds one row and one column to the matrix. Any reduction in the total number of buses will decrease the size of the incidence matrix and reduce simulation time. The bus reduction algorithm presented in this work is able to remove non-vital buses without loss of accuracy on the remaining buses. However, with this reduction meaningful data is only saved for buses that are designated as buses of interest (BOI).

Buses of interest are divided into two categories: automatic BOI and user-selected BOI. Automatic BOI are necessary for the circuit reduction algorithm, as these cannot be reduced while maintaining accuracy. Automatic BOI include capacitors buses, buses that are part of the transmission system, buses with meters and monitors, transformer buses with LTC and voltage regulators, and the source bus. On the other hand, the user-selected BOI are buses the user has designated as significant. Any bus can be a user-selected BOI. Lastly, once the list of automatic and user-selected BOI is constructed, the algorithm adds topological buses and important in-line buses to ensure the circuit is connected and reduction is possible.

Open-source GUI's are provided for selecting the BOI either in generic form (Figure 62) or custom bus selection (Figure 63). After selecting the BOI, the circuit reduction process is run through MATLAB functions to remove 1) empty buses, 2) laterals without BOI, and non-BOI load buses [7, 95].

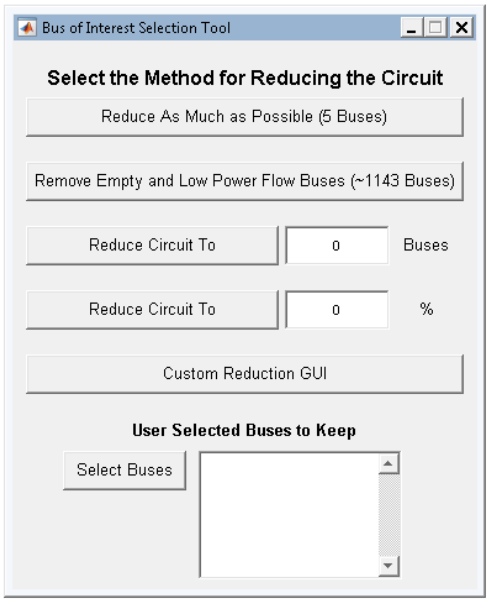


Figure 62. GUI offering simple and custom bus of interest reduction options

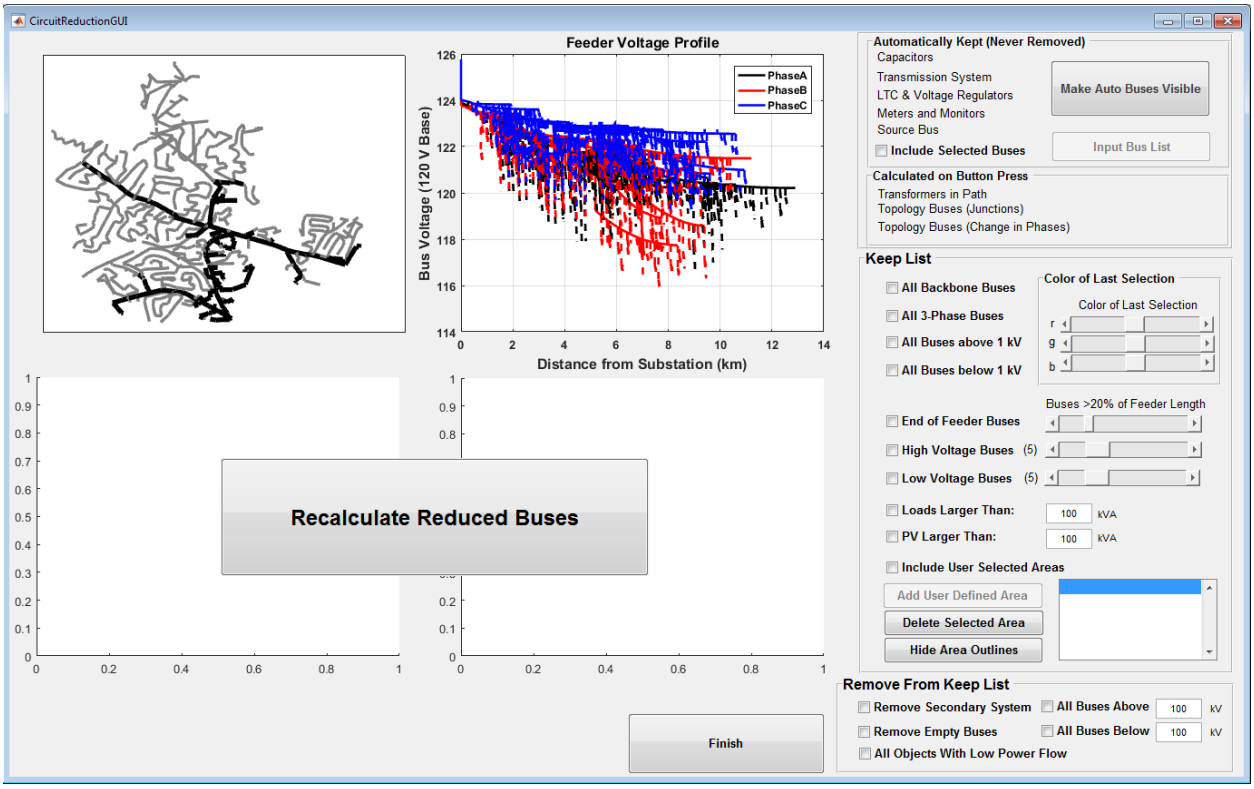


Figure 63. Custom GUI for interactive BOI selection

5.1.4. Cloud Field Modeling

Cloud field modeling was integrated into MATLAB in a format consistent with the GridPV toolbox. For best integration with OpenDSS simulations, the cloud field code reads the PV locations from a

GridPV circuit object. It additionally needs weather inputs: one high-frequency irradiance time-series (can be either measured or synthetic based on the temporal downscaling methods described in the section above) and the measured or estimated cloud speed. The output from the cloud fields code includes the PV power load shapes, the number of points in the load shapes, and the time interval between the data points – all the values needed to create a new PV loadshape in OpenDSS. Thus, OpenDSS simulations can be easily updated to include the cloud-field-generated PV power profiles by editing the PV systems to use these new irradiance loadshapes. This process is illustrated in Figure 64.

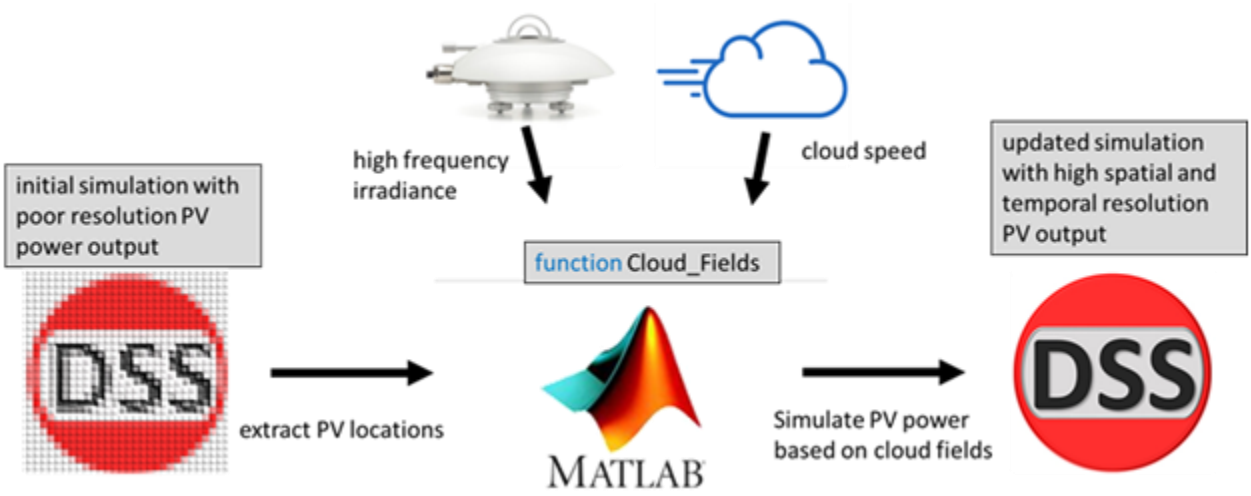


Figure 64. Illustration of "Cloud_Fields" MATLAB function which implements PV profiles based on cloud fields into OpenDSS simulations.

5.1.5. Load Modeling

The methods developed for variability and diversity modeling to increase the effective temporal resolution of commonly available distribution feeder data were implemented in MATLAB. This tool can develop load profiles for multiple loads on a single circuit, with a target diversity factor, and can add load variability up to a 1-minute temporal resolution, all while maintaining the overall feeder load at the aggregate level (i.e. at the start-of-circuit level often available via utility SCADA). This MATLAB tool is provided in a complied executable form as it includes high-resolution load data used within the project to develop the load variability models. This data is included in the complied form to protect its source while still allowing the inclusion of this base (or example) load variability library. The tool is also capable of developing custom load variability models as appropriate for use in areas with significantly different climates and load use patterns as the provided example variability library. A screenshot of the load modeling tool is shown in Figure 65.

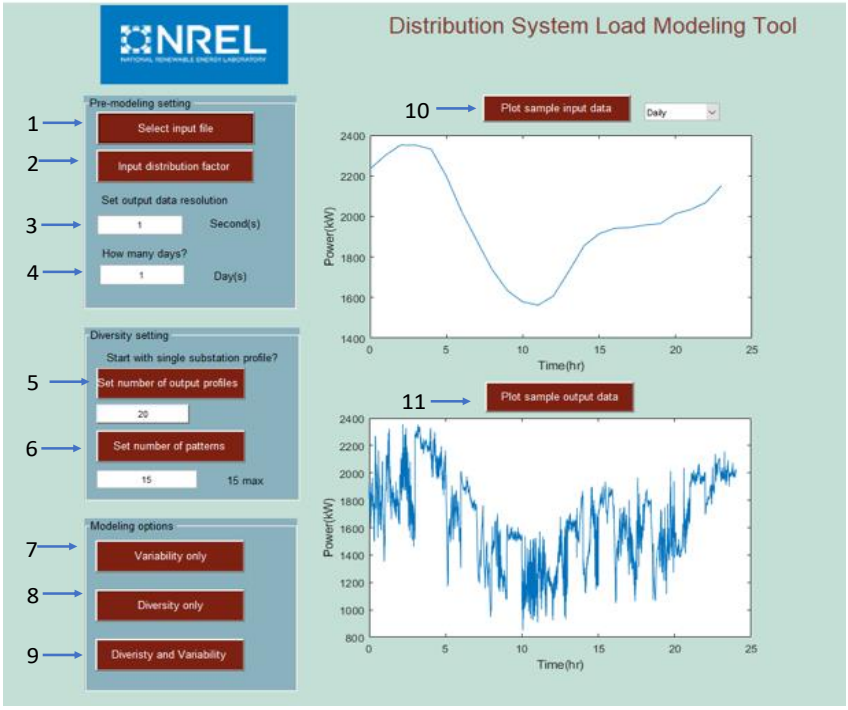


Figure 65. Screenshot of the Load Modeling Tool developed in MATLAB.

5.1.6. Visualization

Visualization techniques have been developed to facilitate the analysis of QSTS simulation results [116]. Once a QSTS simulation has finished, there remains the task of analyzing the results to understand the impacts of different DER or control strategies. After each power-flow solution, a new system state is reached with potentially tens of thousands of data points depending on the size of the circuit being solved, and millions of steady-state power-flows are solved during a single yearlong QSTS simulation. For every time step of the simulation, copious amounts of data can be probed from the simulation including voltages at every bus and node, loading information from power delivery elements (e.g. lines and transformers), active and reactive power injections from distributed generators or energy storage devices, controller states and time delays (e.g. regulator tap position or time until a switching capacitor changes state), and active and reactive power losses. Therefore, the ability to organize and visualize the results of QSTS simulations is critical for analytical purposes. It should be noted that all figures in the following subsections are shown for the test circuit "Feeder CO1," but can be generated for any circuit after a QSTS simulation has been executed using the open-source MATLAB code by simply passing the standardized "QSTS" output structure to one of the plotting functions (also included in the open-source code).

5.1.6.1. Circuit Plots

Conventional circuit plots based on GPS coordinates of devices show the geometry and orientation of a distribution circuit. These plots are useful for understanding where certain components are located with respect to one another and are typically used to show voltages or power flows at an instant in time, commonly referred to as snapshot analyses. While these methods are useful in certain cases, they cannot provide any insights into the time-dependent nature of the system.

With QSTS, circuit plots can be infused with additional data from simulation results, making them more powerful. Each circuit element in Figure 66 is assigned a unique symbol, making them easy to

see. For example, in Figure 66, clicking on the voltage regulator symbol shows the total number of tap position changes that occurred throughout the year, or clicking on a PV system shows the total amount of energy produced. Other information that can be accessed includes the number of capacitor state changes, feeder minimum and maximum active and reactive powers, line loading, etc. QSTS circuit plots can be generated by calling the "plotQSTS_Circuit()" function and sending in the OpenDSS circuit object and QSTS output structure as inputs.

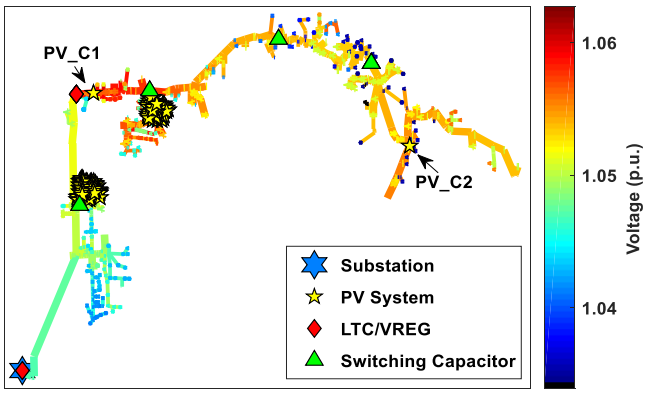


Figure 66. Feeder CO1 QSTS circuit plot with coloring based on the maximum voltage each node reached throughout the year

The lines in the circuit plot can also be colored according to QSTS results. In Figure 66, the lines are colored by maximum voltage. This figure reveals that the highest voltages were not recorded near the substation, but near the middle of the feeder, due to the high penetration of PV systems in that area. Line colors can also be assigned by other QSTS results like loading.

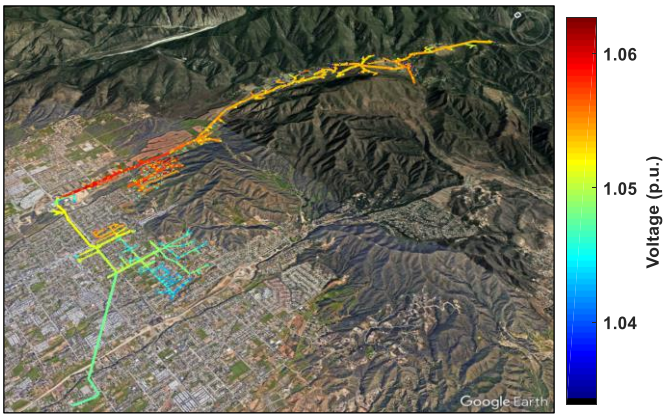


Figure 67. Feeder CO1 in Google Earth with coloring based on maximum voltage each node reached throughout the year

If the model contains real coordinates, the circuit can be plotted in Google Earth using color schemes based on QSTS results. The advantage of Google Earth is that the circuits can be plotted exactly where they are located, giving a better representation of the circuit. In many areas, the "street view" feature in Google Earth is available, making it possible to see actual photos of the circuit components, such as the pole-mounted transformers, voltage regulators, or capacitor banks. Figure 67 shows the same type of circuit plot as Figure 66 but in Google Earth.

5.1.6.2. Voltage Profiles Along a Feeder

Maintaining voltage levels between acceptable thresholds is crucial for distribution circuits. Typically, the voltage along a feeder drops as the distance from the substation is increased. In modern distribution circuits, there can be multiple devices that contribute to voltage regulation on a single circuit. Voltage profile plots show the feeder voltages as a function of distance from the substation at specific points in time. Symbols representing DER or voltage regulating equipment can be plotted on top of the voltage profile to explain what may be contributing to sudden changes.

It is also possible to take a closer look during times at which the circuit experienced high variability or extreme conditions, e.g. when the heaviest overloading occurred or when the feeder experienced its maximum and minimum voltages during the year. Since all the system states throughout the year are known, it is easy to "rewind" to any specific time, *t*. First, the regulator tap positions and switching capacitor states are set to where they were at time *t* and held constant in those positions. Then, the loads and generators are assigned their injections according to their profiles at time *t*. Lastly, the power-flow is solved. This procedure can be accomplished by calling the "goToQSTSTimeIndex( )" function followed by the "plotVoltageProfile( )" function.

The procedure described above was implemented to generate the voltage profiles in Figure 68 for two distinct time points. The top subplot represents the time point when the maximum voltage occurred anywhere on the feeder throughout the simulation, i.e. the yearly global maximum node voltage, and the bottom subplot represents the time point when the minimum voltage occurred anywhere on the feeder throughout the simulation, i.e. the yearly global minimum node voltage. The voltage magnitudes at each node are plotted as a function of the distance from the substation. The vertical dashed lines represent the service transformer and low-voltage secondary network that often has a large per unit voltage drop over a short distance. In Figure 68, the profile at maximum voltage shows that the voltage increased steadily from the substation and reached a maximum near a cluster of PV systems' PCC.

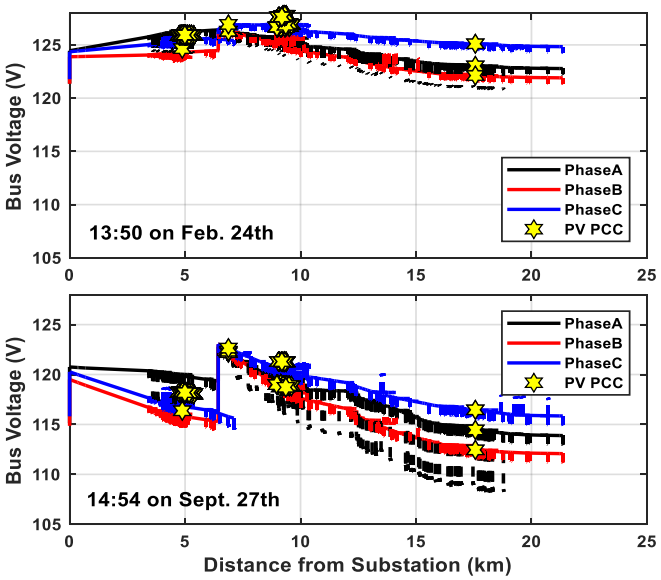


Figure 68. Feeder voltage profile at the time point when the yearly global maximum node voltage occurred (top) and at the time point when the yearly global minimum node voltage occurred (bottom)

5.1.6.3. Time-Series Data

In QSTS simulations, the converged state of a power-flow solution serves as the initial state of the next sequential power-flow. After each solution, data on the state of any circuit element can be collected. Plotting this data against the time point it was taken from gives a very detailed, time-series representation of the system states. Figure 69 shows an example of this, where the tap position of the substation transformer's LTC and tap position of three voltage regulators along the feeder are plotted against time. These types of plots are helpful in determining minimum and maximum values and understanding the relationships of various circuit elements, e.g. interactions between two voltage regulators or between a voltage regulator and a smart inverter with reactive power control. However, underlying trends may still be difficult to identify. These plots can be generated by calling the "plotQSTS_Timeseries()" function.

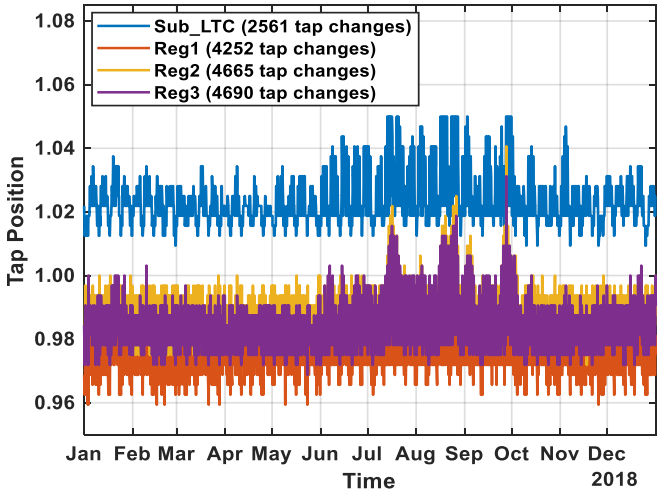


Figure 69. Regulator tap position time-series

5.1.6.4. Aggregated Time-Series Data

Data aggregation is one method for extracting useful information from time-series datasets and may help to identify the underlying trends in the data. The length of the aggregation window can be adjusted to any size based on the variability of the element being analyzed or the desired resolution. For QSTS simulations with a constant time-step, data aggregation is straightforward. First, the time-series data is reshaped into a matrix, then a function is applied along one dimension, e.g. taking the sum or finding the average value. For QSTS simulations with a variable time-step, the data must either be interpolated down to a constant time-step and reshaped or aggregated by looping through each window individually, which can be computationally intensive when using small window sizes.

