Monte Carlo-Based Method for Parallelizing Quasi-Static Time-Series Power System Simulations

Preprint

Aadil Latif and Barry Mather
National Renewable Energy Laboratory

*Presented at the 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)
Boise, Idaho
June 24–28, 2018*

Suggested Citation

Latif, Aadil, and Barry Mather. 2018. "Monte Carlo-Based Method for Parallelizing Quasi-Static Time-Series Power System Simulations: Preprint." Golden, CO: National Renewable Energy Laboratory. NREL/CP-5D00-70714. <https://www.nrel.gov/docs/fy18osti/70714.pdf>.

**NREL is a national laboratory of the U.S. Department of Energy
Office of Energy Efficiency & Renewable Energy
Operated by the Alliance for Sustainable Energy, LLC**

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

Conference Paper
NREL/CP-5D00-70714
September 2018

Contract No. DE-AC36-08GO28308

NOTICE

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by Solar Energy Technologies Office. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

This report is available at no cost from the National Renewable Energy Laboratory (NREL) at www.nrel.gov/publications.

U.S. Department of Energy (DOE) reports produced after 1991 and a growing number of pre-1991 documents are available free via www.OSTI.gov.

Cover Photos by Dennis Schroeder: (left to right) NREL 26173, NREL 18302, NREL 19758, NREL 29642, NREL 19795.

NREL prints on paper that contains recycled content.

Monte Carlo Based Method for Parallelizing Quasi-Static Time-Series Power System Simulations

Aadil Latif
National Renewable Energy Lab
aadil.latif@nrel.gov

Barry Mather
National Renewable Energy Lab
barry.mather@nrel.gov

Abstract—Both commercial and open source power system simulators run quasi-static time-series (QSTS) simulations sequentially. For yearlong high-resolution load profiles this sequential execution of load flows incurs long computation times. Parallelizing QSTS simulations on multiple processors is one possible way to reduce the computation time. Parallelization however introduces errors in the simulation results as the initial state of the system at the beginning of each parallelized time period is not known. In this work a Monte Carlo-based approach has been proposed to estimate the initial state of the system for each parallel simulation run to mitigate the errors in the final results. Classical sequential QSTS simulation is the chosen base case against which all other methods have been compared. Results presented in this paper show that the proposed method improves the simulation results considerably. This paper also discusses possible sensitivities of the proposed method to a number of tunable parameters. Additionally, sensitivity analysis results for a number of parameters have also been presented in the results section.

Index Terms—quasi-static time-series, QSTS, distribution simulation, parallelization, Monte Carlo,

NOMENCLATURE

M	is the total number of active controllers
N	is the number of parallel simulation runs
$nBins$	is the number of bins used to calculate fitting error
T_j	is the total minutes for the j^{th} parallel run
i	is the index of the controller
j	is the index of the simulation in and $\in (1, N)$
k	is the index for data bin
t	is the index for time in minutes
$mode$	is the selected mode for choosing initial controller settings
$X_{i_{act}}^t$	is the base case state of the i^{th} controller at t
$X_{i_{mode}}^t$	is the state of the i^{th} controller at t using $mode$ as initial condition
P_l^t	is the per unit load value at time t
$\bar{P}_l^k$	is the lower limit for the k^{th} bin
ξ	is the total error compared to base case
ε	is the least sum of square error for distribution fitting
T_w	is the time window width in minutes
T_h	is the time window height in days

T_{off} is the time offset from midnight in hours
 $f(x)$ is the frequency of variable $\bar{x}$
 $g(D_p, x)$ is the value from the probability density function at x for a given distribution D with parameters p

I. INTRODUCTION

Power system time-series simulations can be broadly categorized in two groups namely, dynamic simulations (root mean square simulations and electromagnetic transient simulations) and quasi-static time-series simulation (QSTS) [1]. QSTS simulations are used within the distribution utility setting to model the operation of automatic voltage regulation equipment allowing the optimization of system operation and planning [2] [3]. QSTS simulations assume that, for each time period solved, any circuit dynamics have settled and the system has converged to a steady state condition.

QSTS solvers most commonly use Newton Raphson (NR) or forward-backward sweeping methods to calculate state variables at each time step. These processes are iterative and computationally intensive, particularly when completing high-temporal resolution QSTS simulations [4]. For QSTS simulations the previously calculated system states are used as initial conditions for the next time step [5]. The time for a simulation run depends on a number of factors including the size of the network model, time duration of the simulation study and time resolution of the profile data. Increasing the simulation time step may not always be possible, because for larger time steps fidelity and resolution of the generated results is reduced. In [6] [7] the authors suggest using profiles with temporal resolutions less than 5 seconds to accurately all capture systems metrics.