By aggregating the regulator tap position time-series data (Figure 69) into monthly totals (Figure 70), it is easier to see how each regulator operates over time with respect to the others. For example, in January, "Reg1" had the most tap changes, but steadily declined until June when it had the least.

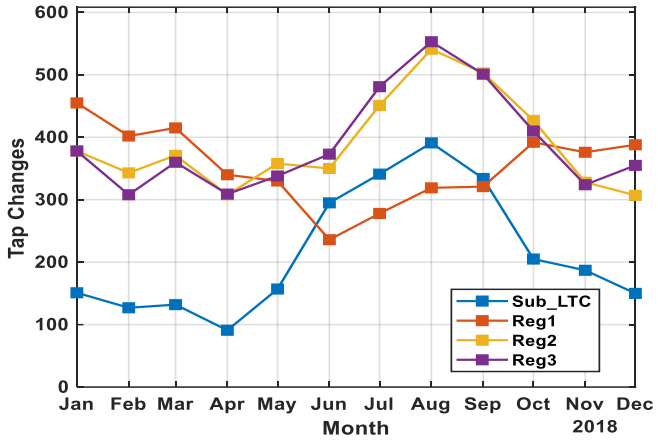


Figure 70. Monthly totals of tap position changes

Another example of data aggregation can be seen in Figure 71. This figure shows the monthly energy production of the two centralized PV systems. In terms of analyzing distributed generation, this type of plot could give insight into the various PV systems on the circuit, such as their sizes or relative tilt and azimuth angles. These figures can be generated using the "plotQSTS_AggregatedTimeseries( )" function.

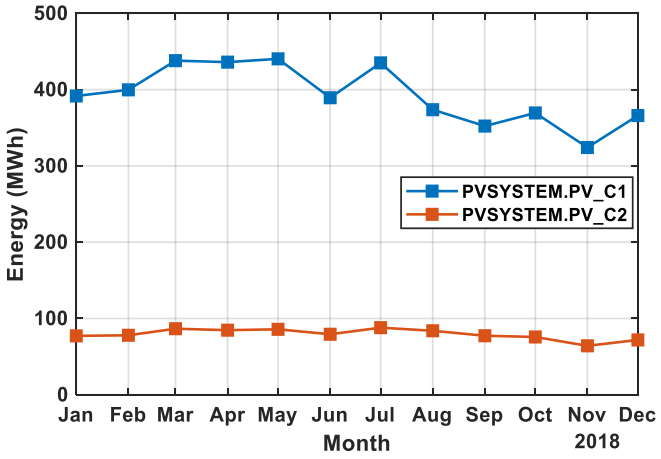


Figure 71. Monthly energy production of the two centralized PV systems

5.1.6.5. Temporal Raster (Heat Maps)

Time-series data can also be characterized in a temporal raster plot where the data is organized into a matrix and the color of each pixel represents its value for that point (or aggregated points) in time. By adding this extra dimension, the diurnal trends in the data manifest themselves through color. These plots could also be represented on three-dimensions axes, where the variable's magnitude is plotted on the third axis. Figure 72 shows an example temporal raster plot, using a one-hour aggregation window, of the energy production of the centralized PV systems with Volt/Var function turned on. While PV_C1 is roughly five times larger, the advanced inverter on PV_C2 consumed a greater amount of reactive power and did so more often. The specific impact of the Volt/Var function can be explored by looking at the feeder's time-series data and aggregated data. The addition of advanced inverters on the two largest PV systems helped to reduce the feeder's yearly maximum voltage by 0.0031 per unit, increase the feeder's yearly minimum voltage by 0.0048 per

unit, and decrease the number of operations of every voltage regulator and switching capacitor in the circuit.

Another benefit of QSTS analysis is the ability to quantify losses in a distribution circuit. Figure 73 shows a raster plot, using a 10-minute aggregation window, of the line losses in the circuit. Thus, each pixel represents the maximum line losses (in kW) that occurred over the aggregation window. This figure shows that the line losses were mostly less than 100 kW throughout the year, with the exception of a few days. Since a 10-minute aggregation window was used in Figure 73, the pixels appear much smaller and capture more detail than the 60-minute aggregation window used in Figure 72. These figures can be generated using the "plotQSTS_Raster( )" function.

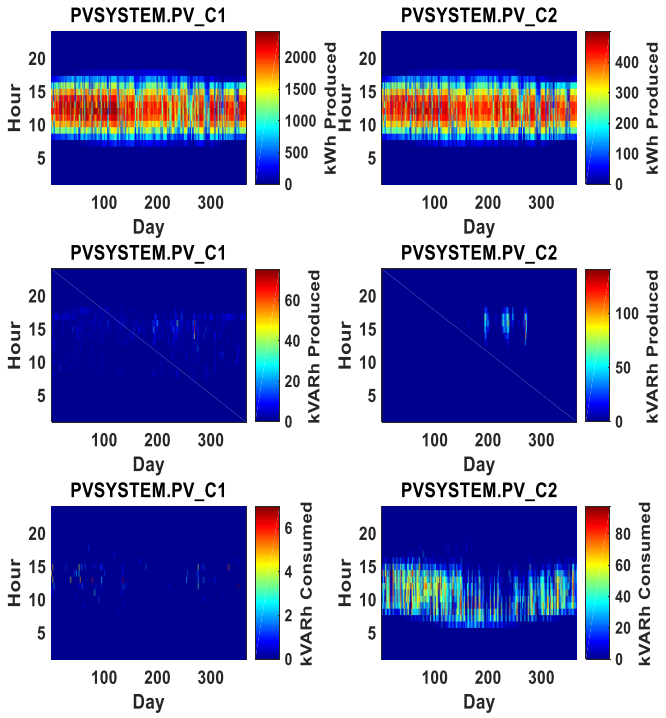


Figure 72. Temporal raster plot with one-hour aggregation of energy production real power produced (top), reactive power produced (middle), and reactive power consumed (bottom) of the two centralized PV systems

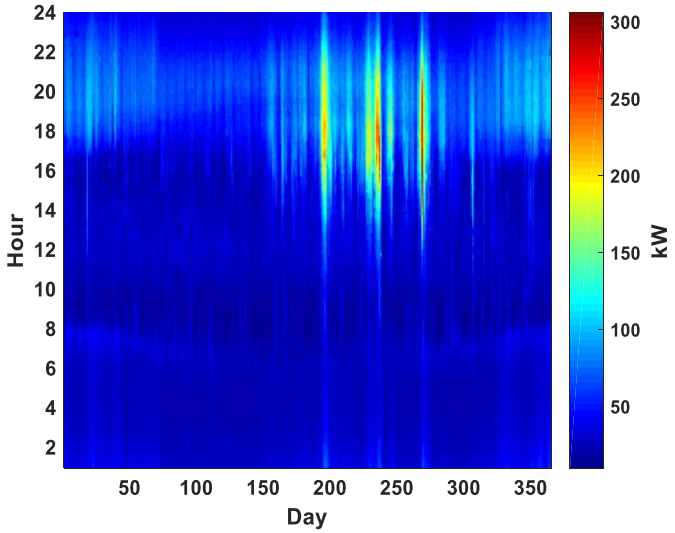


Figure 73. Temporal raster plot of line losses using a 10-minute aggregation window

5.1.6.6. Box Plots

In QSTS simulations, time-varying circuit components like load and PV generation each have profiles associated with them, consisting of a series of multiplier values. These time-series profiles, or loadshapes, are typically created from actual measured data and dictate how the components change through time.

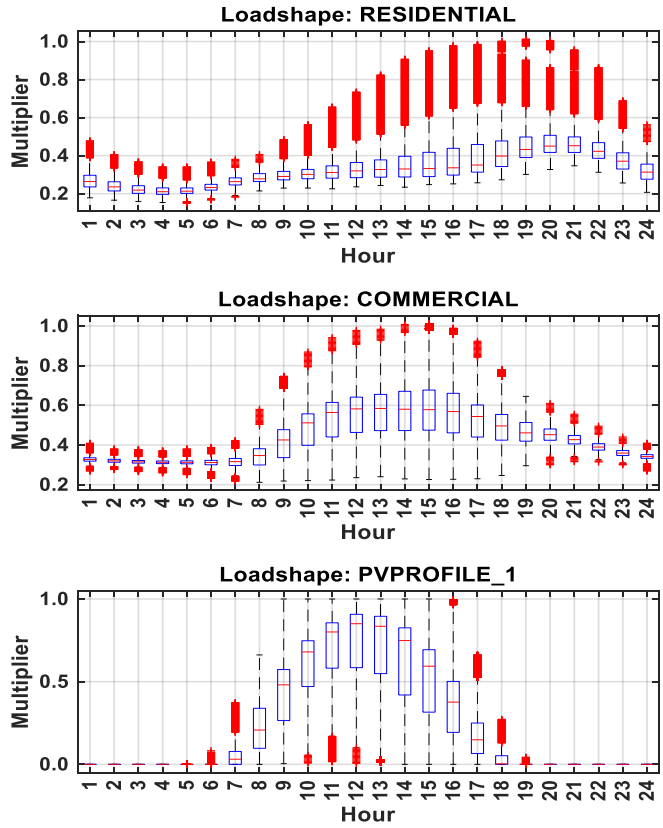


Figure 74. Box plot of 1-second resolution time-series profiles (1,314,000 data points per box)

Figure 74 shows the distribution of multipliers at each hour of the day for three different loadshapes: one for residential loads, commercial loads, and one of the PV profiles. The bottom and top edges of the boxes represent the 25th and 75th percentiles, respectively. The red line inside the box is the median value, while the whiskers extend to the most extreme data points not considered outliers (red crosses). All loadshapes used in these QSTS simulations represent yearly time-series data at a 1-second resolution for a total of 31,536,000 data points per loadshape. This data must first be reshaped before the box plots can be generated. Each hour of the day contains 3600 data points and the time horizon of these loadshapes was 365 days, so each individual box represents 1,314,000 data points. The time-series data can be reshaped in several different ways, depending on what information is important to the user. This would result in days or months being on the x-axis instead of hours. Each loadshape could also be represented in a single box, such that all loadshapes would appear side by side in a single figure. These figures can be generated using the "plotQSTS_BoxPlot( )" function.

5.1.6.7. Shaded Percentile Plots (Density Plots)

Shaded percentile plots can show how statistical distributions change over time. In these plots, the denser areas of the distribution appear as darker colors and the red line represents the median value. In Figure 75, each vertical slice represents the distribution of the daily minimum voltage each node recorded. These daily minimum values are then sorted to find the values of the various percentiles. For example, in Figure 75, the 75th percentile voltage on the first day of the year was 1.0126 per unit. So on January 1st, 75% of the nodes in the circuit had a minimum voltage of less than 1.0126 per unit. This figure also shows that the distribution of percentiles stays relatively constant throughout the year, except for a few periods of time in late summer. The benefit of this type of plot is that it gives insight into the severity of certain extreme circuit conditions. For instance, the lowest node voltage throughout the year, which occurred on Sept. 27th, was 0.9013 per unit—well below the predetermined threshold for this simulation of 0.95 per unit. However, on that same day, 90% of the nodes on the feeder remained above 0.9408 per unit.

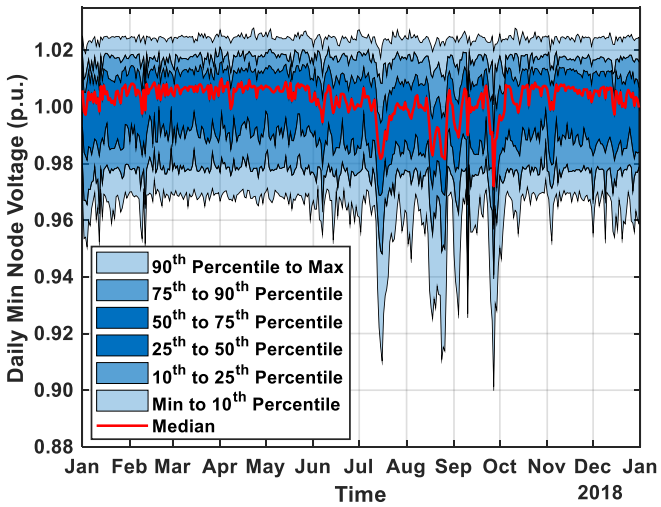


Figure 75. Distribution of daily minimum node voltages (5,469 nodes)

Shaded percentile plots are particularly useful when analyzing a large number of elements at once, such as the 5,469 nodes in Figure 75. Circuits with thousands of nodes would also have a large number of power delivery elements (lines and transformers) to connect those nodes. Thus, we can use shaded percentile plots to analyze the loading characteristics of power delivery elements as well.

Figure 76 shows the daily maximum loading of all 2,970 power delivery elements in the test feeder. In this case, overloading is not an issue. In fact, 90% of the power deliver elements experienced a maximum loading of less than 40%, which indicates (from a capacity standpoint) there is room for load and PV growth. These figures can be generated using the "plotQSTS_ShadedPercentile( )" function.

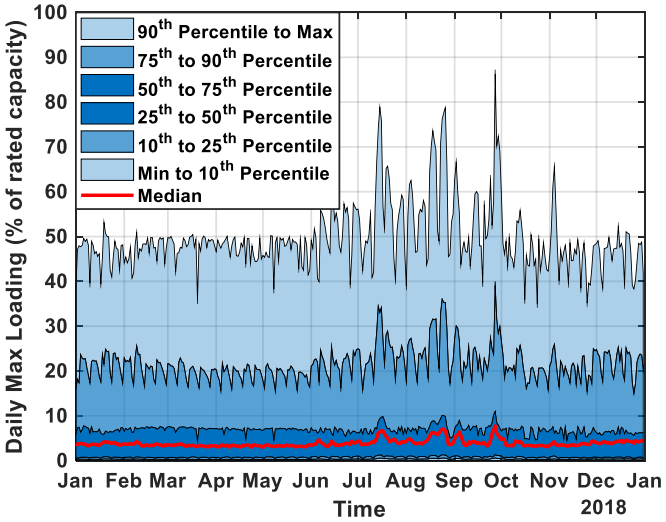


Figure 76. Distribution of daily maximum loading of every transformer and line in the test feeder (2,970 elements)

5.1.6.8. Cumulative Distribution Functions

Cumulative distribution functions, or duration curves, show the proportion of time for which a variable exceeds a certain level. For example, Figure 77 shows a duration curve for the real and reactive power of the feeder. For over 80% of the time, the feeder had more than 5 MW of power flowing through it, and it only had more than 15 MW of feeder power consumption for 0.1% of the time (less than 9 hours). These figures can be generated using the "plotQSTS_Duration( )" function.

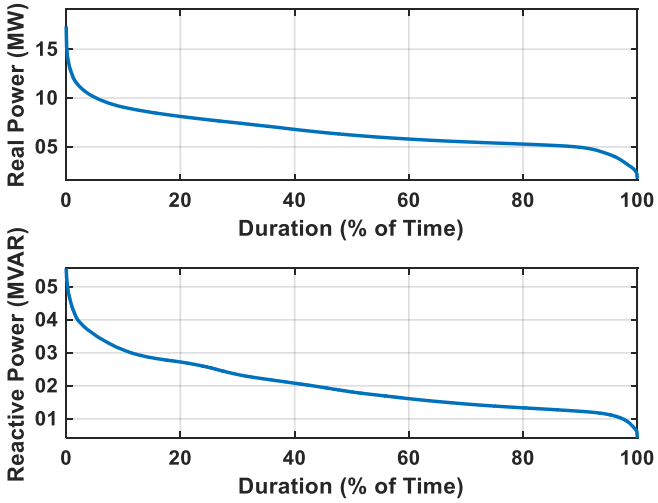


Figure 77. Duration curves of feeder real and reactive power into the feeder

Duration curves of maximum and minimum voltages can show how much time the feeder spent outside its predetermined operating limits, for example, ANSI C84.1. In Figure 78, the x-axis of the

duration curve is shown in terms of hours instead of as a percentage of total time. From this figure, it is clear that the maximum and minimum feeder voltages had very different characteristics. The feeder's minimum voltage reached violations of nearly 0.050 per unit, while the feeder's maximum voltage reached violations of only 0.015 per unit. However, the minimum voltage spent only 130 hours outside its threshold (intersection at x-axis), while the maximum voltage spent 195 hours outside its threshold. These figures can be generated using the "plotQSTS_Violation( )" function.

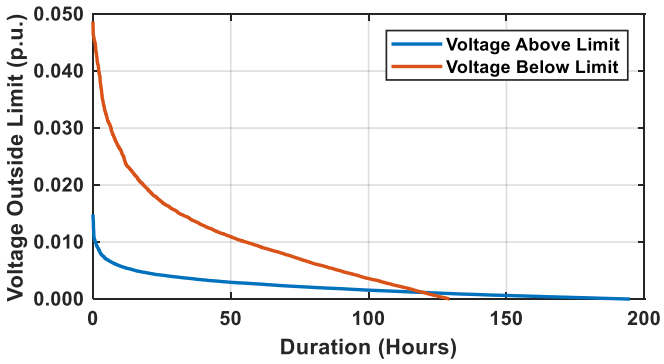


Figure 78. Duration curve of feeder voltage violations anywhere on the feeder

5.1.6.9. Controller States

One major benefit of QSTS simulations is the ability to model discrete controls and capture the time-dependent states of controllable elements like switching capacitors and voltage regulators. These expensive devices tend to operate more frequently in the presence of increased variability, such as in circuits with a high penetration of PV. Therefore, understanding their activity is a critical component of distribution system analysis. In Figure 79, the percent of time each capacitor spent switched on is represented as a stacked bar graph. This figure shows that "Cap2" was switched on for more than 95% of the time. Analysis of the data also shows that this capacitor switched states 146 times throughout the year. These results suggest that investing in a static capacitor bank near that location could be beneficial to reduce the number of operations of "Cap2."

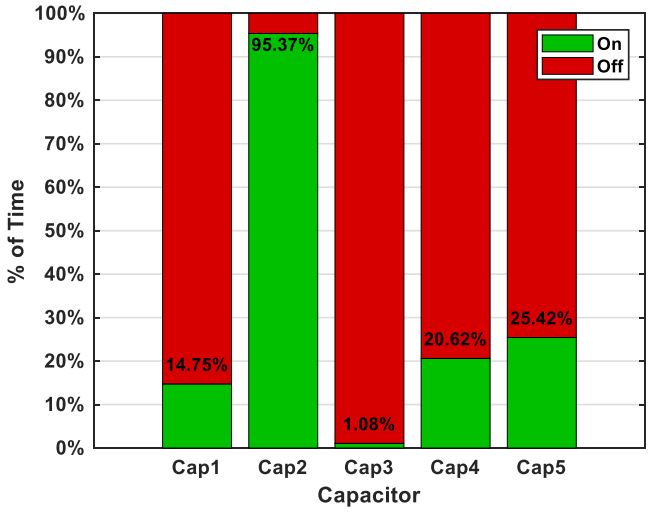


Figure 79. Stacked bar graph of capacitor states

Voltage regulators adjust their tap positions over time to help maintain line voltages within predetermined limits. Understanding how they operate throughout the year can help distribution

system engineers find ways to minimize maintenance costs or prolong the lifetimes of the devices. In Figure 80, the x-axis of the subplots shows the available tap positions of each regulator, and the y-axis shows the total amount of time spent in those positions. One interesting thing to note is that only the substation transformer's on-load tap changer (Sub_LTC) spent the majority of its time boosting the voltage. These plots can be generated using the "plotQSTS_CtrlStates( )" function.

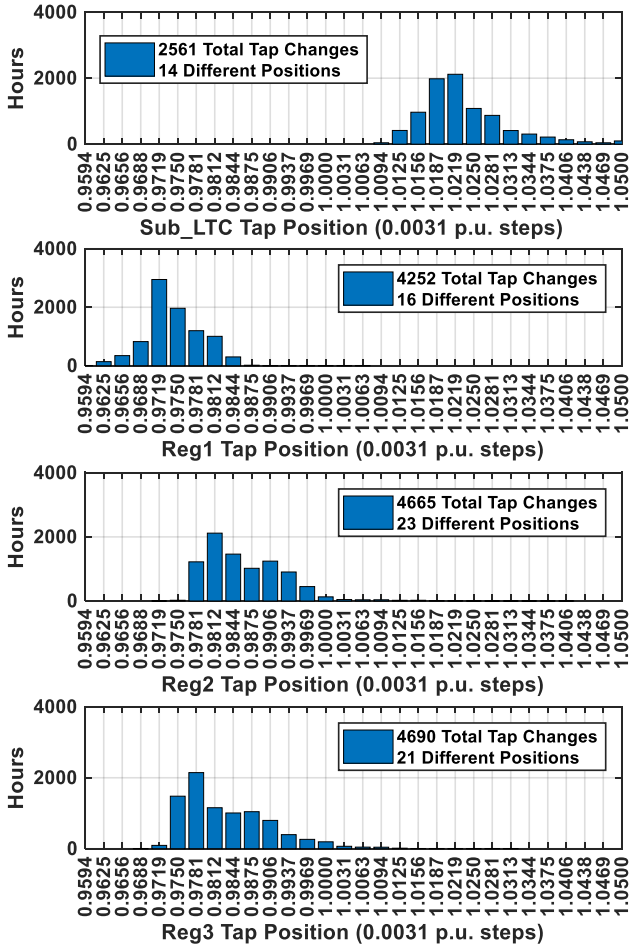


Figure 80. Total time spent at each tap position

5.2. OpenDSS

The following subsections highlight the improvements made to OpenDSS in support of reducing the computational burden associated with high-resolution QSTS simulations. The Open source Distribution System Simulator (OpenDSS) is a comprehensive system simulation tool for electric utility distribution systems. The program supports nearly all RMS steady-state (i.e. frequency domain) analyses commonly performed for electric utility distribution system planning and analysis. In addition, it supports many new types of analyses that are designed to meet future needs. Many of the features found in the program address the needs of distributed generation analysis, such as the ability to model the time-varying and time-dependent circuit components. OpenDSS also has several solution and control modes available to the user for performing QSTS simulations. By integrating the rapid QSTS algorithms into OpenDSS, the academic and research community that uses

OpenDSS can immediately see the benefits of faster QSTS simulations without having to run any external code.