QSTS solvers typically implement three nested loops as shown in the flowchart in Figure 1. Traditional QSTS solvers operate sequentially in time with inner most loop using either a NR solver or a forward-backward sweeping solver to calculate system states. The control iteration loop ensure that controllers within the power system converge to a steady state solution at each time period. These iterations are computationally expensive as each iteration requires recalculation of the power flow solution. Finally, the outer most iterations loop is used to increment time in a sequential manner.

Many simulation studies require running simulations for multiple years to be able to estimate O&M costs, understand and account for variations resulting from seasonal variability as well as load growth. Running a simulation for one year with 1 second temporal resolution requires solving fully converged power flows over 30 million times. For a large network this may take many several days or even weeks [8]. In [5], the authors have detailed several factors that pose a challenge in reducing computation time of QSTS simulations. These factors include but are not limited to *circuit complexity*, *number of power flow solutions*, *time step time dependency*, *multiple valid solutions*, *controllable element interactions* and *data logging and metric calculation*. In literature a number of methods have been proposed to reduce the computational time of QSTS simulations. All these methods aim to address one or more of the bottlenecks listed above.

Model reduction techniques aim to reduce the set of equations that need to be solved using a NR solver thus reducing complexity of the circuit. In [9] the authors have proposed a method for network reduction for networks with distributed generation. The methods implemented on the IEEE 123 node test feeder is able to reduce the simulation time by up to 60% while ensuring that the error in voltage remains under 0.1%. Reiman et al. [10] have proposed a segment substitution based method for systematic network reduction. Results presented in the paper show that the method is able to reduce certain networks up to 98% while keeping the error in voltage magnitude lower than 0.2%. A feeder reduction method proposed in [11] has been tested on a real California feeder. Results show that computation time can be reduced by as much as 96% while keeping the mean error in voltage magnitude less than 0.21%.

Intelligent solvers have also been proposed in literature that aim to reduce the number of control iterations by converging to the steady state solution faster. In [12] an algorithm has been proposed that aims to reduce the number of iterations required to converge the control iterations loop more quickly when modeling smart PV inverters. Results presented in the paper show that for the chosen test case, the algorithm was able to reduce the control iterations by up to 75%.

A number of authors have proposed using dynamic time step rather than fixed time step to speedup QSTS simulations by reducing the number of time loop iterations. In [8] a backtracking algorithm with binary search has been proposed that is capable of reducing computation time by over 90% by dynamically switching between small and large time steps. In [4] a predetermined time step solver is proposed that uses only input profiles to adjust the time step, hence, implementation is straight-forward. The authors report 95% reduction in computation time.

This work explores solutions for parallelizing QSTS simulations to reduce computational time without compromising the resolution of the results or network model detail. The main idea is to divide the load and generation profiles and run multiple simulations in parallel using all available computational resources as described in [13]. Dividing the input profiles and running simulations in parallel leads to potential error in the

Fig. 1: Classical methods for improving simulation time

simulations due to the introduced discontinuities in the system state at the beginning of each temporal division. The initial conditions at the point where profiles have been divided are not known a priori and thus must be estimated. Randomly setting initial conditions at this point may lead to large deviations between the calculated and actual state, as more than one valid power flow solutions may exist and the solver may converge to the incorrect solution. Also, controllers within a power system typically implement relatively wide deadband-based control architectures resulting in the “stickiness” of control settings, sometimes over long periods of time. This results in potentially persistent errors in QSTS simulations over long

Fig. 2: Source of error in QSTS parallelization

periods of simulation time. It is also important to understand that increasing the number of temporal discontinuities leads to more states with unknown initial conditions as shown in Figure 2. The red dots represent the points at which the initial condition is not known. This work explores the possibility of using Monte Carlo-based approach to estimate the initial state of the system for each parallel simulation run.

This paper has been structured as follows. Section II presents an overview of the proposed method as well as the performance metrics used to compare the results to the base case. In Section III the four study cases investigated in this work have been detailed. Section IV gives a brief overview of the simulation setup. In Section V results have been presented for the chosen test network. Finally in Section VI, conclusions have been drawn based on the presented results.