5.2.1. Power Flow Solution Improvements

QSTS simulations require a power flow solution to be computed for each time step in the given time horizon. For high-resolution simulations, millions of power flow solutions will be performed. Several key improvements have been implemented in OpenDSS that address the minimum number of iterations required before convergence, add functionality to time reporting, and reduce hard drive interactions.

5.2.1.1. Reducing Minimum Iterations

To solve a power flow solution, OpenDSS uses a standard Nodal Admittance formulation of the circuit model that can be found documented in many basic power system analysis texts. This simple iterative solution has been found to converge quite well for most distribution systems that have adequate capacity to serve the load, which is the case for most distribution systems. While performing QSTS simulations, the solution at the present time step is used as the starting point for the solution at the next time step. Unless there is a large change in load, the solution will typically converge in 1 or 2 iterations. It has been found that the first guess at the next time step (when the time step size is small) is often good enough. Therefore, the *MinIterations* option has been added to OpenDSS to give the user the ability to reduce the default value of 2 iterations down to 1, and additional iterations will be executed automatically, if needed, for better convergence. In QSTS simulations, setting *MinIterations* to 1 results in a 1.7x speed improvement.

5.2.1.2. Solution Process Timing

Improving the computational speed of QSTS simulations requires an accurate accounting of all associated process timings. To address this concern, timer functionality has been incorporated in OpenDSS to record the time required for the different segments of the power flow solution. The power flow solution in OpenDSS involves three main stages: solution, sampling, and simulation. The solution stage solves the circuit for a single time step and updates the injected currents from loads and generators until the circuit solution converges. The sampling stage contains a routine for sampling the meters and monitors distributed by the user on the circuit. This action involves dynamic memory allocations and, possibly, disk file I/O. If there are no monitors/meters defined on the circuit this stage should not require much time.

There are three options for accessing the timer data using the *get* command. *ProcessTime* is a read-only value that measures the duration of the solution stage. *TimeStep* is a read-only value that measures the duration of the time step including the power flow solution and sampling all the meters and monitors placed on the circuit. Lastly, *TotalTime* is a cumulative value that captures the duration of a QSTS simulation, including both stages 1 and 2. The user can reset the simulation timer by issuing the "*set TotalTime=0"* command. The first two timers, *ProcessTime* and *TimeStep*, can be captured at each time step using monitors in mode 5, where the sum of *TimeStep* at each simulation step would equal the value stored in *TotalTime*.

5.2.1.3. Reducing I/O Operations

As mentioned in the section above, part of the power flow solution in OpenDSS contains a routine for sampling the *Monitor* and *EnergyMeter* elements present in the circuit model. The *Monitor* element

has long employed in-memory file streams to make this operation as fast as possible. However, the *EnergyMeter* element would write up to 5 text files to disk at each time step, adding significant overhead to QSTS simulations. The *EnergyMeter* element has been modified to write samples directly into RAM in native binary format using byte streams, virtually eliminating I/O operations to the hard drive during the simulation. The content of the memory is then flushed to the hard drive in text form once the simulation is over, generating the same files that the earlier version of OpenDSS would deliver. Upgrading the *EnergyMeter* element to use the in-memory byte stream has been shown to have a computational time reduction in 89.3% of simulation steps. The user can now also override the default behavior of sampling the *EnergyMeter* element at the end of the solution loop with the *SampleEnergyMeters* options.

5.2.2. Multi-Rate Control Mode

The "multi-rate" control mode aim to reduce the systematic error added when performing QSTS simulations at different step sizes, keeping the simulation fidelity in accordance with the QSTS parameters set by OpenDSS users [117]. The *controlmode* feature in OpenDSS defines the way control actions will take place during the simulation. This is also conditioned by the simulation step size chosen by the user when performing the simulation. As a result, the combination of these parameters can speed up the simulation but at the same time, introduce a considerable error when performing time-based simulations. For example, if a simulation is performed in OpenDSS using the *time* control mode and 1-s step size, the results will differ in a certain percentage for the same time instants when compared with a simulation performed using 1-minute step size. However, this situation depends directly on the need for control actions during the simulation and if these control actions will generate new actions to be performed in later simulation steps.

In classic control modes, the control actions are performed after a simulation step solution is performed, which causes the control actions generated by the latest control actions performed to be executed in the next solution step. This will delay the effect generated by the control actions resulting in an error between simulations performed at different step sizes. The multi-rate control mode deals with this issue by solving all the control actions that fit within the time window determined by the simulation step size.

With the multi-rate control mode, OpenDSS will check for the control actions within the time window defined by the step size. Once a power flow solution is performed and the control devices sampled within the normal simulation cycle of OpenDSS, the solution algorithm will enter the *do control action* routine. When sampled, the control devices will store their next control actions into a queue called the *control queue*. Then, these control actions will be executed by the *do control action* routine if the control action was programmed to be executed before the next time step. However, as mentioned previously, this control action may trigger new control actions, which are also stored in the *control queue* to be performed after the next solution step. As a difference with respect to the existing control modes, the multi-rate control mode will evaluate whether the new control actions generated by the latest actions executed fit within the actual step size. If so, OpenDSS will solve the system and sample the control devices looking for control actions still within the time window. This process will be performed iteratively until there are no more control actions to perform or when the next control action in the control queue exceeds the time window. In this case, the control action will remain in the control queue to be performed in the next solution step.

One major benefit of the multi-rate control mode for rapid QSTS simulations is its ease of use. Once OpenDSS is set up to run a QSTS simulation, the user only needs to change the *controlmode* setting to *multirate* and adjust the *timestep* setting to a coarse value (e.g. a value greater than or equal to

the longest time delay of any controllable device). Overall, multi-rate control mode enables speed improvements for QSTS simulations by maintaining simulation fidelity at coarse time steps, particularly on circuits with multiple controllable devices.

5.2.3. Circuit Reduction

In OpenDSS, circuit reduction can be performed on a compiled circuit in memory via the *Reduce* command. Being that the reduction algorithms are implemented as part of the *EnergyMeter* object, the user can choose to reduce the entire circuit at once or reduce specific sections of the circuit by placing energy meters accordingly. Additionally, the circuit reduction algorithm can be modified through several available options using the "*Set ReduceOption*=…" command. To prevent specific buses from being eliminated, the user can mark those buses using the *Keeplist* option.

The reduce *ShortLines* option merges line objects with less positive-sequence impedance than the value specified by the *Zmag* option with the downline Line object, meaning the bus between the two lines is eliminated. *Zmag* may be set on the same command line as a *ReduceOption* command. In most cases, the *Load*, *PVSystem,* and *Storage* elements on the eliminated bus are moved to the upline bus of the merged line. If the identified short line is at the end of a branch, the load elements are moved to the downline, or "to", bus. If the bus that would be eliminated is marked as a 'keep' bus or has either a shunt capacitor or shunt reactor, no merge is performed. Similarly, if the short line identified has a Control element or is being monitored by another element, no merge is performed. Thus, controls and monitors are preserved.

The reduce *Laterals* option can be used to quickly eliminate all 1-phase laterals to achieve a dramatic reduction in node count on circuits where the main planning interest is on the 3-phase sections. If the *KeepLoad* option is set, all loads and other shunt elements are moved to the corresponding node on the bus at the head of the lateral, and voltage bases are adjusted, if necessary. It is possible, and likely, that there could be dozens of Power Conversion (PC) elements on the same node. This is not a conceptual problem for OpenDSS because each element simply contributes its compensation current to the main current array and there is no limit on the number of PC elements and other shunt elements that may be connected to a node. While there will be some overhead in the solution algorithm to collect all the individual current contributions, this will likely be offset for large circuits in which hundreds of 1-phase buses and corresponding nodes are eliminated and removed from the System Y matrix. One downside of this reduction algorithm is that the losses in the single-phase lateral are no longer represented in the analysis.

Controlled and monitored elements are removed or moved when using the *Laterals* option, which is the opposite of what happens in the reduce *ShortLine* option. Any 1-phase Capacitor or Reactor elements present on the lateral are also moved and their contribution to reactive power is preserved. These elements are linear, so they are completely modeled in the System Y matrix and do not contribute to the main current array. In the future, it may be possible to merge all single-phase PC elements of the same class to save computational effort, but there are complications with this with respect to OpenDSS loadshapes, irradiance shapes, and storage controllers.

5.2.4. OpenDSS-PM

The classic OpenDSS program is a simulation platform built for execution in a single, sequential process. Each procedure/function is called from each object sequentially to perform a QSTS simulation. The performance that can be achieved is based on the structure of the low-level routines, the simplicity of the routines, and the efficiency of the compiler.

In order to improve the speed of OpenDSS, EPRI has evolved OpenDSS into a more modular, flexible, and scalable parallel processing platform called OpenDSS-PM (Now OpenDSS v8 and later) based on the following guidelines:

1. The parallel processing machine will be interface-independent
2. Each component of the parallel machine should be able to work independently
3. The simulation environment should deliver information consistently
4. The data exchange between the components of the parallel machine should respect the interface rules and procedures
5. The user handle of the parallel machine should be easy and support the already acquired knowledge of OpenDSS users

To create the parallel machine, OpenDSS-PM uses the actor model [118-121]. Each actor is created by OpenDSS-PM runs on a separate processor (if possible) using separate threads and has its own assigned core and priority (*real-time* priority for the process and *time-critical* for the thread).

The interface for sending and receiving messages from other actors will be the one selected by the user—either the COM interface, the Direct DLL API, or a text script using the EXE version of the program. From this interface, the user will be able to create a new actor (instance), send/receive messages from these actors, and define the execution properties of the actors such as the execution core, simulation mode, and circuit to be solved, among others. This concept is shown in Figure 81.

Using the existing interfaces, the user can do the same things he can do with the classic version plus the operations related to parallel processing, such as:

1. Request the number of available cores and the number of physical processors available.
2. Create/Destroy actors
3. Execute the simulation of each circuit concurrently and in parallel (hardware dependent)
4. Assign the core where the actor will be executed
5. Modify the simulation settings for the active actor
6. Set the name of the circuit that will be simulated

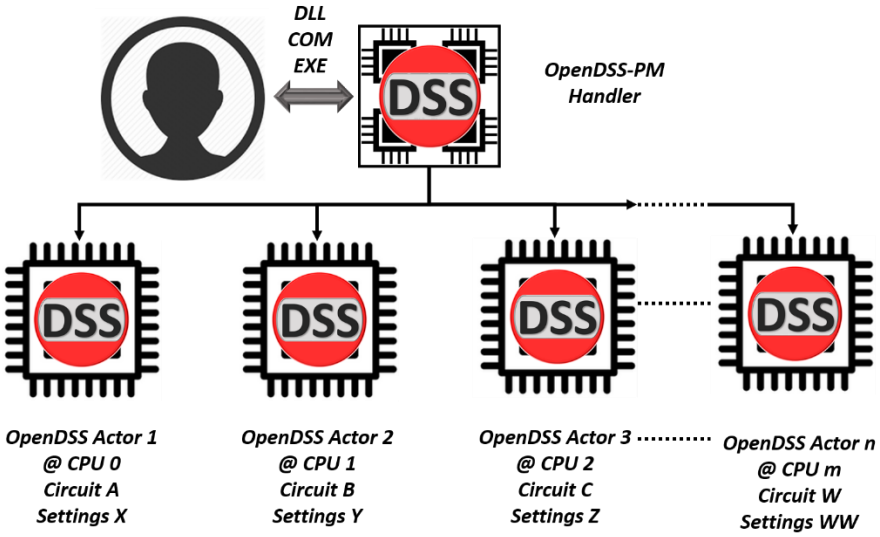


Figure 81. Operational scheme proposed for evolving OpenDSS into a parallel processing machine based on actors

As can be seen in Figure 81, the interface will work as the communication medium between the different actors on the parallel machine. This feature enables several simulation modes inspired by parallel computations such as temporal parallelization and Diakoptics, among others. These types of simulations will be driven by an external program that will handle the actors within the parallel machine, in keeping with the actor concept as a *message-driven state machine*.

5.2.4.1. Temporal Parallelization

Temporal parallelization of QSTS simulations can be performed using the functions within OpenDSS-PM, by assigning different sections of the desired time horizon to be solved in parallel by multiple actors. The user can check to see how many cores and CPUs are available for a parallelized simulation by issuing the "*get NumCores"* and "*get NumCPUs"* commands, respectively. To prevent the computer from becoming unresponsive, it is recommended that at least one CPU is left unused by the simulation.

Once a circuit has been compiled, the user can issue the *clone* command to copy the circuit onto additional actors. After the new actors have been established, the *ActiveActor* will be set to 1 by default, meaning any commands issued will be directed to that specific actor. The user can send commands to any individual actor, *N,* by issuing the "*set ActiveActor=N"* followed by any other desired command. The user can also send commands to all actors at once by using "*set ActiveActor=**." So, any settings common to all actors, like *mode* and *controlmode*, can be assigned all at once, followed by individual settings for each actor, like the start time or assigning its corresponding CPU.

After each actor has been properly initialized for a QSTS simulation, the user can reset the *ActiveActor* to 1, set "*Parallel=true*," and call *SolveAll*. While the simulation is underway, the user can check the progress of the actors by using the *ActorProgress* function. The user can also elect to call the *Wait* function, which will block any new commands from executing until all actors have finished their assigned solutions. By using this function, it is probable that the user may not be able to see updates on the summary/results tab, but it is a very good mechanism for synchronizing all the actors within the parallel environment and make classical scripts compatible with OpenDSS-PM.

Upon completion of a parallelized QSTS simulation, the user can set the *Concatenate*Reports option to enable/disable the report concatenation of the monitor's content. When disabled (default) the user needs to specify the actor to gain access to the monitor's data using the commands show or export. When enabled, this option allows summarizing all monitor content with the same name working at different actors into a single file.

5.2.4.2. Diakoptics

Diakoptics is a technique for tearing large physical circuits into a number of sub-circuits to reduce the modeling complexity and accelerate the solution of the power flow problem using a computer network. Each computer will handle a separate piece of the circuit to find a total solution. Using modern multi-core computers, this technique can be used for accelerating QSTS simulations by using the actor model as a framework for coordinating the interactions between the distributed pieces proposed in Diakoptics. Actor-based Diakoptics, or A-Diakoptics, seeks to simplify the power flow problem to achieve a faster solution at each simulation step. Consequently, the total time reduction when performing QSTS will be evident at each simulation step with A-Diakoptics. This technique has been implemented in OpenDSS-PM and includes automated circuit tearing and other tools to execute and debug the method step by step. An important feature of the simplified A-

Diakoptics algorithm proposed here is that even small-scale circuits will obtain simulation time reduction

To activate A-Diakoptics, there must already be a circuit compiled into memory in Actor 1 (default) and the circuit needs to be solved once in *snapshot* mode. When enabled, A-Diakoptics will partition the system using the number of sub-circuits specified with "*set num_subcircuits=NS*", where *NS* is the number of sub-circuits. Then, the sub-circuits will be compiled and loaded into memory for the simulation. After activating A-Diakoptics, OpenDSS-PM will deliver an output describing the result of the initialization including statistics to help the user to understand and estimate what to expect from the circuit tearing. Each compiled sub-circuit will reside in a subfolder inside the original circuit model folder in the hard drive. OpenDSS-PM will create a subfolder called *Torn_Circuit* that contains the models for each zone and the interconnected circuit separately.

At this point, the system is ready to solve in A-Diakoptics mode. To solve the circuit, make sure that the active actor is Actor 1, then set the simulation mode and the simulation features for QSTS accordingly. Finally, execute the *solve* command. The following considerations need to be fulfilled:

1. No monitors or energy meters can be created in Actor 1. Actor 1 is the simulation coordinator and it hosts Y4 and the vectors for calculating Ic. Actor 1 also hosts the admittance matrix of the interconnected circuit. The total voltages are transmitted to Actor 1 from other actors when the simulation step is completed, however, control actions are performed on the sub-circuit actors, making it difficult to monitor those actions from Actor 1. This is a feature on which development is continuing. Nevertheless, voltages, powers, and currents can be extracted from Actor 1 to visualize the variations in the system. The control actions can be monitored using the sub-circuit actors.
2. Actor 1 is the simulation coordinator; the others are slaves of the coordinator.
3. The number of actors depends on the number of sub-circuits configured by the user. If the number of sub-circuits set by the user surpasses the number of CPUs – 2 in the local PC, the Initialization algorithm will force the number of sub-circuits to the number of CPUs – 2.
4. The CPUs are assigned automatically. If the user wants a better performance (e.g. 1 thread per core) it is necessary to do the redistribution manually.
5. The algorithm is only implemented for the standalone OpenDSS.EXE version; once there is a stable version it will be implemented in the other interfaces.
6. DO NOT use the wait command when using A-Diakoptics, the simulation coordinator will use it internally with some variations. Instead, use the *get ActorProgress* command to check the simulation progress.

5.3. CYME

The R&D executed throughout the QSTS Project by CYME and the project partners resulted in numerous key findings and improvements allowing greatly accelerated QSTS simulations. Some of these improvements were later implemented in existing (CYME 8.1) or future commercial versions of the CYME power system analysis software.

Newton-Raphson Unbalanced power flow engine

The efficiency of CYME's Newton-Raphson Unbalanced (NRU) power flow engine has been improved by around 55 to 60% for networks without voltage regulators and transformers with load tap changers (LTCs), and up to more than 80% for networks with multiple voltage regulators and

transformers with LTCs. Improvements ranged from augmented flat start initialization techniques to improved code optimization, calculations, and models.

Voltage Drop Unbalanced power flow engine

While most of the early focus of this project was on improving the NRU power flow algorithm, the implemented accelerated QSTS solution also works with CYME's Voltage Drop Unbalanced (VDU) power flow algorithm. An improvement was made for the convergence of systems with smart inverters with advanced control functions (damping).

Load Flow with Profiles module

It was initially assumed that the accelerated QSTS solution in CYME would be based on CYME's existing Load Flow with Profiles (LFwP) module. This ended up not being the case, but multiple improvements had already been made to LFwP in the first year of the project. These improvements will still benefit users running the LFwP module, for instance for batch power flows. Improvements ranged from improved initialization, enhanced data flow, and allowing single iteration solutions using NRU.

Long-Term Dynamics

It was decided during the course of the project that the accelerated QSTS simulation in CYME would use the existing Long-Term Dynamics (LTD) module. Several modifications were made in the final 2 years of the project to allow for use of the LTD module. Improvements ranged from new timer functions, new initialization solutions, enhanced data exchange, and implementation of variable time step approximation methods.

5.3.1. Faster Power Flow Solver

Computing a power flow solution in a commercial-grade environment does not only consist of the solution of the power flow equations. Database access, data exchange between the solver and the main application, initialization, and reporting of results all add up to a significant part of the computational time. Significant savings in CYME (sometimes more than 60%) are reported in [122] for the solution of individual power flows by optimizing various aspects of the algorithm. When applying these improvements to QSTS simulation, 10-fold reductions of the total simulation time have been observed. Although significant improvements have been made around the power flow solver and its implementation, there is still a need for additional speed improvements outside of how individual power flows are solved. The number of time-steps is still a challenge in speeding up the QSTS simulation.

The objective of this task is to reduce the CPU time necessary to execute a single power flow using CYME's Newton-Raphson Unbalanced (NRU) engine. To clearly demonstrate the results of this research, a new and incremental version of CYME (e.g. CYME-QSTS.v01) is created for each improvement, i.e., each new version includes all the previous improvements. Figure 82 shows fifteen versions that were created. And shows the CPU time to solve a single power flow of the base-case CKT5 network (2998 nodes) for each version. This total CPU time is further split into segments of the code represented by the colored lines. The major changes affecting the CPU time are displayed in Figure 82

The total CPU time to solve 1440 time-series power flows of a modified IEEE 34 Node Test Feeder using LFP is presented in Figure 83; segment splits are also included. The major changes affecting the CPU time are also displayed in Figure 83. The modified IEEE 34 Node Test Feeder includes five PVs with high-resolution data (1 s) and a specific lower-resolution profile (5 min.) per load.

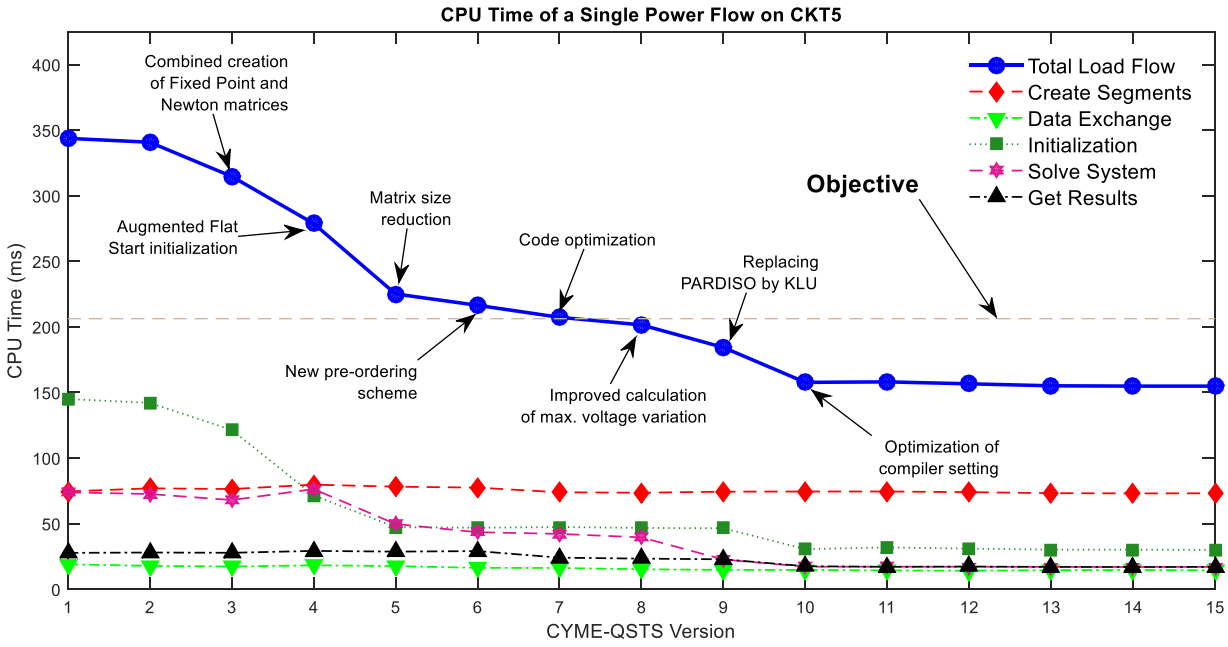


Figure 82. Simulation time for a single power flow of CKT5.

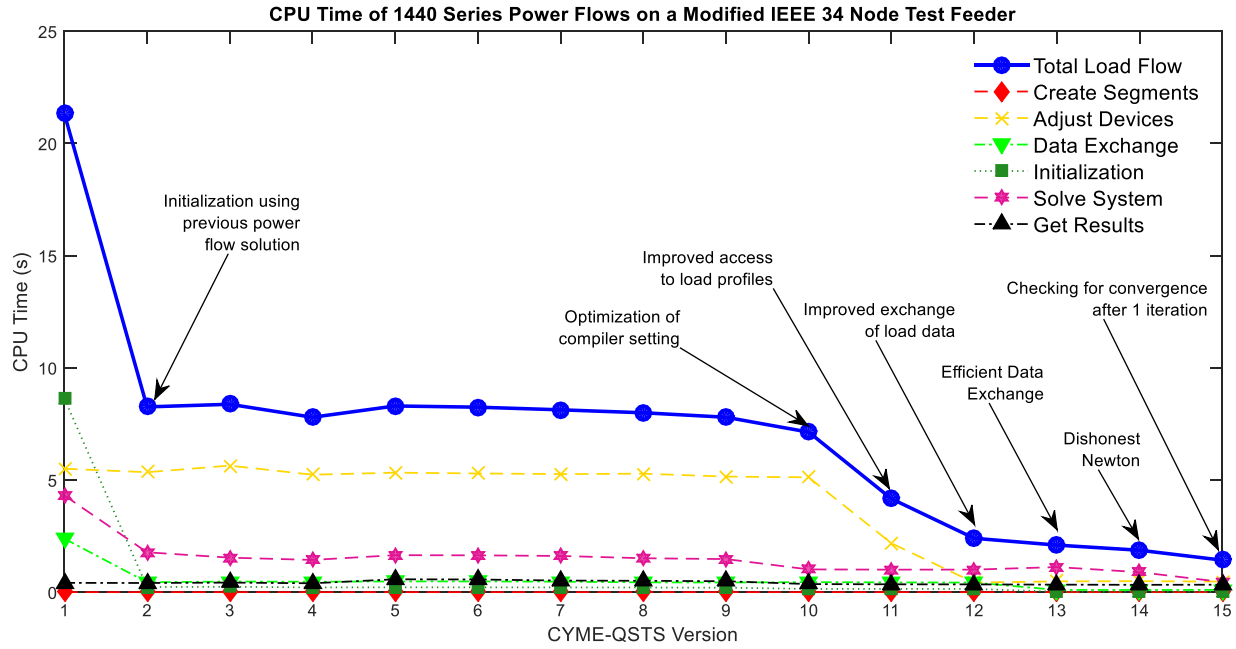


Figure 83. Simulation time for 1440 power flows of a modified IEEE 34 Node Test Feeder using LFP.

The CPU times of the 4 networks for a single NRU power flow are presented in stack bar form for the first and last CYME-QSTS versions in Figure 84. For each network, the CPU times are normalized with respect to the initial CPU time (with CYME-QSTS.v01). For a given version, the proportion of the timings of the different segments are fairly consistent between networks.

All the reported CPU times include the "Create segment" part of the code. It is necessary to execute this part of the code to solve a load flow; however, "Create segment" is also necessary to execute any

type of analysis in CYME. It could therefore be seen as computational overhead intrinsic to commercial-grade software such as CYME. If we remove the "Create segment" section from our CPU times, we achieve the following reductions:

- CKT5: 70%.
- IEEE 123 Node Test Feeder: 70%.
- IEEE 8500 Node Test: 69%.
- Large customer network: 64%.

The improvements in LFP are even more important. For the IEEE 34 Node Test Feeder (1440 time steps), the CPU time decreased from 21.32 to 1.43 s, which is equal to a reduction of 93%.

The improved LFP analysis was also tested on other networks. The following reductions were achieved (again with 1440 time steps):

- Modified IEEE 123 Node Test Feeder: 93%.
- Modified CKT5: 79%.

The CPU times of the 3 networks for the 1440 series power flows are presented in stack bar form for the first and last CYME-QSTS versions in Figure 85. For each network, the CPU times are normalized with respect to the initial CPU time (with CYME-QSTS.v01). It is interesting to note that the "Get results" segment takes a good proportion of the CPU time with CYME-QSTS.v15. Significant savings could be obtained by only querying and sending back the information of interest to the user (e.g., number of tap changes).

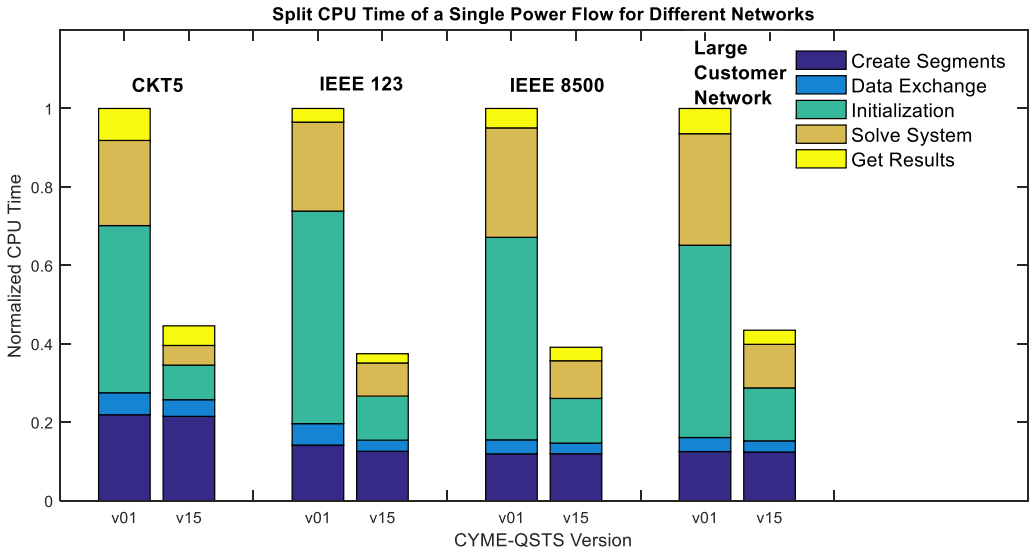


Figure 84. Normalized CPU time of a single power flow for different networks with CYME versions 1-15.

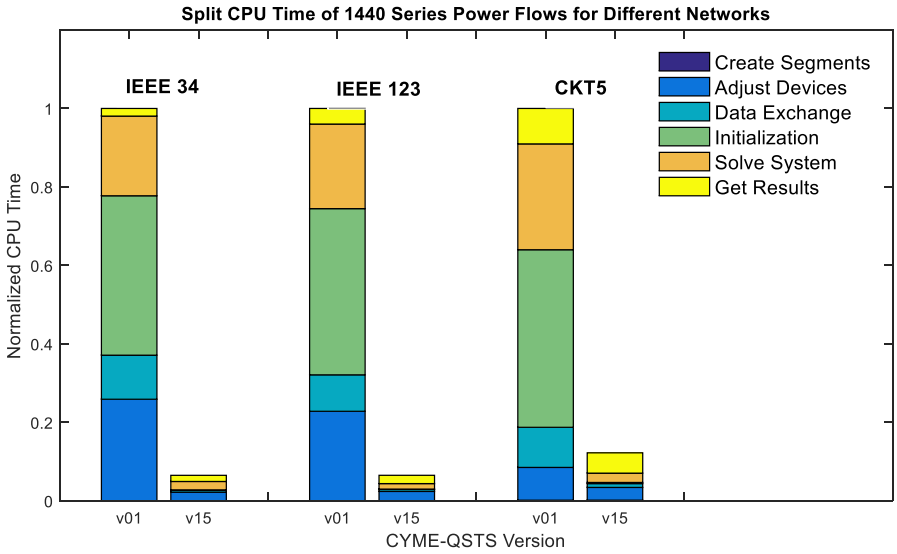


Figure 85. Normalized CPU time for 1440 series power flows for different networks with CYME versions 1-15.

5.3.2. Circuit Reduction

Quasi-static time-series (QSTS) simulations are used to understand the long-term behaviors of distribution systems, which involve a series of load-flow solution of the distribution feeder. This analysis is also useful for distributed energy resource (DER) planning. In general, a QSTS simulation has a small time-step (typically in seconds) and a long duration (typically in hours up to years) to simulate. When the distribution feeder contains thousands of buses/nodes, it can take a lot of computation efforts to carry out the simulation.

The objective of this part of the project is to implement a circuit reduction technique that is applicable to QSTS simulations. The reduction procedure will handle circuit configurations such as multiphase connections (i.e., 3 phases to single-phase), unbalanced loads, mutual coupling between lines, line charging, etc. The reduction procedure needs only to be performed once before running QSTS simulations.

One possible solution is to reduce the number of buses in the distribution feeder to reduce the simulation times, while differences in bus voltages, currents, and powers are minimized between the original and the reduced circuits. In CYME, a network reduction tool already exists. The tool derives a reduced circuit by linearizing the circuit around one operating point based on load flow solutions. However, the approach is less applicable to QSTS studies, as the operating points (time-series data of load profiles) could vary significantly in QSTS simulations and the reduced circuit needs to dynamically adapt to any change in operating points (different load profiles). Finally, the error differences between the original and the reduced circuits should be minimal.

The circuit reduction approach is based on the algorithm developed by Pecenak et. al [123], with some modifications in order to include line charging and to improve reduction accuracy. A CYME design including an adapted version of [123] for commercial implementation was produced in April 2018.

As of September 2018, the circuit reduction technique was not directly implemented in the CYME code. However, it has been implemented as a Python script/prototype, which can open and modify

CYME networks, yielding reduced networks in the desired format. The prototype can also automatically preserve the circuit main line if desired, which will create a larger network but allow monitoring abnormal conditions on a larger number of sections and devices. The main line can be defined and customized using CYME's existing "Main Line Detection" feature, which is shown in Figure 86.

The reduction algorithm starts by topology detection of the original circuit and specifies a set of critical buses (CB), which are the buses that should not be reduced from the original circuit and form the main tree in the reduced circuit. Then all buses or laterals that are off the main tree and do not include any CB are removed, and the connected loads are aggregated to the closest bus on the main tree. The final step is to reduce all buses that are between CB on the main tree and allocate the corresponding loads onto the adjacent CB according to line impedances.

To quickly highlight the benefits of the proposed method, a test circuit from the Western United States with medium PV penetration is considered. The number of nodes and different device types of the original (unreduced) circuit are presented in Table 14. The table also contains the same information for two reduced versions of the given circuit: with and without the main line. It is seen that the number of nodes is considerably reduced, by factors of 7 and 18 with and without the main line, respectively. It is also observed that the number of PVs is identical. This is a design choice, wherein we decided not to aggregate generators (and motors). It would however be possible, under certain circumstances, to aggregate generators similarly to what is done for loads. Finally, an overview of the one-line diagrams of the three versions of the network is presented in Figure 87.

Table 14. Properties of an original test circuit and its reduced equivalents (with and without the main line).

	Network		
	Original	Reduced (With Main Line)	Reduced (Without Main Line)
Nodes *	1937	277	108
Lines/Cables	1932	273	104
Photovoltaics	50	50	50
Protective Devices	529	6	2
Regulators	3	3	3
Spot Loads	862	170	65
Transformers	1	1	1

* Here, and throughout this document, we use CYME's definition to count nodes. In CYME, a series section always contains 2 nodes; however, a section may contain up to 3 devices (e.g., a line, a switch, and a regulator). This results in a higher number of effective nodes than what is counted by CYME.

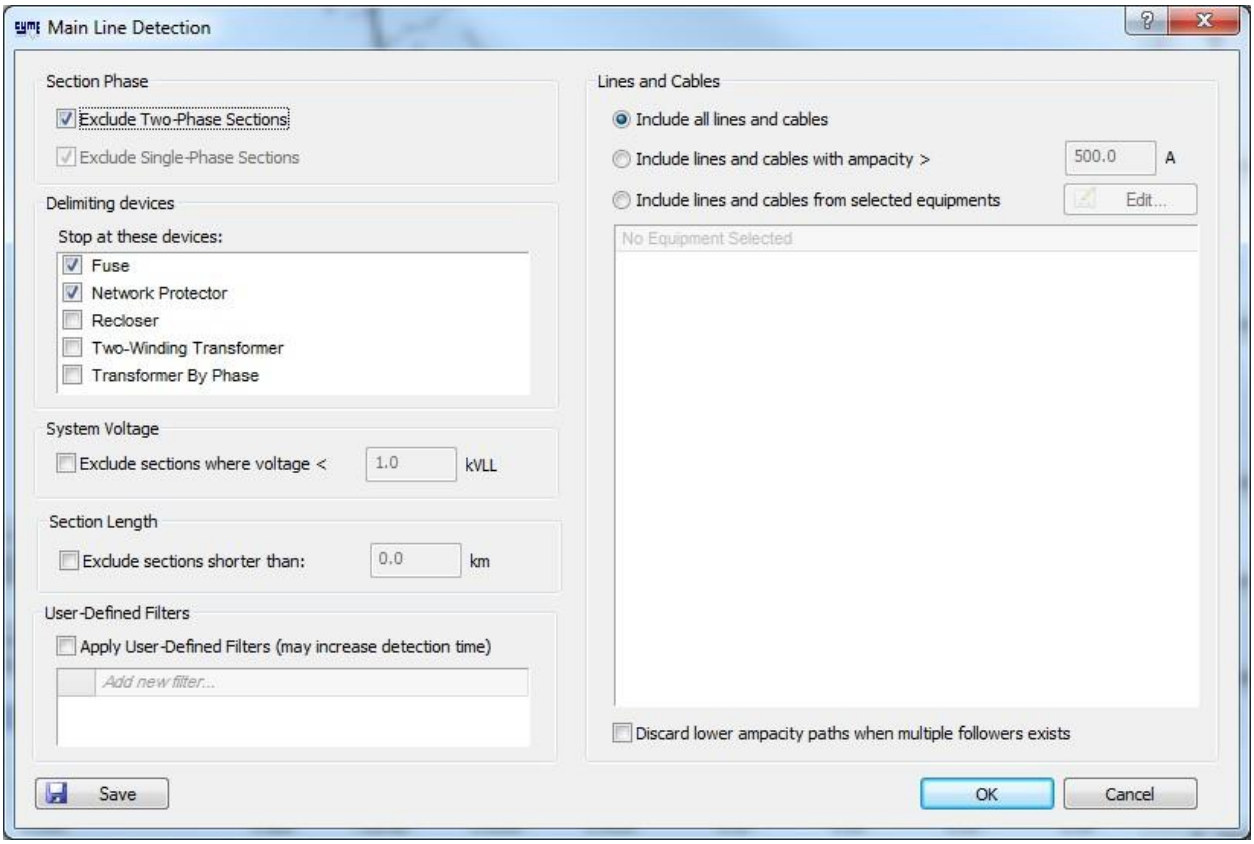


Figure 86. Main Line Detection

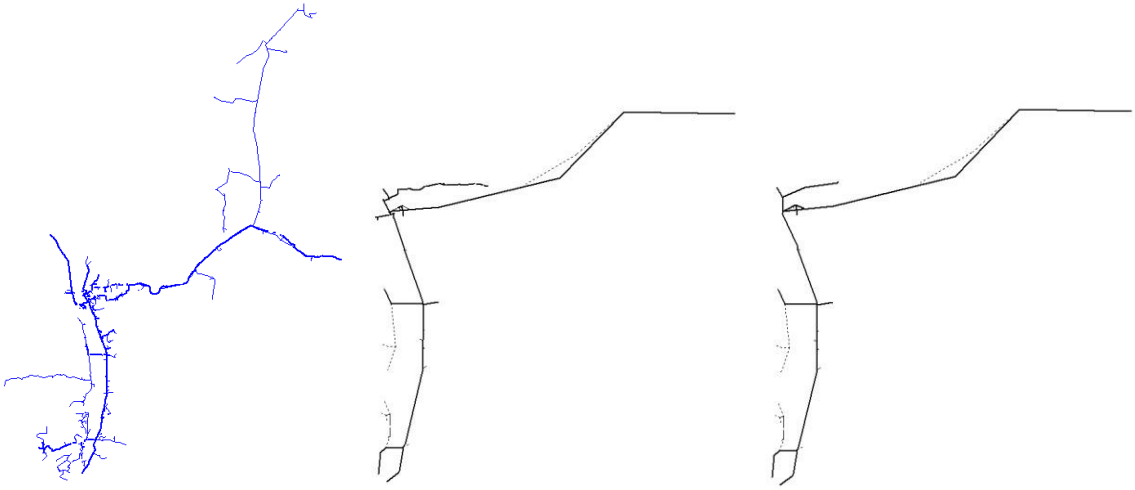


Figure 87. One-line diagram of the original circuit(left), the reduced circuit with the main line preserved (middle), and the reduced circuit without preserving the main line (right).

5.3.2.1. Limitations of the Circuit Reduction Method

The limitations of the circuit reduction methodology are the following:

1. The circuit can only contain one source node.
2. Only one section exists between any pair of nodes/buses (cannot reduce loops).

3. Buses hosting special devices (voltage regulator, transformer, shunt capacitance, etc.) cannot be reduced.
4. Only constant PQ (fixed) generations or motor loads can be aggregated. Generators and motors with varying powers should be considered as special devices and not reduced.
5. Reduction is performed sequentially (only one bus can be reduced at a time), resulting in prolonged reduction time if the original circuit contains a large number of buses; however, such reduction time is a one-time cost.

Detailed test cases highlighting the impact of circuit reduction on QSTS simulation efficiency and accuracy are presented in the validation part of the report.

5.3.3. Variable Time-Step Methods

In the final year of the project, two variable time-step (VTS) methods were implemented in the development version of CYME: the pre-determined variable time-step (PDVTS) and backtracking variable time-step (BTVTS) techniques. These methods were originally developed by other team members and presented in [75] and [124] . Minor modifications to these methods for commercial implementation were done by the CYME team.

As of September 2018, the two VTS methods were not yet directly available through CYME's Long-Term Dynamic (LTD) analysis dialog box, instead, these methods can be activated and parameterized through CYME's configuration file using the flags and variables. All other features of LTD (e.g., reports, curves, disabling delays, …) still work with the VTS methods. Results using the VTS methods are presented in the validation part of this report.

5.3.4. Temporal Parallelization Using CYMEServer

A CYMEServer environment was set up with a total of 20 individual licenses or "agents" for use as a prototype testbed for completing temporal parallelization using CYME. The approach taken was very similar to the temporal parallelization efforts completed earlier in the project using OpenDSS. First, the input files for each of the temporal divisions (or slices) needed to be separated from the large, yearlong, input data files for PV irradiance profiles and load profiles. Secondly, the many individual agents needed to be initialized and then allowed to complete the requested analysis. Finally, the data from the many agents are accumulated to determine the overall results of the yearlong analysis.

CYME uses Microsoft Access databases for input data to its power flow engine and thus an automatic workflow was developed to populate individual Access databases with a subset of the data needed for each temporal slice (up to 20 – limited by the number of concurrent licenses).