II. PROBLEM FORMULATION

The QSTS parallelization tool developed in Python extracts data from N points, fits a distribution to each dataset and run Monte-Carlo using this distribution. Calculated metrics are then used as initial the state of the controllers and errors are calculated during post processing. The six steps have been detailed in this section.

A. Step 1: Pivoting and time stamping

In the first step, each load and generation profile is time stamped and the dataset is pivoted from an array to a 2 dimensional matrix. The process is shown in Figure 3 allows fast data manipulation for the next steps.

B. Step 2: Data extraction

In the second step, load and generation profiles are divided into N sub profiles each, where, N is the number of parallel QSTS simulations predefined by the user (shown as orange circles in Figure 3). Additionally, a new dataset is created for each sub profile for a predefined day and hour window (Figure 3). By varying the parameters of day and hour window it is possible to change the size of dataset used for distribution fitting. Choosing the right parameters is critical. A small data window will return small datasets which can be problematic for the next step - distribution fitting. Over fitting is a well known problem for small datasets that leads to selection bias [14].

Fig. 3: Simulation Setup

C. Step 3: Distribution Fitting

In the third step, the dataset generated in Step 2 is divided into two subsets. The first subset (colored blue in Figure 4 and 5) is used to evaluate the distribution that fit best to the given data. The second subset (colored pink in Figure 4 and 5) is used to validate that choice. Error ε_{D_p} in distribution fitting has been calculated using Equation 1.

The implemented code provides the option of fitting both unimodal distributions (using SciPy Python library [15]) and multi modal distributions (using Scikit-learn Python library [16]). Multi modal distribution fitting uses kernel density-based method and is currently limited to multi modal Gaussian and Tophat distributions. SciPy library provides sixty distribution functions that can be fitted to the dataset. In this work the distributions have been limited to the distributions listed in Equation 2.

$$\varepsilon_{D_p} = \sum_{k=1}^{nBins} [f(\bar{P}_l^k) - g(D_p, \bar{P}_l^k)]^2 \quad (1)$$

Where,

$$D \in \begin{cases} \{normal, beta, expon, \\ gamma, laplace, lognorm, \\ uniform, triang, weibull\} & \text{if unimodal} \\ \{normal, tophat\} & \text{if multimodal} \end{cases} \quad (2)$$

$$\bar{\varepsilon}_{D_p} = \frac{\varepsilon_{D_p}(training) + \varepsilon_{D_p}(Validation)}{2} \quad (3)$$

D. Step 4: Run Monte Carlo Simulation

In step four, the distribution function with the lowest average error $\bar{\varepsilon}_{D_p}$ for training and validation datasets (Equation 3) is chosen as the representative distribution function to estimate the initial condition for a given sub profile. Next Monte Carlo samples are generated for the chosen distribution function. It should be noted that a representative distribution function is chosen for each point in time point in the QSTS simulation where the initial state is unknown.

Fig. 4: Unimodal distribution fitting using SciPy

Fig. 5: Multimodal distribution fitting using Scikit-learn

E. Step 5: Run Parallel Simulation

A snapshot power flow solution is calculated for each Monte Carlo sample and the resulting controller states are saved in a matrix. The statistical properties of the resulting distributions of controller states are calculated. The properties include: mean, median, mode and combination mode. These parameters can potentially be used as estimates for the initial conditions. Figure 6 shows the calculated metrics for OLTC 1 along with the actual state of the regulator. Finally, parallel QSTS simulations are completed using the initial conditions determined in this step.

$$mode \in \{Random, Zero, Preinit, Mean, Median, Mode, CombMode\} \quad (4)$$

$$\xi_N = \sum_{j=1}^N \sum_{t=1}^{T_j} \sum_{i=1}^M |X_{i_{act}}^t - X_{i_{mode}}^t| \quad (5)$$

F. Step 6: Post Processing

At the processing stage, results from the parallelized simulation are compared to the base case (classical sequential QSTS simulation). The error metric ξ_N , calculated using Equation 5, has been defined as the absolute deviation between the actual and the calculated state summed over every controller and every time step.

III. CASE STUDY

The proposed method has been tested using the IEEE 13 node test feeder [17]. The circuit has three single-phase voltage regulators and two controlled capacitors. The control parameters for the regulators and capacitors are as defined in [6]. Four case studies are presented in this section to demonstrate the effectiveness of the Monte Carlo-based initial condition determination method.