As described in earlier sections the error introduced by temporal parallelization occurs due to the discontinuity of memory-containing states of the solution. These states are namely the state of automatic voltage regulation equipment contained within the distribution feeder model (i.e. voltage regulators, LTCs, and automatic switching capacitors). To reduce the error introduced by temporal parallelization and allow higher levels of temporal parallelization, as the error introduced is directly related to the number of time divisions implemented, a Monte-Carlo method was developed to try and determine what the most likely states for the memory-containing states at the beginning of every temporal division. The implemented method, which is fully described in [92] used a relatively small number of snapshot power flow solutions to populate a distribution of potential memory states (i.e. automatic voltage regulation equipment states like tap position and on/off for capacitors). This

distribution was then used to choose the most likely initial state for that particular piece of modeled distribution equipment. Figure 88 shows example distributions for a single memory state, an LTC position in this case, for a temporal parallelization using 5 divisions or slices. For reference, the mean, median, mode, combined mode, and actual correct value (derived from a non-temporally parallelized yearlong simulation) are shown in the figure as well. As can be seen, the selection of this initial condition is often best selected using the most likely state as determined via the Monte-Carlo method developed. The method doesn't always identify the correct initial state, as shown in the data presented for parallel run number 4, but is more reliable than using other statistically derived values as the initial condition (i.e. using the mean for instance). For the CYMEServer temporal parallelization framework, the above described Monte-Carlo method of selecting the most likely initial condition was implemented and the whole system tested for multiple temporal test cases that were previously developed.

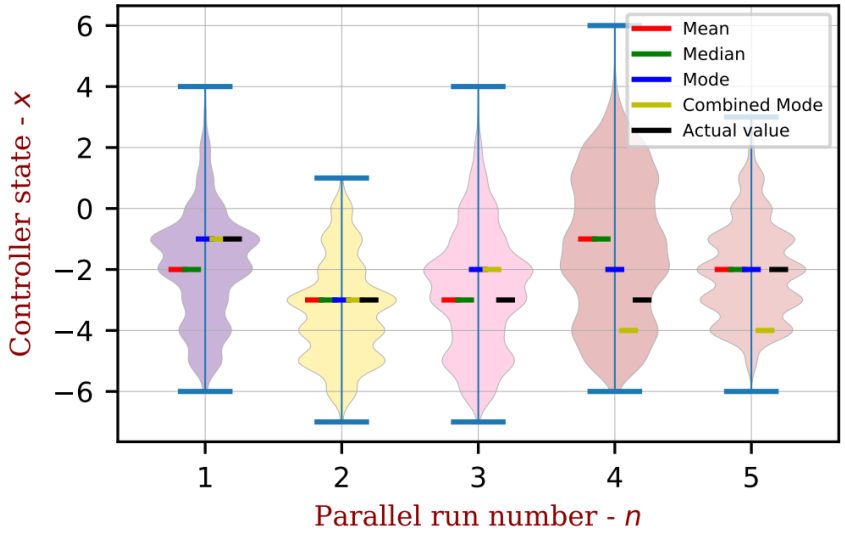


Figure 88. Example Monte-Carlo method derived most likely state of a single load tap changer (LTC) for a temporal parallelization into 5 temporal periods.

The primary goal of the CYMEServer temporal parallelization was to investigate the possibility of using the existing CYMEServer product as the parallelization engine. The work completed within this project showed that it could be used for temporal parallelization (or any other form of parallelization as well) and the setup of the framework was highly valuable to CYME itself as it was representative of how customers might approach fast QSTS implementation. Considerable amounts of time and effort were spent just getting the CYMEServer environment set up but once the license availability and application connection issues were identified and addressed with the expert help of the CYME team, the CYMEServer temporal parallelization framework operated reliably.

5.3.5. Long-Term High-Resolution Studies

- In the most recent commercial version of CYME, QSTS studies (using the LTD module) can only be run for a maximum of a single-day period, with input profiles entered manually or read from text files. The former obviously prevents running long-term QSTS simulations, whereas the latter may significantly increase the computational effort for large files due to the cost of parsing them.
- In the internal development version of CYME, it is now possible to read load and irradiance profiles from binary files, which removes the computational overhead associated with parsing

text files. This uses a plugin (a DLL) called through CYME's new Dynamic Data Pull module. Moreover, when reading profiles from binary files, the 24-hour period limit is automatically bypassed, allowing yearlong – or even longer – QSTS simulations.

- These modifications fit within an overhaul of CYME's time-series simulations currently in progress. As of September 2018, the plugin is hardcoded in CYME and can be activated through a flag in CYME's configuration file. The plugin can read any binary file stored in "single" format and requires a data point configuration file. The mapping between the profiles and the spot loads and PV systems is done through CYME's *Curves* tab of the LTD analysis dialog box (see Figure 89) and specific variables defined in the data point configuration file.

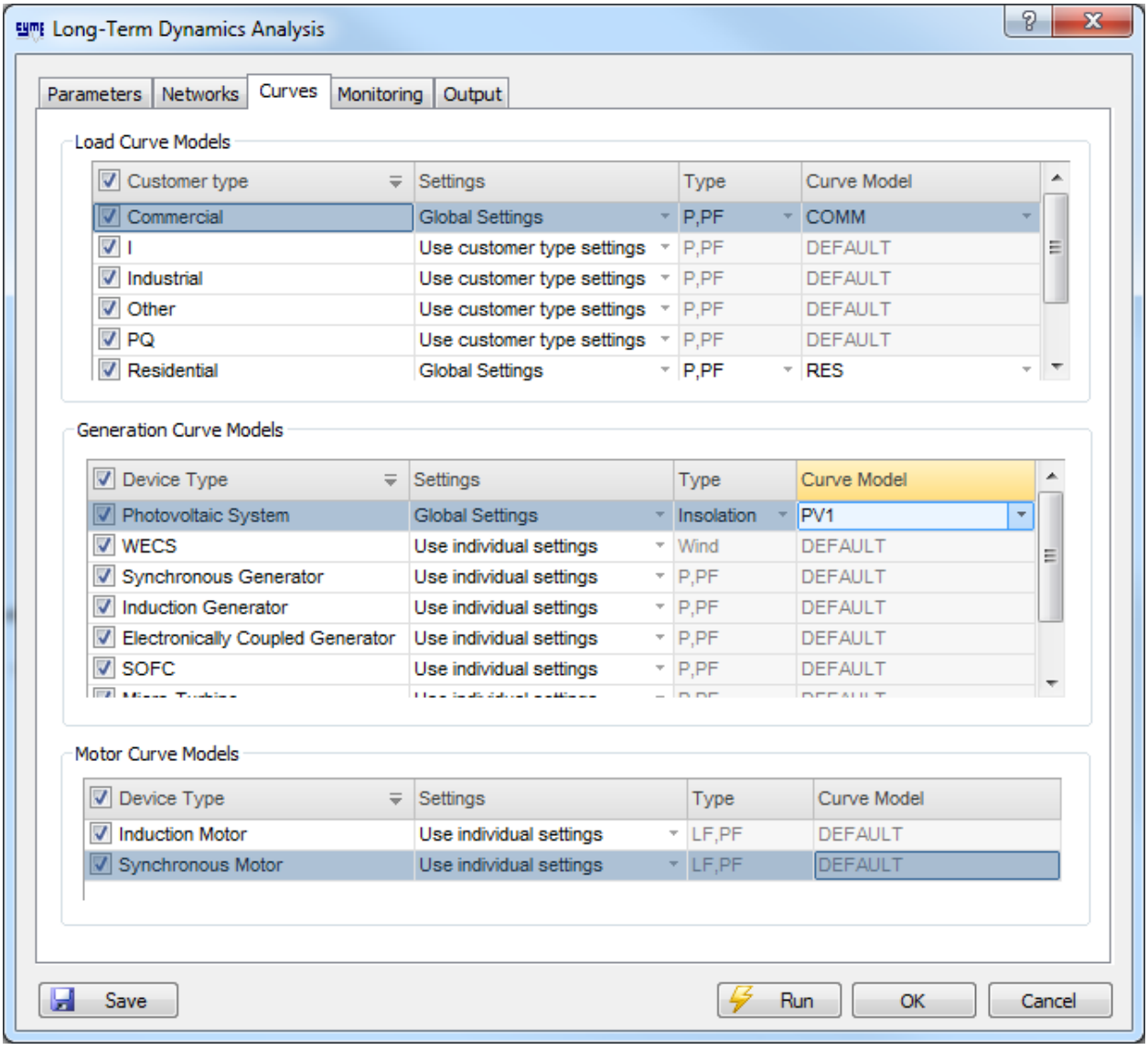


Figure 89. "Curves" tab of CYME's Long-Term Dynamic Analysis dialog box

5.3.6. *Validation- Simulation*

Several combinations of the acceleration methods were then tested on two North American circuits with PV systems for a period of one month. In the first network, the western network, the 31 days were simulated in as short as 30 seconds, with a tap change error of around 5%. This translates to a yearlong simulation in approximately 6 minutes, compared to more than 100 hours for the base case, yielding an acceleration factor of over 1000. An acceleration factor of 50 is achieved in the second network (the North Eastern Network) while maintaining a tap change error below 10%. The improvements are smaller with this network due to the large presence of distribution transformers in the model – which cannot be reduced – as well the higher number of tap changes.

Table 15. CYME Acronym table

	Description
BTVTS+PDVTS	Combination of the BTVTS and PDVTS methods. The specific parameters vary for each test case.
BTVTS	BTVTS method only. The specific parameters vary for each test case.
CR+ML	Circuit reduction preserving the main line.
CR	Circuit reduction without any specific preservation of the main line.

5.3.6.1. Western Network

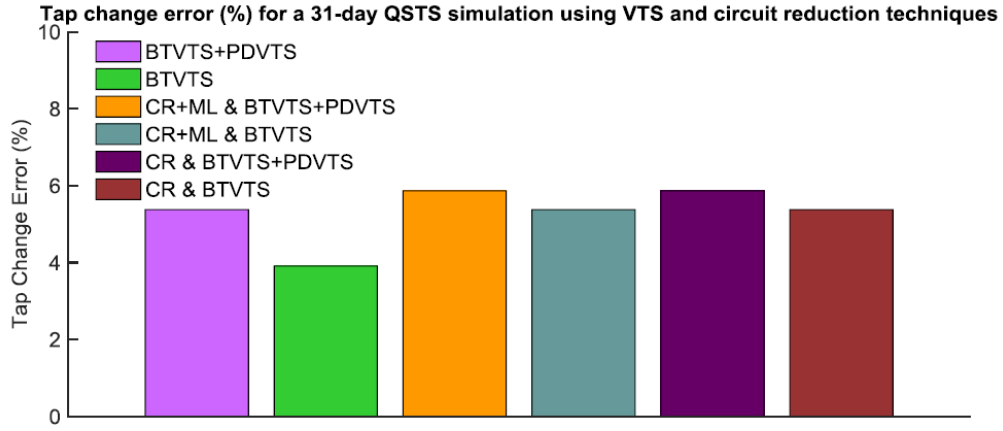


Figure 90. Tap change error for the Western network for various acceleration techniques.

Table 16. CPU time for the Western network for various acceleration techniques.

	Base Case	BTVTS+ PDVTS	BTVTS	CR+ML & BTVTS+ PDVTS	CR+ML & BTVTS	CR & BTVTS+ PDVTS	CR & BTVTS
CPU Time (min)	512.6	12.6	4.26	2.75	0.930	1.57	0.500

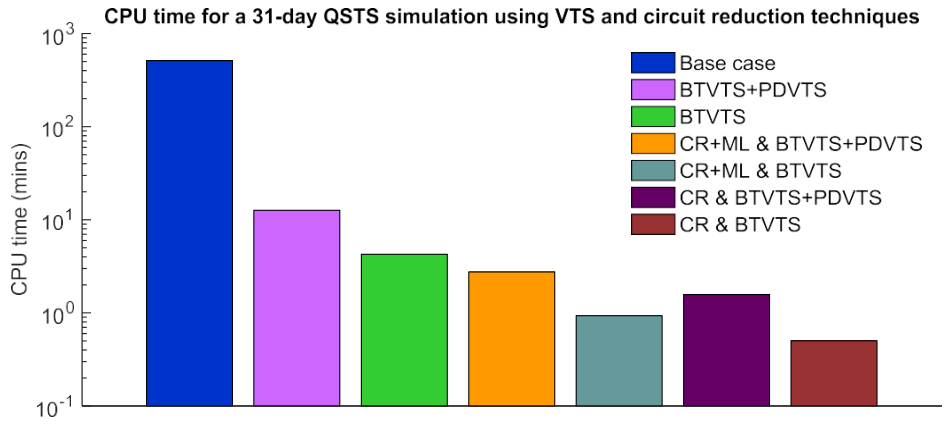


Figure 91. CPU time for the Western network for various acceleration techniques (logarithmic scale).

5.3.6.2. North Eastern Network

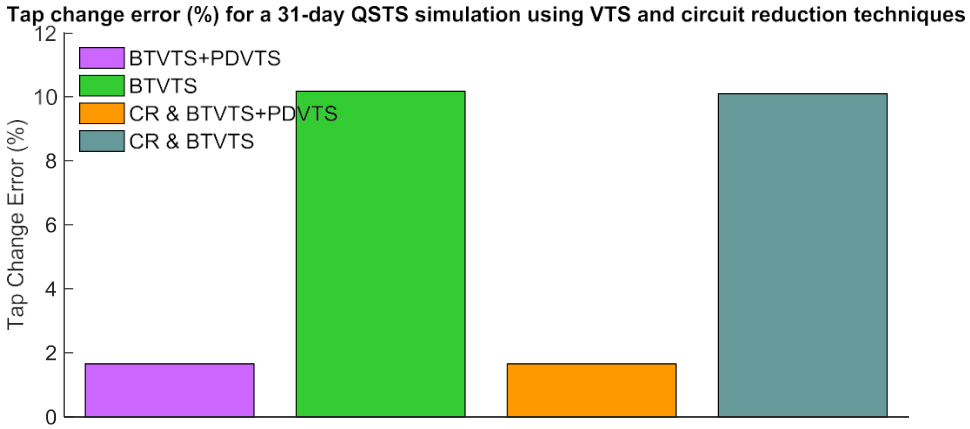


Figure 92. Tap change error for the North Eastern network for various acceleration techniques.

Table 17. CPU time for the North Eastern network for various acceleration techniques.

	Base Case	BTVTS+PDVTS	BTVTS	CR & BTVTS+PDVTS	CR & BTVTS
CPU Time (min)	417	50.3	11.5	38.4	8.71

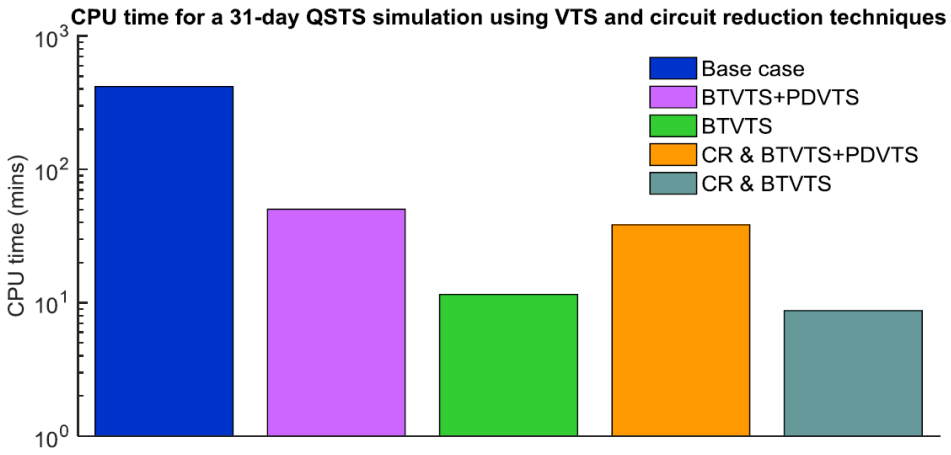


Figure 93. CPU time for the Northeastern network for various acceleration techniques (logarithmic scale).

5.3.7. Yearlong QSTS

In this section, we validate accelerated yearlong QSTS simulations in CYME. To have a more manageable base case, we consider a modified version of the IEEE 34 Node Test Feeder [125] with a 300 kW PV system added on node 840. Its one-line diagram in CYME is reproduced in Figure 94. The acronyms and metrics defined at the beginning of Section 5 are also used for this validation case.

We consider two profiles: one for loads with a 1-minute time interval, and another one for solar irradiance with a 1-second interval. We also only do a "full" circuit reduction, i.e., we do not reduce the network while explicitly preserving the main line. This is because the main line comprises the majority of the original network. The one-line diagram of the reduced version of the feeder is shown in Figure 95.

To better highlight the compromise between accuracy and efficiency, we run the BTVTS method with two different values of Δt_{max}: 300 sec and 120 sec. These are referred to as BTVTS(1) and BTVTS(2), respectively. The BTVTS+PDVTS method is run with ΔP_{max} = 1 kW.

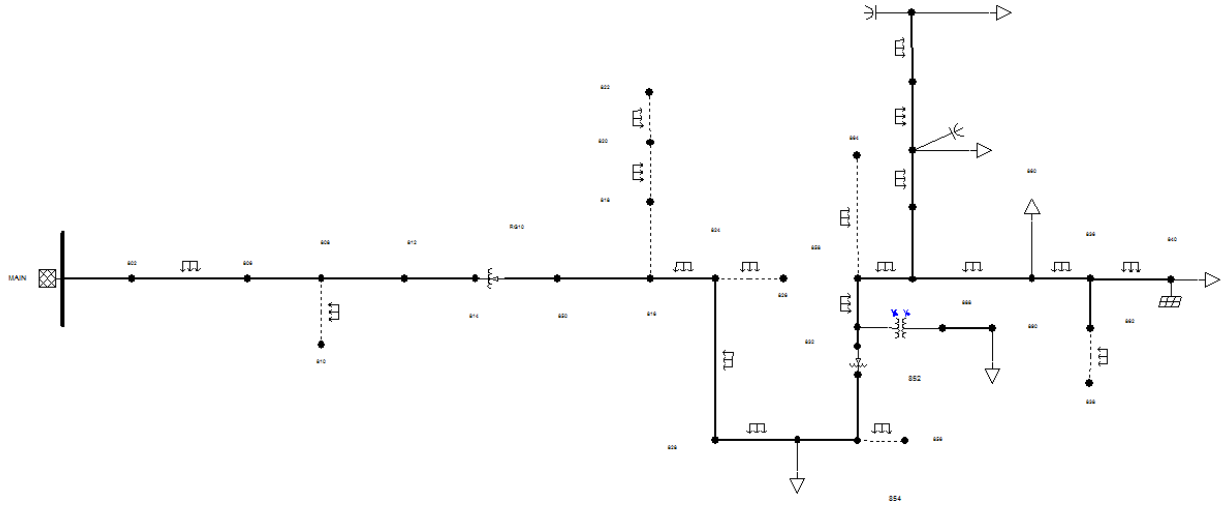


Figure 94. One-line diagram of the modified IEEE 34 Node Test Feeder in CYME

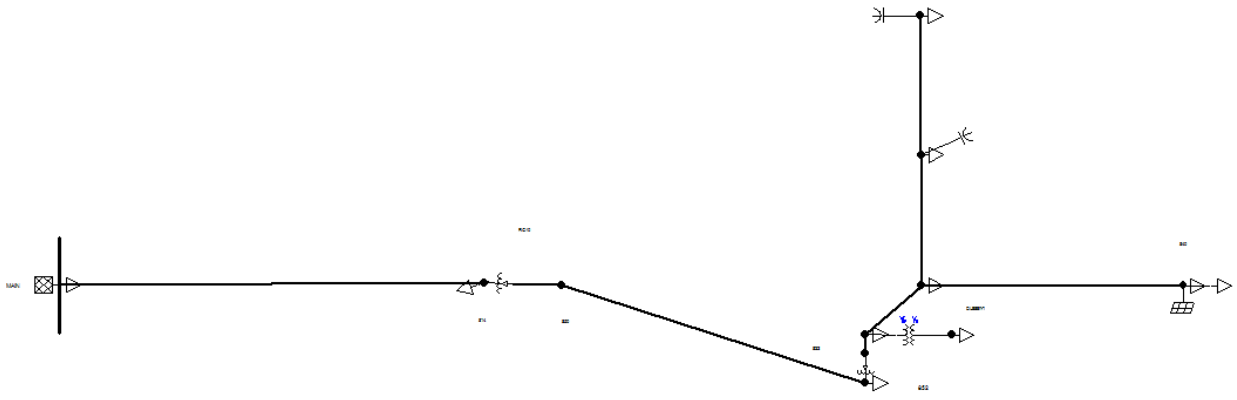


Figure 95. One line-diagram of the reduced modified IEEE 34 Node Test Feeder in CYME

The number of regulator tap changes for the modified IEEE 34 Node Test Feeder with different combinations of acceleration techniques is presented in Table 18 and Table 19. The error count is

larger than the simulations in the previous section described above, but this is due to running the simulation for a longer period. To get a better comparison, the overall tap change error computed using the modified mean absolute error is plotted in Figure 96. The error varies significantly as a function of the acceleration techniques but remains below 10% for all but CR & BTVTS(1). In that specific case, the error is only slightly above 10%.

Table 18. Tap position accuracy for the IEEE 34 Node Test Feeder for various acceleration techniques (1 of 2).

	Base Case	BTVTS+PDVTS		BTVTS(1)		BTVTS(2)	
Device	*Nb. Tap Changes*	*Nb. Tap Changes*	*Error*	*Nb. Tap Changes*	*Error*	*Nb. Tap Changes*	*Error*
Reg. 1-A	5316	5024	292	5044	272	5256	60
Reg. 1-B	2832	2690	142	2736	96	2796	36
Reg. 1-C	3073	2961	112	2937	136	3041	32
Reg. 2-A	7426	6993	433	6556	870	7264	162
Reg. 2-B	5933	5661	272	5321	612	5795	138
Reg. 2-C	6399	6095	304	5697	702	6191	208

Table 19. Tap position accuracy for the IEEE 34 Node Test Feeder for various acceleration techniques (2 of 2).

	Base Case	CR & BTVTS+PDVTS		CR & BTVTS(1)		CR & BTVTS(2)	
Device	*Nb. Tap Changes*	*Nb. Tap Changes*	*Error*	*Nb. Tap Changes*	*Error*	*Nb. Tap Changes*	*Error*
Reg. 1-A	5316	5350	34	5300	16	5544	228
Reg. 1-B	2832	3012	180	3066	234	3142	310
Reg. 1-C	3073	3028	45	3032	41	3146	73
Reg. 2-A	7426	9252	1826	8816	1390	9746	2320
Reg. 2-B	5933	5548	385	5280	653	5636	297
Reg. 2-C	6399	6292	107	5863	536	6421	22

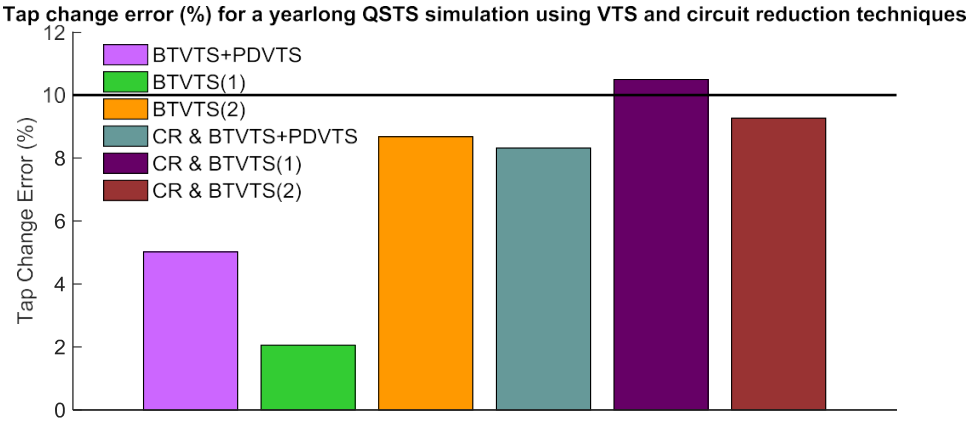


Figure 96. Tap position error for the IEEE 34 Node Test Feeder for various acceleration techniques.

Table 20. CPU time for the IEEE 34 Node Test Feeder for various acceleration techniques.

	Base Case	BTVTS+ PDVTS	BTVTS(1)	BTVTS(2)	CR & BTVTS+ PDVTS	CR & BTVTS(1)	CR & BTVTS(2)
CPU Time (min)	145.5	14.9	9.47	6.23	12.4	7.08	4.91

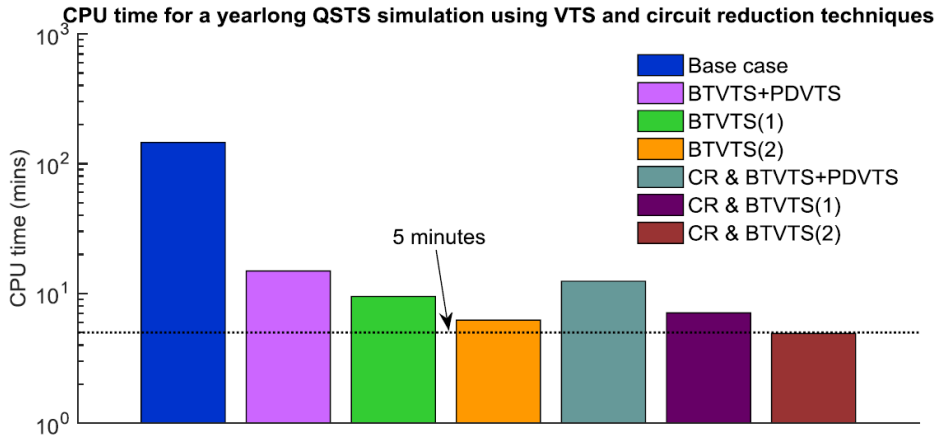


Figure 97. CPU time for the IEEE 34 Node Test Feeder for various acceleration techniques (logarithmic scale).

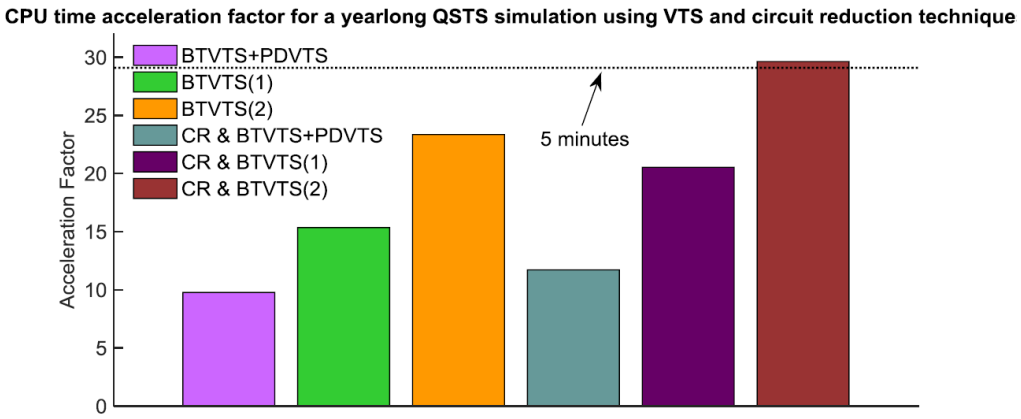


Figure 98. CPU time acceleration factor for the IEEE 34 Node Test Feeder for various acceleration techniques.

In conclusion, the CYME acceleration methods were compared on a validation test case network, a modified version of the IEEE 34 Node Test Feeder. In this case, the simulation was run for an entire year. By combining circuit reduction and BTVTS with a fairly large maximum step size (5 minutes), we were able to run a yearlong QSTS simulation in **4.91 minutes** with a tap change error of **9.26%**, thereby achieving the objectives set at the beginning of the project.

6. CONCLUSIONS

Understanding the impact of distributed solar photovoltaic (PV) resources on various elements of the distribution feeder is imperative for their cost-effective integration. High-resolution quasi-static time series (QSTS) simulations are an effective tool to identify and mitigate these impacts. However, the significant computational burden associated with running such simulations and the lack of measured PV power inputs for such simulations are major challenges to their widespread adoption. The work presented in this project report develops rapid QSTS algorithms that drastically reduce the computational time of quasi-static time-series simulation and presents methods for simulating PV power inputs from readily-available data. The rapid QSTS algorithms are thus capable of accurately modeling solar PV systems as well as the operation of voltage-regulating equipment (e.g. number of tap actions).

The rapid QSTS algorithms are evaluated for scalability, accuracy, and robustness on both standard IEEE test circuits and real utility-scale distribution feeders, with low voltage secondary side modeled. Table 21 shows how the new rapid QSTS algorithms can accurately capture all potential PV impacts on a distribution system while reducing the computational time significantly. Several algorithms have also been implemented in MATLAB as open-source functions, in updated versions of OpenDSS, and CYME software.

Table 21. Comparison of the new rapid QSTS algorithms to other analysis methods.

	Extreme Voltages	Thermal Loading	Regulators Tap Changes	Capacitor Switching	Time outside ANSI	Losses	Computation Time[1]
Snapshot	Good	Good	-	-	-	-	<1 sec
Hourly Timeseries	Great	Great	-	-	Good	Great	5 sec
1 day QSTS	Poor	Poor	Decent	Decent	Poor	Poor	5 minutes
1 year QSTS	Great	Great	Great	Great	Great	Great	36 hours
New Rapid QSTS Algorithms	Great	Great	Great	Great	Great	Great	30 sec

An example of how to get these several orders of magnitude in speed improvements is shown in Figure 99. The original simulation time is demonstrated with the length of many red wait bars at the top of the figure. This computational time is 127760 seconds, or 35.5 hours! In order to decrease the computational time, circuit reduction is performed on the circuit, which takes roughly 19 seconds to process and create the new reduced-order circuit. The QSTS simulation is run with the variable time-step algorithm that must process through the time-series to establish the time points of interest, which takes roughly 11 seconds of processing. Finally, the QSTS simulation is run with variable time-step on the reduced circuit, which takes 49 seconds. The total time is 1.3 minutes instead of 35.5 hours. This speed improvement demonstrates the QSTS simulation running roughly 1600 times faster. The magnitude of this speed improvement is shown in Figure 100 where that is the equivalent of compressing the entire Eiffel Tower into the size of a volleyball. It is also roughly the number of drops of water in 21 gallons. While the exact numbers of speed improvement varying slightly from circuit to circuit, the rapid QSTS algorithms can significantly decrease the amount of computational effort and time while maintaining a high level of accuracy.

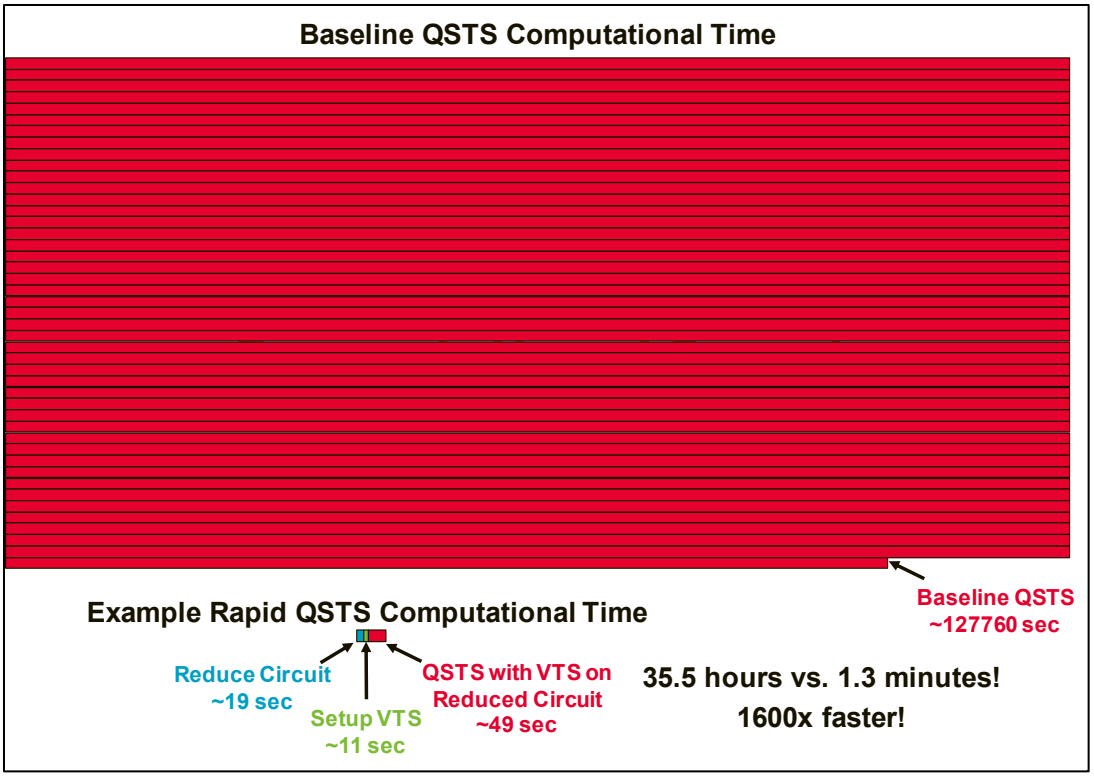


Figure 99. Example of reducing a very long simulation to a couple of minutes using circuit reduction and variable time-step (VTS).

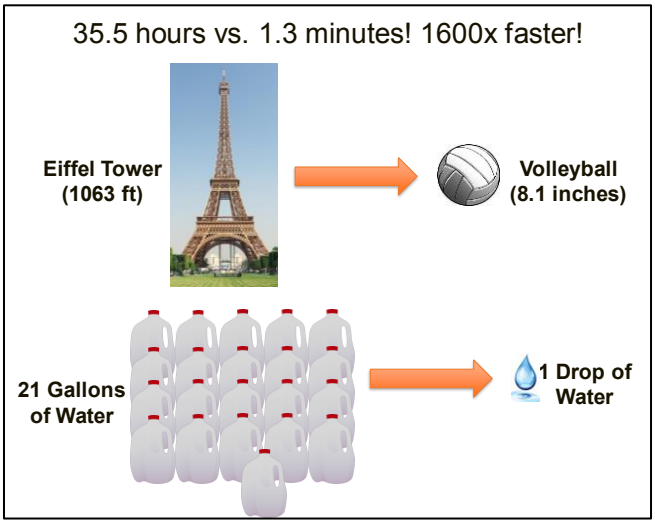


Figure 100. Examples of dramatic reduction in computational time achieved from rapid QSTS algorithms reducing a 35.5-hour simulation to 1.3 minutes.

PV modeling methods were presented for (a) creating a simulated high-frequency irradiance time-series based solely on low-frequency, readily available satellite-derived irradiance, and (b) producing high-frequency PV inputs at all simulated PV locations on a distribution feeder based on high-frequency simulated or measured irradiance time-series. These methods were generally found to match the solar variability of ground measurements, specifically to lead to an accurate number of tap

change operations when integrated into QSTS simulations, though additional refinement of the algorithms is required to increase the accuracy.

In the coming years, QSTS simulations will become an integral part of the distribution system analysis for solar PV impact studies. The significant computational time reduction and the presented methods to simulated PV power profiles mean that electric utilities can more efficiently and accurately utilize QSTS simulation to evaluate PV interconnection requests. In addition, the ability to perform fast and accurate QSTS simulations has widespread applications to demand response control, conservation voltage reduction, locational hosting capacity, and feeder reconfiguration strategies.

6.1. Final Deliverables:

The Final Deliverables for the project were the following:

- New and innovative methods were developed to improve the accuracy and speed of QSTS and implemented into commercial and open-source software.
- Temporal parallelization implemented in OpenDSS using open-source tools and on CYMServer as detailed in a report.
- Develop an IEEE QSTS distribution test feeder.
- Implemented accelerated QSTS analysis into CYME Long-Term Dynamics commercial distribution system analysis software package and OpenDSS and demonstrate solution times of 5 minutes or less for year-long simulations using at least three complex feeder types.
- Public release of user interfaces for the HRIA and load data models.

6.1.1. Temporal parallelization implemented in OpenDSS using open-source tools and on CYMServer

Temporal parallelization was implemented in OpenDSS for all versions since v8.0. The user manual has been published online with the software "User Instructions for Parallel Processing". On online video tutorial "Introduction to OpenDSS Parallel Machine - Parallel Processing with OpenDSS" is also available at http://resourcecenter.ieee-pes.org/pes/product/education/PESVIDWEBGPS0008.

Temporal parallelization in CYME was developed and demonstrated. Research tools for temporal parallelization were developed in CYME Server for 20 instances/licenses, and improvements were made to the parallelization structure.

The parallelization publications from the project include:

- D. Montenegro, R. C. Dugan, and M. J. Reno, "Open Source Tools for High Performance Quasi-Static-Time-Series Simulation Using Parallel Processing," IEEE Photovoltaic Specialists Conference (PVSC), 2017.
- R. Hunsberger and B. Mather, "Temporal Decomposition of Distribution System Quasi-Static Time-Series Simulation," IEEE PES General Meeting, 2017.
- D. Montenegro and R. C. Dugan, "OpenDSS and OpenDSS-PM open source libraries for NI LabVIEW," *2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, 2017.
- A. Latif, B. Mather, "Monte Carlo Based Method for Parallelizing Quasi-Static Time-Series Simulations," IEEE Int. Conf. Prob. Methods Appl. Pwr. Sys., 2018.

- J. A. Azzolini, M. J. Reno, D. Montenegro, "Implementation of Temporal Parallelization for Rapid Quasi-Static Time-Series (QSTS) Simulations," IEEE Photovoltaic Specialists Conference (PVSC), 2019.
- A. Latif, B. Mather, "Monte Carlo Based QSTS Parallelization Tool for the CYME Server Application," [Submitted], 2019.

6.1.2. Develop an IEEE QSTS distribution test feeder.

Five test circuits were developed for the project to test the QSTS algorithms. Each circuit contains a yearlong simulation at 1-second resolution with high penetrations of PV. The five QSTS test circuits are shown in Figure 101.

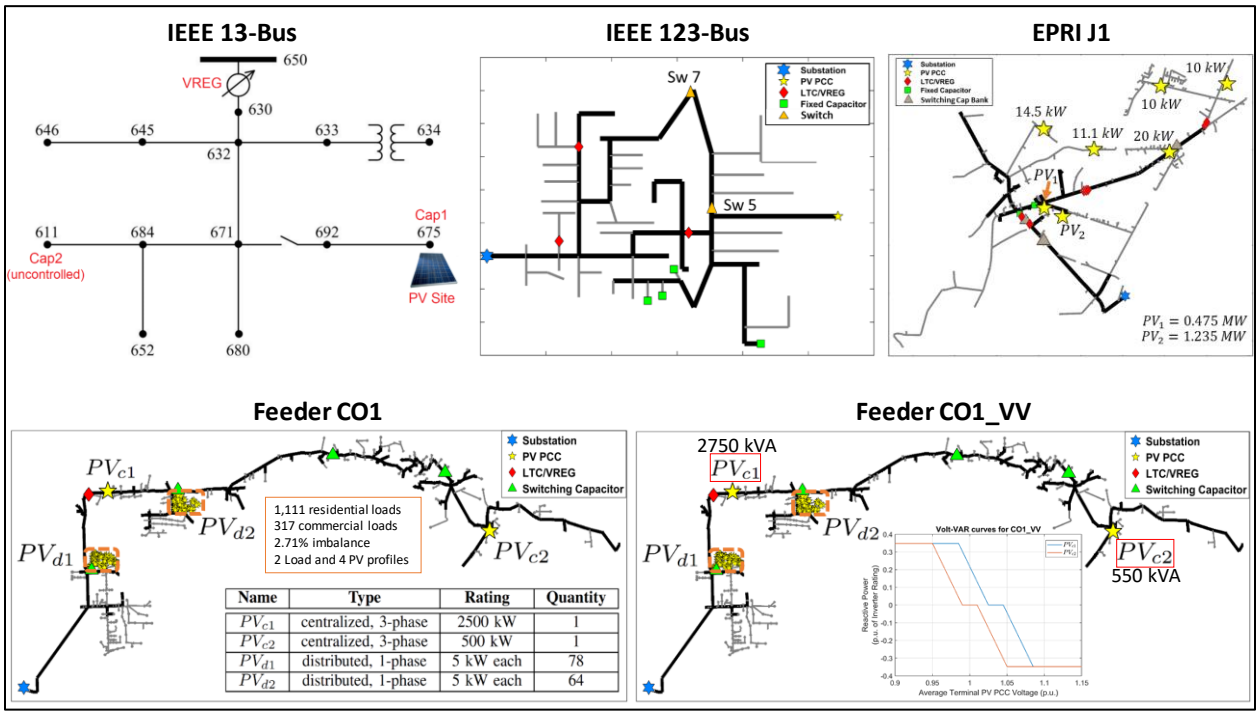


Figure 101. Five QSTS Test Circuits

6.1.3. Implement accelerated QSTS analysis into CYME Long-Term Dynamics commercial distribution system analysis software package and OpenDSS and demonstrate solution times of 5 minutes or less.

The CYME Long-Term Dynamics (LTD) solver is now able to perform yearlong 1-second resolution simulations. CYME demonstrated speed improvements in power flow solver (roughly 10x faster), circuit reduction (CR), predetermined variable time-step (PDVTS), and backtrack variable time-step (BTVTS).

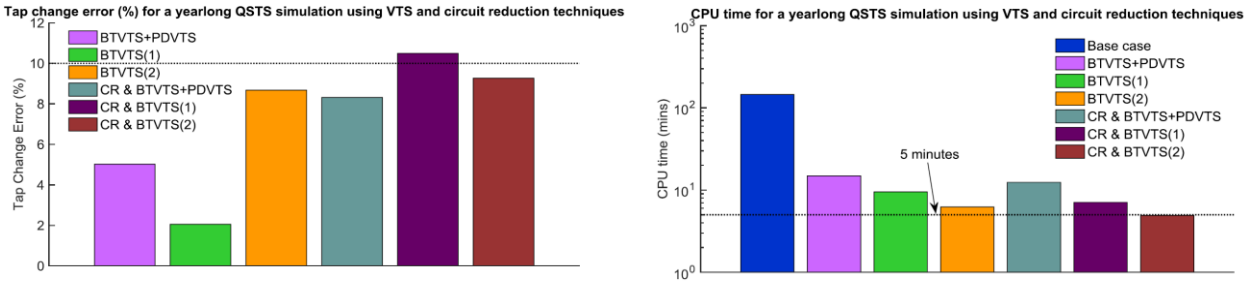


Figure 102. CYME QSTS simulations showing percent error and speed improvement.

In OpenDSS, combined with algorithms implemented in MATLAB, a yearlong high-resolution simulation with <10% tap change error was solved in less than 5 minutes.

6.1.4. Public release of user interfaces for the HRIA and load data models.

The user interface shown in Figure 103 was developed for the load variability and load diversity models in MATLAB for public release on the PVPMC website. Variability and diversity example data (training) is also provided. The algorithms can generate sets of diversity-related load data (size depends on the depth of input data). A user manual for the algorithm and GUI is available.

The most recent HRIA down-scaling methods code (in R) was developed and will be included in PVPMC. This algorithm ingests the latest data format of NREL's NSRDB and uses the improved 30-minute data segment interpolation. A wrapper for running the HRIA can also be developed for MATLAB to make all tools "runnable" from the MATLAB framework

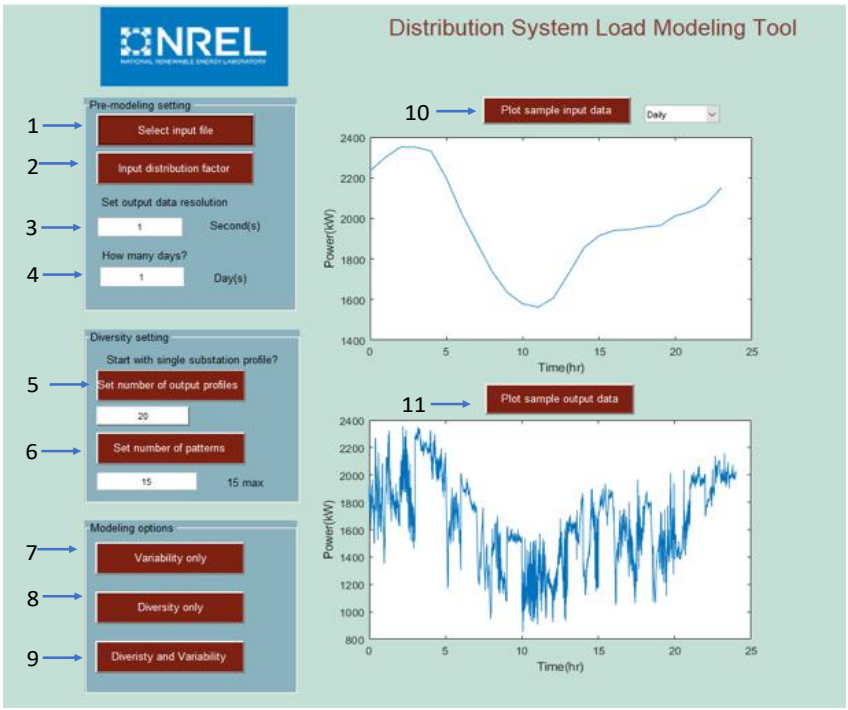


Figure 103. GUI for load variability and diversity modeling.

6.2. Inventions, Patents, Publications, and Other Results

The journal papers published as part of the project were:

- Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl, "Multiphase Distribution Feeder Reduction," IEEE Transactions on Power Systems, 2017. [96]
- D. Montenegro, G. A. Ramos and S. Bacha, "A-Diakoptics for the Multicore Sequential-Time Simulation of Microgrids Within Large Distribution Systems," in *IEEE Transactions on Smart Grid*, vol. 8, 2017. [89]
- J. Deboever, S. Grijalva, M. J. Reno, and R. J. Broderick, "Fast Quasi-Static Time-Series (QSTS) for Yearlong PV Impact Studies using Vector Quantization," Solar Energy, 2018. [78]
- D. Montenegro, G. A. Ramos and S. Bacha, "An Iterative Method for Detecting and Localizing Islands Within Sparse Matrixes Using DSSim-RT," in *IEEE Transactions on Industry Applications*, vol. 54, 2018. [90]
- M. U. Qureshi, S. Grijalva, M. J. Reno, J. Deboever, X. Zhang, and R J. Broderick, "A Fast Scalable Quasi-Static Time Series Analysis Method for PV Impact Studies using Linear Sensitivity Model," IEEE Transactions on Sustainable Energy, 2019. [84]
- W. Zhang, W. Kleiber, A. R. Florita, B-M. Hodge, and B. Mather, "A Stochastic Downscaling Approach for Generating High-Frequency Solar Irradiance Scenarios," Solar Energy vol. 176, pp. 370-379, 2018. [110]
- Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl, "Inversion Reduction Method for Real and Complex Distribution Feeder Models," IEEE Transactions on Power Systems, 2019.[97]
- X. Zhu and B. Mather, "Data-Driven Distribution System Load Modeling for Quasi-Static Time-Series Simulation," *IEEE Trans. on Smart Grids,* 2020. [100]
- Z. K. Pecenak, H. V. Haghi, C. Li, M. J. Reno, V. R. Disfani, J. Kleiss "Aggregation of Voltage-Controlled Devices During Distribution Network Reduction", IEEE Transactions on Smart Grid, 2020. [126]

The conference papers and technical reports published as part of the project were:

- M. Lave and A. Weekley, "Comparison of High-Frequency Solar Irradiance: Ground Measured vs. Satellite-Derived", IEEE Photovoltaic Specialists Conference (PVSC), 2016. [127]
- M. Lave, J. E. Quiroz, M. J. Reno, and R. J. Broderick, "High Temporal Resolution Load Variability Compared to PV Variability", IEEE Photovoltaic Specialists Conference (PVSC), 2016. [65]
- M. J. Reno and R. J. Broderick, "Predetermined Time-Step Solver for Rapid Quasi-Static Time Series (QSTS) of Distribution Systems", IEEE Innovative Smart Grid Technologies (ISGT), 2017. [98]
- J. Deboever, X. Zhang, M. J. Reno, R. J. Broderick, S. Grijalva, and F. Therrien "Challenges in reducing the computational time of QSTS simulations for distribution system analysis," Sandia National Laboratories, SAND2017-5743, 2017. [71]
- D. Montenegro, J. Gonzalez and R. Dugan, "Multi-rate control mode for maintaining fidelity in Quasi-Static-Time-Simulations," *2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, 2017. [117]
- D. Montenegro and R. C. Dugan, "OpenDSS and OpenDSS-PM open source libraries for NI LabVIEW," *2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, 2017. [128]

- X. Zhang, S. Grijalva, M. J. Reno, J. Deboever, and R. J. Broderick, "A Fast Quasi-Static Time Series (QSTS) Simulation Method for PV Impact Studies Using Voltage Sensitivities of Controllable Elements," IEEE Photovoltaic Specialists Conference (PVSC), 2017. [82]
- F. Therrien, M. Belletête, J. Lacroix, and M. J. Reno, "Algorithmic Aspects of a Commercial-Grade Distribution System Load Flow Engine", IEEE Photovoltaic Specialists Conference (PVSC), 2017. [122]
- J. Galtieri, M. J. Reno, "Intelligent Sampling of Periods for Reduced Computational Time of Time Series Analysis of PV Impacts on the Distribution System," IEEE Photovoltaic Specialists Conference (PVSC), 2017. [67]
- M. Lave, M. J. Reno, R. J. Broderick, "Creation and Value of Synthetic High-Frequency Solar Simulations for Distribution System QSTS Simulations", IEEE Photovoltaic Specialists Conference (PVSC), 2017. [112]
- M. J. Reno, R. J. Broderick, and L. Blakely, "Machine Learning for Rapid QSTS Simulations using Neural Networks," IEEE Photovoltaic Specialists Conference (PVSC), 2017. [68]
- J. Deboever, S. Grijalva, M. J. Reno, X. Zhang, and R. J. Broderick, "Scalability of the Vector Quantization Approach for Fast QSTS Simulation for PV Impact Studies," IEEE Photovoltaic Specialists Conference (PVSC), 2017. [79]
- D. Montenegro, R. C. Dugan, and M. J. Reno, "Open Source Tools for High Performance Quasi-Static-Time-Series Simulation Using Parallel Processing," IEEE Photovoltaic Specialists Conference (PVSC), 2017. [93]
- B. Mather, "Fast Determination of Distribution-Connected PV Impacts Using a Variable Time-Step Quasi-Static Time-Series Approach," IEEE Photovoltaic Specialists Conference (PVSC), 2017. [76]
- M. J. Reno, J. Deboever, and B. Mather, "Motivation and Requirements for Quasi-Static Time Series (QSTS) for Distribution System Analysis", IEEE PES General Meeting, 2017. [53]
- R. Hunsberger and B. Mather, "Temporal Decomposition of Distribution System Quasi-Static Time-Series Simulation," IEEE PES General Meeting, 2017. [91]
- M. Chamana, B. Mather, "Variability Extraction and Synthesis via Multi-Resolution Analysis Using Distribution Transformer High-Speed Power Data," IEEE Intelligent System Applications to Power Systems, 2017. [101]
- L. Blakely, M. J. Reno, R. J. Broderick, "Decision Tree Ensemble Machine Learning for Rapid QSTS Simulations," IEEE Innovative Smart Grid Technologies (ISGT), 2018. [69]
- W. Li, B. Mather, J. Deboever, M. J. Reno, "Fast QSTS for Distributed PV Impact Studies using Vector Quantization and Variable Time-Steps," IEEE Innovative Smart Grid Technologies (ISGT), 2018. [81]
- A. Latif, B. Mather, "Monte Carlo Based Method for Parallelizing Quasi-Static Time-Series Simulations," IEEE Int. Conf. Prob. Methods Appl. Pwr. Sys., 2018. [92]
- X. Zhu, B. Mather, "DWT-Based Aggregated Load Modeling and Evaluation for Quasi-Static Time-Series Simulation on Distribution Feeders," IEEE PES General Meeting, 2018. [99]
- M. U. Qureshi, S. Grijalva, and M. J. Reno, "A Fast Quasi-Static Time Series Simulation Method for PV Smart Inverters with Var Control using Linear Sensitivity Model," IEEE Photovoltaic Specialists Conference (PVSC), 2018. [84]
- M. Lave, M. J. Reno, and R. J. Broderick "Implementation of Synthetic Cloud Fields for PV Modeling in Distribution Grid Simulations," IEEE Photovoltaic Specialists Conference (PVSC), 2018. [129]

- J. Deboever, S. Grijalva, M. J. Reno, and R. J. Broderick, "Algorithms to Effectively Quantize Scenarios for PV Impact Analysis using QSTS Simulation," IEEE Photovoltaic Specialists Conference (PVSC), 2018. [80]
- M. J. Reno, J. A. Azzolini, and B. Mather, "Variable Time-Step Implementation for Rapid Quasi-Static Time-Series (QSTS) Simulations of Distributed PV," IEEE Photovoltaic Specialists Conference (PVSC), 2018. [77]
- W. Zhang, W. Kleiber, A. R. Florita, B.-M. Hodge, and B. Mather, "Modeling and Simulation of High-Frequency Solar Irradiance," IEEE Journal of Photovoltaics, vol. v. 9, pp. 124-131, 2018. [109]
- J. A. Azzolini, M. J. Reno, and D. Montenegro, "Implementation of Temporal Parallelization for Rapid Quasi-Static Time-Series (QSTS) Simulations," IEEE Photovoltaic Specialists Conference (PVSC), 2019. [94]
- J. A. Azzolini, M. J. Reno, and M. Lave, "Visualization Methods for Quasi-Static Time-Series (QSTS) Simulations with High PV Penetration," IEEE Photovoltaic Specialists Conference (PVSC), 2019. [116]
- M. U. Qureshi, S. Grijalva, and M. J. Reno, "A Rapid Quasi-Static Time Series Method for Evaluating Current-Related Distributed PV Impacts including Feeder Loading and Line Losses," IEEE PES General Meeting, 2019. [85]

6.3. Path Forward

Although the rapid QSTS algorithms discussed provide attractive computational speed, there are several research opportunities to further improve the performance of the algorithms. Namely, combining different fast time-series algorithms can provide additional computational time reduction as discussed in Section 3.5. This will especially become more important with larger circuits for distribution systems over larger areas or combined transmission and distribution systems. All test circuits analyzed were single feeders, so the computational effort will increase when simulating an entire substation or larger areas. The computational requirements for QSTS are also expected to continue increasing in the future with more detailed distribution system models, more complex controls like demand response, and higher penetrations of distributed energy resources.

Extensive work was done on improving the speed of QSTS and demonstrating a practical and robust distribution system simulation tool. By making quasi-static time-series simulations practical, there are vast research opportunities for the application of this simulation tool. The applications discussed focused on the impact analysis of new solar PV interconnections, but other distributed energy resources could also be studied. For instance, wind turbines could be connected to medium-voltage distribution feeders and their impact should be studied prior to their connection. Furthermore, this fast time-series algorithm could also be used to determine appropriate settings in different controllable devices on distribution feeders.

The results from the fast QSTS simulations could also be used to improve the hosting capacity (HC) estimates. Existing methods used by electric utilities to estimate the HC involves fixing the states of VR devices to their nominal positions (or in some cases extreme positions). Without any regulation, the amount of solar PV that can be installed at a particular node in a distribution feeder is usually limited by the overvoltage constraint. As a result, the existing HC estimates are extremely conservative and unnecessarily limit the PV integration. On the other hand, rapid QSTS algorithms accurately track the states of VR devices through time, and the impact of the number of tap changes can also be included into the hosting capacity analysis. This can lead to more realistic HC estimates that are specific to the feeder loading conditions.

Lastly, the algorithm developed in this work can be used to determine the optimal smart inverter settings. Previous research done in this domain uses a very coarse resolution brute-force QSTS, mainly due to the computational time constraints [130]. However, the fast QSTS algorithm overcomes this limitation and allows for high fidelity time series simulations to be used for such an analysis.

REFERENCES

[1] (2009). *High Penetration Solar Deployment Funding Opportunity*. Available: https://www.energy.gov/eere/solar/high-penetration-solar-deployment-funding-opportunity

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

[3] M. J. Reno, K. Coogan, and R. J. Broderick, "Impact of PV Variability and Ramping Events on Distribution Voltage Regulation Equipment," in *IEEE Photovoltaic Specialists Conference (PVSC)*, 2014.

[4] J. E. Quiroz, M. J. Reno, and R. J. Broderick, "Time Series Simulation of Voltage Regulation Device Control Modes," in *IEEE Photovoltaic Specialists Conference*, 2013.

[5] B. A. Mather, "Quasi-static time-series test feeder for PV integration analysis on distribution systems," in *IEEE Power and Energy Society General Meeting*, 2012.

[6] D. Paradis, F. Katiraei, and B. Mather, "Comparative analysis of time-series studies and transient simulations for impact assessment of PV integration on reduced IEEE 8500 node feeder," in *IEEE Power and Energy Society General Meeting*, 2013, pp. 1-5.

[7] M. J. Reno, K. Coogan, R. J. Broderick, and S. Grijalva, "Reduction of Distribution Feeders for Simplified PV Impact Studies," in *IEEE Photovoltaic Specialists Conference*, 2013.

[8] B. Mather, "Analysis of high-penetration levels of PV into the distribution grid in California," 2011.

[9] B. Mather, B. Kroposki, R. Neal, F. Katiraei, A. Yazdani, J. Aguero, T. Hoff, B. Norris, A. Parkins, and R. Seguin, "Southern California Edison High-Penetration Photovoltaic Project–Year 1 Report," NREL Technical Report TP5500-508752011.

[10] J. W. Smith, R. Dugan, and W. Sunderman, "Distribution modeling and analysis of high penetration PV," in *IEEE Power and Energy Society General Meeting*, 2011, pp. 1-7.

[11] C. D. López, "Shortening time-series power flow simulations for cost-benefit analysis of LV network operation with PV feed-in," vol. http://www.diva-portal.org/smash/get/diva2:782313/FULLTEXT01.pdf., January 2015.

[12] M. J. Reno, A. Ellis, J. Quiroz, and S. Grijalva, "Modeling Distribution System Impacts of Solar Variability and Interconnection Location," in *World Renewable Energy Forum*, 2012: American Solar Energy Society.

[13] J. Quiroz and M. J. Reno, "Detailed Grid Integration Analysis of Distributed PV," in *IEEE Photovoltaic Specialists Conference*, 2012.

[14] M. J. E. Alam, K. M. Muttaqi, and D. Sutanto, "An Approach for Online Assessment of Rooftop Solar PV Impacts on Low-Voltage Distribution Networks," *IEEE Transactions on Sustainable Energy,* vol. 5, no. 2, pp. 663-672, 2014.

[15] M. Schmidt, A. Panosyan, E. Barthlein, K. O'Brien, and O. Mayer, "Method of generating sets of PV plant power time series for grid simulation purposes," in *3rd IEEE PES International Conference and Exhibition on Innovative Smart Grid Technologies (ISGT Europe)*, 2012, pp. 1-7.

[16] M. Baggu, R. Ayyanar, and D. Narang, "Feeder model validation and simulation for high-penetration photovoltaic deployment in the Arizona Public Service system," in *IEEE Photovoltaic Specialist Conference (PVSC)*, 2014.

[17] 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.
[18] *IEEE P1547.7 D110 - Draft Guide to Conducting Distribution Impact Studies for Distributed Resource Interconnection*, 2013.
[19] J. Seuss, M. J. Reno, R. J. Broderick, and S. Grijalva, "Maximum PV Size Limited by the Impact to Distribution Protection," in *IEEE Photovoltaic Specialists Conference*, 2015.
[20] C. B. Jones, M. Lave, M. J. Reno, R. Darbali-Zamora, A. Summers, and S. Hossain-McKenzie, "Volt-Var Curve Reactive Power Control Requirements and Risks for Feeders with Distributed Roof-Top Photovoltaic Systems," *Energies,* 2020.
[21] M. A. Cohen and D. S. Callaway, "Effects of distributed PV generation on California's distribution system, Part 1: Engineering simulations," *Solar Energy,* vol. 128, 2016.
[22] A. Hariri, M. O. Faruque, R. Soman, and R. Meeker, "Impacts and interactions of voltage regulators on distribution networks with high PV penetration," *North American Power Symposium (NAPS),* 2015.
[23] A. Khoshkbar-Sadigh and K. M. Smedley, "The necessity of time-series simulation for investigation of large-scale solar energy penetration," in *IEEE Innovative Smart Grid Technologies Conference (ISGT)*, 2015.
[24] A. Pagnetti and G. Delille, "A simple and efficient method for fast analysis of renewable generation connection to active distribution networks," *Electric Power Systems Research,* vol. 125, pp. 133-140, 2015.
[25] J. W. Smith, W. Sunderman, R. Dugan, and B. Seal, "Smart inverter volt/var control functions for high penetration of PV on distribution systems," in *IEEE PES Power Systems Conference and Exposition (PSCE)*, 2011, pp. 1-6.
[26] R. Yan, B. Marais, and T. K. Saha, "Impacts of residential photovoltaic power fluctuation on on-load tap changer operation and a solution using DSTATCOM," *Electric Power Systems Research,* vol. 111, 2014.
[27] M. J. Reno, K. Coogan, J. Peppanen, and S. Grijalva, "Using Distribution LMP and Time-of-Delivery Pricing to Promote Optimal Placement and Increased Profitability of Residential PV Systems," in *North American Power Symposium*, 2014.
[28] 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.
[29] M. A. Abdel-warth, M. Abdel-akher, and M. M. Aly, "Quasi-Static Time-Series Simulation of Congested Power Systems with Wind Power Plant," in *17th International Middle East Power System Conference MEPCON'15*, Mansoura, 2015.
[30] T. Boehme, A. R. Wallace, and G. P. Harrison, "Applying Time Series to Power Flow Analysis in Networks With High Wind Penetration," *IEEE Transactions on Power Systems,* vol. 22, pp. 951-957, 2007.
[31] J. R. Aguero, P. Chongfuangprinya, S. Shao, L. Xu, F. Jahanbakhsh, and H. L. Willis, "Integration of Plug-in Electric Vehicles and distributed energy resources on power distribution systems," *2012 IEEE International Electric Vehicle Conference,* pp. 1-7, 2012.
[32] S. Shao, F. Jahanbakhsh, J. R. Aguero, and L. Xu, "Integration of PEVs and PV-DG in power distribution systems using distributed energy storage—Dynamic analyses," in *IEEE Innovative Smart Grid Technologies Conference (ISGT)*, 2013.
[33] M. Kleinberg, J. Harrison, and N. Mirhosseini, "Using energy storage to mitigate PV impacts on distribution feeders," in *IEEE Innovative Smart Grid Technologies Conference (ISGT)*, 2014.

[34] M. J. Reno, M. Lave, J. E. Quiroz, and R. J. Broderick, "PV Ramp Rate Smoothing Using Energy Storage to Mitigate Increased Voltage Regulator Tapping," in *IEEE Photovoltaic Specialists Conference*, 2016.

[35] P. Sirataranssophon, M. Hernandez, J. Peppanen, J. Deboever, and M. J. Reno, "Improved Load Modelling for Emerging Distribution System Assessments," in *CIRED Berlin 2020 Workshop*, 2020.

[36] J. Deboever, M. Hernandez, J. Peppanen, P. Sirataranssophon, and M. J. Reno, "Impact of AMI Data Time Granularity on Quasi-Static Time-Series Load Flow Simulation," in *IEEE/PES Transmission and Distribution Conference and Exposition (T&D)*, 2020.

[37] K. Ashok, M. J. Reno, L. Blakely, and D. Divan, "Systematic Study of Data Requirements and AMI Capabilities for Smart Meter Analytics," in *IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*, 2019.

[38] J. Peppanen, J. Grimaldo, M. J. Reno, S. Grijalva, and R. Harley, "Increasing Distribution System Model Accuracy with Extensive Deployment of Smart Meters," *IEEE PES General Meeting,* 2014.

[39] Meghasai, S. Monger, R. Vega, and H. Krishnaswami, "Simulation of smart functionalities of photovoltaic inverters by interfacing OpenDSS and Matlab," *2015 IEEE 16th Workshop on Control and Modeling for Power Electronics, COMPEL,* 2015.

[40] 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.

[41] M. J. Reno, J. E. Quiroz, O. Lavrova, and R. H. Byrne, "Evaluation of Communication Requirements for Voltage Regulation Control with Advanced Inverters," in *North American Power Symposium (NAPS)*, 2016.

[42] J. E. Quiroz, M. J. Reno, O. Lavrova, and R. H. Byrne, "Communication requirements for hierarchical control of volt-VAr function for steady-state voltage," in *IEEE Innovative Smart Grid Technologies Conference (ISGT)*, 2017.

[43] J. A. Azzolini, M. J. Reno, and K. A. W. Horowitz, "Evaluation of Curtailment Associated with PV System Design Considerations," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2020.

[44] M. Rylander, M. J. Reno, J. E. Quiroz, F. Ding, H. Li, R. J. Broderick, B. Mather, and J. Smith, "Methods to Determine Recommended Feeder-Wide Advanced Inverter Settings for Improving Distribution System Performance," in *IEEE Photovoltaic Specialist Conference (PVSC)*, 2016.

[45] J. A. Azzolini and M. J. Reno, "Impact of Load Allocation and High Penetration PV Modeling on QSTS-Based Curtailment Studies," in *IEEE Power & Energy Society General Meeting (PESGM)*, 2021.

[46] Y. P. Agalgaonkar, B. C. Pal, and R. A. Jabr, "Distribution voltage control considering the impact of PV generation on tap changers and autonomous regulators," *IEEE Transactions on Power Systems,* vol. 29, pp. 182-192, 2014.

[47] M. Kraiczy, M. Braun, G. Wirth, S. Schmidt, and J. Brantl, "Interferences between local voltage control strategies of a hv/mv- transformer and distributed generators," in *Photovoltaic Solar Energy Conference and Exhibition*, 2013.

[48] R. Michael, "Thesis: Online optimization of capacitor switching in electric power distribution systems," ed: Drexel University, 2015.

[49] K. P. Schneider, J. C. Fuller, and D. Chassin, "Evaluating conservation voltage reduction: An application of GridLAB-D: An open source software package," *IEEE Power and Energy Society General Meeting,* pp. 1-6, 2011.

[50] J. Johnson, A. Summers, R. Darbali-Zamora, C. Hansen, M. J. Reno, and e. al., "Optimal Distribution System Voltage Regulation using State Estimation and DER Grid-Support Functions," Sandia National Laboratories, SAND2020-2331, 2020.
[51] J. Johnson, J. Quiroz, R. Concepcion, F. Wilches-Bernal, and M. J. Reno, "Power system effects and mitigation recommendations for DER cyberattacks," *IET Cyber-Physical Systems: Theory & Applications,* vol. 4, no. 3, pp. 240-249, 2019.
[52] X. Zhang, M. J. Reno, and S. Grijalva, "A Time-Variant Load Model Based on Smart Meter Data Mining," in *IEEE PES General Meeting*, 2014.
[53] M. J. Reno, J. Deboever, and B. Mather, "Motivation and Requirements for Quasi-Static Time Series (QSTS) for Distribution System Analysis," in *IEEE PES General Meeting*, 2017.
[54] R. Seguin, J. Woyak, D. Costyk, J. Hambrick, and B. Mather, "High-Penetration PV Integration Handbook for Distribution Engineers," National Renewable Energy Laboratory, Golden, CO, NREL/TP-5D00-63114, 2016.
[55] R. J. Broderick, J. E. Quiroz, M. J. Reno, A. Ellis, J. Smith, and R. Dugan, "Time Series Power Flow Analysis for Distributed Connected PV Generation," Sandia National Laboratories, SAND2013-0537, 2013.
[56] M. J. Reno, K. Coogan, R. J. Broderick, J. Seuss, and S. Grijalva, "Impact of PV Variability and Ramping Events on Distribution Voltage Regulation Equipment," Denver, CO, IEEE Photovoltaic Specialists Conference, 2014.
[57] D. Paradis, F. Katiraei, and B. Mather, "Comparative analysis of time-series studies and transient simulations for impact assessment of PV integration on reduced IEEE 8500 node feeder," in *IEEE PES General Meeting*, 2013, pp. 1-5.
[58] EPRI. (2015). *Open Distribution System Simulator (OpenDSS)*. Available: http://sourceforge.net/projects/electricdss/
[59] V. Ramachandran, S. K. Solanki, and J. Solanki, "Steady state analysis of three phase unbalanced distribution systems with interconnection of photovoltaic cells," in *IEEE PES Power Systems Conference and Exposition (PSCE)*, 2011, pp. 1-7.
[60] M. J. Reno and K. Coogan, "Grid Integrated Distributed PV (GridPV)," Sandia National Labs, SAND2013-6733, 2013.
[61] M. J. Reno, C. W. Hansen, and J. S. Stein, "Global Horizontal Irradiance Clear Sky Models: Implementation and Analysis," Sandia National Laboratories, SAND2012-2389, 2012.
[62] M. Lave, J. Kleissl, and J. S. Stein, "A Wavelet-Based Variability Model (WVM) for Solar PV Power Plants," *IEEE Transactions on Sustainable Energy,* 2013.
[63] M. Lave and J. Kleissl, "Testing a wavelet-based variability model (WVM) for solar PV power plants," in *IEEE Power and Energy Society General Meeting*, 2012.
[64] M. Lave, J. Stein, and A. Ellis, "Analyzing and Simulating the Reduction in PV Powerplant Variability Due to Geographic Smoothing in Ota City, Japan and Alamosa, CO," in *IEEE Photovoltaic Specialists Conference*, Austin, TX, 2012.
[65] M. Lave, J. Quiroz, M. Reno, and R. Broderick, "High Temporal Resolution Load Variability Compared to PV Variability," presented at the Photovoltaic Specialists Conference, Portland, OR, 2016.
[66] M. Lave, M. J. Reno, and R. J. Broderick, "Characterizing local high-frequency solar variability and its impact to distribution studies," *Solar Energy,* vol. 118, pp. 327-337, 2015.
[67] J. Galtieri and M. J. Reno, "Intelligent Sampling of Periods for Reduced Computational Time of Time Series Analysis of PV Impacts on the Distribution System," in *IEEE Photovoltaic Specialists Conference*, 2017.
[68] M. J. Reno, R. J. Broderick, and L. Blakely, "Machine Learning for Rapid QSTS Simulations Using Neural Networks," in *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, 2017.

[69] L. Blakely, M. J. Reno, and R. J. Broderick, "Decision tree ensemble machine learning for rapid QSTS simulations," in *2018 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2018, pp. 1-5.
[70] L. Blakely, M. J. Reno, and R. J. Broderick, "Evaluation and Comparison of Machine Learning Techniques for Rapid QSTS Simulations," Sandia National Laboratories, SAND2018-8018, 2018.
[71] J. Deboever, X. Zhang, M. J. Reno, R. J. Broderick, S. Grijalva, and F. Therrien, "Challenges in reducing the computational time of QSTS simulations for distribution system analysis," Sandia National Laboratories, SAND2017-5743, 2017.
[72] W. H. Kersting, *Distribution System Modeling and Analysis*, Third ed. Boca Raton, FL: CRC Press, 2012.
[73] J. B. Ward and H. W. Hale, "Digital Computer Solution of Power-Flow Problems," *AIEE Trans. (Power Appl. Syst.),* vol. 75, pp. 398–404, 1956.
[74] R. J. Broderick, J. R. Williams, and K. Munoz-Ramos, "Clustering Method and Representative Feeder Selection for the California Solar Initiative," Sandia National Laboratories, SAND2014-1443, 2013.
[75] M. Reno and R. Broderick, "Predetermined time-step solver for rapid quasi-static time series (QSTS) of distribution systems," *Proc. IEEE PES Innovative Smart Grid Technologies (ISGT) Conf,* pp. pp. 1-5, 2017.
[76] B. Mather, "Fast Determination of Distribution-Connected PV Impacts Using a Variable Time-Step Quasi-Static Time-Series Approach," in *IEEE Photovoltaic Specialists Conference (PVSC)*, 2017.
[77] M. J. Reno, J. A. Azzolini, and B. Mather, "Variable Time-Step Implementation for Rapid Quasi-Static Time-Series (QSTS) Simulations of Distributed PV," in *IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC)*, 2018.
[78] J. Deboever, S. Grijalva, M. J. Reno, and R. J. Broderick, "Fast Quasi-Static Time-Series (QSTS) for yearlong PV impact studies using vector quantization," *Solar Energy,* vol. 159, pp. 538-547, 2018.
[79] J. Deboever, S. Grijalva, M. J. Reno, X. Zhang, and R. J. Broderick, "Scalability of the Vector Quantization Approach for Fast QSTS Simulation," in *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, 2017.
[80] J. Deboever, S. Grijalva, M. J. Reno, and R. J. Broderick, "Algorithms to Effectively Quantize Scenarios for PV Impact Analysis using QSTS Simulation," in *IEEE Photovoltaic Specialists Conference*, 2018.
[81] Q. Li, B. Mather, J. Deboever, and M. J. Reno, "Fast QSTS for Distributed PV Impact Studies using Vector Quantization and Variable Time-Steps," in *IEEE Innovative Smart Grid Technologies (ISGT)*, 2018.
[82] X. Zhang, S. Grijalva, M. J. Reno, J. Deboever, and R. J. Broderick, "A Fast Quasi-Static Time Series (QSTS) Simulation Method for PV Impact Studies Using Voltage Sensitivities of Controllable Elements," in *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, 2017.
[83] M. U. Qureshi, S. Grijalva, and M. J. Reno, "A Fast Quasi-Static Time Series Simulation Method for PV Smart Inverters with VAR Control using Linear Sensitivity Model," in *IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC)*, 2018.
[84] M. U. Qureshi, S. Grijalva, M. J. Reno, J. Deboever, X. Zhang, and R. Broderick, "A Fast Scalable Quasi-Static Time Series Analysis Method for PV Impact Studies using Linear Sensitivity Model," *IEEE Transactions on Sustainable Energy,* 2019.

[85] M. Qureshi, S. Grijalva, and M. Reno, "A Rapid Quasi-Static Time Series Method for Evaluating Current-Related Distributed PV Impacts including Feeder Loading and Line Losses," in *IEEE PES General Meeting*, 2019.
[86] M. U. Qureshi, "A Fast Quasi-Static Time Series Simulation Method Using Sensitivity Analysis to Evaluate Distributed PV Impacts," Georgia Institute of Technology, 2019.
[87] K. Mahmoud and M. Abdel-Nasser, "Efficient SPF approach based on regression and correction models for active distribution systems," (in En), *IET Renewable Power Generation,* vol. 11, no. 14, 2017.
[88] K. Mahmoud and M. Abdel-Nasser, "Fast yet Accurate Energy-Loss-Assessment Approach for Analyzing/Sizing PV in Distribution Systems Using Machine Learning," *IEEE Transactions on Sustainable Energy,* vol. 10, no. 3, 2019.
[89] D. Montenegro, G. A. Ramos, and S. Bacha, "A-Diakoptics for the Multicore Sequential-Time Simulation of Microgrids Within Large Distribution Systems," *IEEE Transactions on Smart Grid,* vol. 8, no. 3, pp. 1211-1219, 2017.
[90] D. Montenegro, G. A. Ramos, and S. Bacha, "An Iterative Method for Detecting and Localizing Islands Within Sparse Matrixes Using DSSim-RT," *IEEE Transactions on Industry Applications,* vol. 54, no. 1, pp. 675-684, 2018.
[91] R. Hunsberger and B. Mather, "Temporal Decompostion of Distribution System Quasi-Static Time-Series Simulation," in *IEEE PES General Meeting*, 2017.
[92] A. Latif and B. Mather, "Monte Carlo Based Method for Parallelizing Quasi-Static Time-Series Power System Simulations," in *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, 2018, pp. 1-6.
[93] D. Montenegro, R. C. Dugan, and M. J. Reno, "Open Source Tools for High Performance Quasi-Static-Time-Series Simulation Using Parallel Processing," in *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, 2017, pp. 3055-3060.
[94] J. A. Azzolini, M. J. Reno, and D. Montenegro, "Implementation of Temporal Parallelization for Rapid Quasi-Static Time-Series (QSTS) Simulations," in *IEEE Photovoltaic Specialists Conference (PVSC)*, 2019.
[95] M. J. Reno, R. J. Broderick, and S. Grijalva, "Formulating a Simplified Equivalent Representation of Distribution Circuits for PV Impact Studies," Sandia National Laboratories, SAND2013-2831, 2013.
[96] Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl, "Multiphase Distribution Feeder Reduction," *IEEE Transactions on Power Systems,* 2017.
[97] Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl, "Inversion Reduction Method for Real and Complex Distribution Feeder Models," *IEEE Transactions on Power Systems,* vol. 34, no. 2, pp. 1161-1170, 2019.
[98] M. J. Reno and R. J. Broderick, "Predetermined Time-Step Solver for Rapid Quasi-Static Time Series (QSTS) of Distribution Systems," in *IEEE Innovative Smart Grid Technologies (ISGT)*, 2017.
[99] X. Zhu and B. Mather, "DWT-Based Aggregated Load Modeling and Evaluation for Quasi-Static Time-Series Simulation on Distribution Feeders," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*, 2018, pp. 1-5.
[100] X. Zhu and B. Mather, "Data-Driven Distribution System Load Modeling for Quasi-Static Time-Series Simulation," *IEEE Trans. on Smart Grids*.
[101] M. Chamana and B. Mather, "Variability extraction and synthesis via multi-resolution analysis using distribution transformer high-speed power data," in *19th International Conference on Intelligent System Application to Power Systems (ISAP)*, 2017.

[102] National Renewable Energy Laboratory. (Jan. 22, 2016). *Solar Power Data for Integration Studies*. Available: http://www.nrel.gov/electricity/transmission/solar_integration_methodology.html#algorithm
[103] NREL. (2015). *National Solar Radiation Data Base*. Available: http://rredc.nrel.gov/solar/old_data/nsrdb/
[104] M. Hummon, E. Ibanez, G. Brinkman, and D. Lew, "Sub-Hour Solar Data for Power System Modeling from Static Spatial Variability Analysis," in *2nd International Workshop on Integration of Solar Power in Power Systems*, Lisbon, Portugal, 2012.
[105] M. Hummon, A. Weekley, K. Searight, and K. Clark, "Downscaling Solar Power Output to 4-Seconds for Use in Integration Studies," 2013.
[106] M. Baran and T. McDermott, "Distribution system state estimation using AMI data," in *Power Systems Conference and Exposition, 2009. PSCE'09. IEEE/PES*, 2009, pp. 1-3: IEEE.
[107] M. Lave, R. J. Broderick, and M. J. Reno, "Solar variability zones: Satellite-derived zones that represent high-frequency ground variability," *Solar Energy,* vol. 151, 2017.
[108] M. Lave, J. Kleissl, and E. Arias-Castro, "High-frequency irradiance fluctuations and geographic smoothing," *Solar Energy,* vol. 86, no. 8, pp. 2190-2199, 8// 2012.
[109] W. Zhang, W. Kleiber, A. R. Florita, B.-M. Hodge, and B. Mather, "Modeling and Simulation of High-Frequency Solar Irradiance," *IEEE Journal of Photovoltaics,* vol. v. 9, pp. 124-131, 2018.
[110] W. Zhang, W. Kleiber, A. R. Florita, B-M. Hodge, and B. Mather, "A Stochastic Downscaling Approach for Generating High-Frequency Solar Irradiance Scenarios," *Solar Energy* vol. 176, pp. 370-379. , 2018.
[111] P. Ineichen and R. Perez, "A new airmass independent formulation for the Linke turbidity coefficient," (in English), *Solar Energy,* vol. 73, no. 3, 2002.
[112] M. Lave, M. J. Reno, and R. J. Broderick, "Creation and Value of Synthetic High-Frequency Solar Inputs for Distribution System QSTS Simulations," presented at the IEEE PVSC 44, Washington, D.C., 2017.
[113] M. J. Reno and K. Coogan, "Grid Integrated Distributed PV (GridPV) Version 2," Sandia National Labs, SAND2014-20141, 2014.
[114] K. Coogan, M. J. Reno, and S. Grijalva, "Locational Dependence of PV Hosting Capacity Correlated with Feeder Load," in *IEEE PES Transmission & Distribution Conference & Exposition*, 2014.
[115] 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.
[116] J. A. Azzolini, M. J. Reno, and M. Lave, "Visualization Methods for Quasi-Static Time-Series (QSTS) Simulations with High PV Penetration," in *IEEE Photovoltaic Specialists Conference (PVSC)*, 2019.
[117] D. Montenegro, J. Gonzalez, and R. Dugan, "Multi-rate control mode for maintaining fidelity in Quasi-Static-Time-Simulations," in *IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, 2017.
[118] C. Hewitt, E. Meijer, and C. Szyperski. (2012, 05-15). *The Actor Model (everything you wanted to know, but were afraid to ask)*. Available: http://channel9.msdn.com/Shows/Going+Deep/Hewitt-Meijer-and-Szyperski-The-Actor-Model-everything-you-wanted-to-know-but-were-afraid-to-ask
[119] N. Instruments. (2013, 24, 05). *Actor Framework Template documentation*. Available: http://www.ni.com/white-paper/14115/en

[120] L. Jie, J. Eker, J. W. Janneck, L. Xiaojun, and E. A. Lee, "Actor-oriented control system design: a responsible framework perspective," *IEEE Transactions on Control Systems Technology,* vol. 12, no. 2, pp. 250-262, 2004.
[121] D. Montenegro, "Actor's based diakoptics for the simulation, monitoring and control of smart grids," Université Grenoble Alpes, 2015GREAT106, 2015.
[122] F. Therrien, M. Belletête, J. Lacroix, and M. J. Reno, "Algorithmic Aspects of a Commercial-Grade Distribution System Load Flow Engine," in *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*, 2017, pp. 1579-1584.
[123] Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl, "Multiphase distribution feeder reduction," *IEEE Trans. Power Syst., vol. 33, no. 2,* pp. pp. 1320-1328., 2018.
[124] B. Mather, "Fast Determination of Distribution-Connected PV Impacts Using a Variable Time-Step Quasi-Static Time-Series Approach," *Proc. IEEE 44th Photovoltaic Specialists Conference (PVSC) 2017,* pp. pp. 1-6, 2017.
[125] W. H. Kersting, "Radial distribution test feeders," *IEEE Trans. Power Syst,* vol. vol. 33, no. 2, pp. 975–985, 1991.
[126] Z. K. Pecenak, H. V. Haghi, C. Li, M. J. Reno, V. R. Disfani, and J. Kleiss, "Aggregation of Voltage-Controlled Devices During Distribution Network Reduction," *IEEE Transactions on Smart Grid,* 2020.
[127] M. Lave and A. Weekley, "Comparison of high-frequency solar irradiance: Ground measured vs. satellite-derived," in *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, 2016, pp. 1101-1106.
[128] D. Montenegro and R. C. Dugan, "OpenDSS and OpenDSS-PM open source libraries for NI LabVIEW," in *2017 IEEE Workshop on Power Electronics and Power Quality Applications (PEPQA)*, 2017, pp. 1-5.
[129] M. Lave, M. J. Reno, and R. J. Broderick, "Implementation of Synthetic Cloud Fields for PV Modeling in Distribution Grid Simulations," in *2018 IEEE 7th World Conference on Photovoltaic Energy Conversion (WCPEC)*, 2018.
[130] J. Seuss, M. J. Reno, R. J. Broderick, and S. Grijalva, "Analysis of PV Advanced Inverter Functions and Setpoints under Time Series Simulation," Sandia National Laboratories, SAND2016-4856, 2016.

DISTRIBUTION

Email—External

Name	Company Email Address	Company Name
Guohui Yuan	Guohui.Yuan@EE.Doe.Gov	U.S. Department of Energy
Jeremiah Miller	Jeremiah.Miller@ee.doe.gov	U.S. Department of Energy
Kemal Celik	Kemal.Celik@EE.DOE.Gov	U.S. Department of Energy

Email—Internal

Name	Org.	Sandia Email Address
Robert Broderick	8812	rbroder@sandia.gov
Matthew Reno	8813	mjreno@sandia.gov
Matthew Lave	8812	mlave@sandia.gov
Joseph Azzolini	8813	jazzoli@sandia.gov
Logan Blakely	8813	lblakel@sandia.gov
Summer Ferreira	8812	srferre@sandia.gov
Technical Library	01177	libref@sandia.gov

This page left blank

Sandia National Laboratories

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.