A. Random

For the *Random* study case, the initial state for each controller is set randomly between the controller limits. For regulators this limit is an integer between -16 and 16. For capacitor the initial state is randomly set to either 0 or 1.

Fig. 6: Violin plot for OLTC 1 state after Monte Carlo simulation

B. Zero

For the *Zero* study case, the initial state for each controller is set to zero.

C. Preinit

For the *Preinit* study case, like the *Random* study case, the states of the controllers are initially set randomly. However, a preliminary power flow calculation is then carried out and whatever state the controllers converge to during this power flow calculation are set as the initial state for the full simulation run. This ensures that the initial states result in a plausible power flow solution.

D. Monte Carlo

This study case uses the method detailed in Section II. In this work, simulations have been carried out for each of the four metrics (*Mean*, *Median*, *Mode*, *Combined Mode*) calculated during *Step 5*. Each of the four metrics have been used as the initial state of the controllers.

IV. SIMULATION SETUP

The IEEE 13 node test feeder along with regulating equipment and control are modeled in OpenDSS [18]. OpenDSS additionally serves as the QSTS simulation engine for the parallelization tool implemented in Python. Python is an open source high level scripting language [19] that interfaces with the power system simulator easily. OpenDSS provides multiple external interfacing options to facilitate co-simulation. In this work, the COM interface has been used to interface Python with multiple instances of OpenDSS to realize temporally parallelized QSTS simulations capability. The simulations have been carried out on a server machine with 88 cores and the maximum number of parallel QSTS simulations carried out in this work is 70. For the Monte Carlo-based method 4000 samples were used to build up a distribution of the controller states.

V. RESULTS AND DISCUSSION

Figure 6 in Section II shows metrics calculated using the Monte Carlo approach along with the actual state of the system for $N = 5$. The figure shows that the proposed method does perform well to estimate the actual initial state of the system.

Figure 7 shows the impact of increasing the number of parallel simulations N on the computational time. As expected, increasing N decreases QSTS simulation time. However the time taken by the Monte Carlo tool to calculate the initial conditions increases. Post processing takes approximately 10 seconds irrespective of the value of the parameter N . For the IEEE 13 node test feeder the time required by the Monte Carlo tool is a significant portion of the total computation time. However, it is expected that, as the scale and detail of the modeled distribution system increases, the relative amount time taken by the Monte Carlo tool will decrease significantly. By parallelizing QSTS simulations the computational time can be decreased by up to 79.3% without comprising on profile fidelity/resolution or reducing the network details.

Figure 8 shows the sensitivity of the total error, ξ_N , to the number of parallel simulations, N . For this experiment hour window T_h and hour window T_w are fixed at 5 days and 5 hours respectively. For this work the hour offset parameter, T_{off} , is set to 2 hours to ensure that the input data profiles are split at 2am. This ensures that there is no PV generation and load variability is low, thus reducing the number of possible valid states for the voltage regulators and capacitor controllers. As expected, *Random* and *Zero* initialization methods result in the highest error values. Initial states calculated using the Monte Carlo-based method result in a reduction in the total error by up to 90%. It should be noted that, the error value tends to increase as the number of parallel QSTS simulations is increased. One final observation from the figure is that using the *Mean* consistently outperforms all other metrics.

In the next series of simulations, the sensitivity of total error, ξ_N , to hour window, T_w , is investigated. For this simulation study, the number of parallel QSTS simulations N is fixed at 70 with *Mean* used to estimate the initial state. Figure 9 shows the impact of varying the size of hour window on the calcu-

Fig. 8: Simulation Setup

lated error. For small values of T_w the calculated error is high. For small T_w values the resultant dataset is small and over fitting becomes an issue when fitting a distribution decreasing the accuracy of the representation of likely controller states via the Monte Carlo-based method. The error value decreases as T_w increases. After a certain value however, increasing T_w has no impact on the error value. The reason for this is that increasing dataset from this point on has limited impact on the probability distribution of the dataset. For the given test case, adjusting the value of T_w resulted in reduction in the error value by up to 26.3%.

In the final simulation based experiment, the sensitivity of errors calculated from all metrics to hour window T_w is investigated. For this experiment the number of parallel QSTS simulations N is again fixed at 70. Results presented in Figure 10 show that for the chosen test case, increasing the value of T_w results in error values calculated using metrics other than *Mean* converge to the error value calculated using the *Mean* metric as the initial state of the system. The underlying reason for the observed trend is that as the dataset size increases, median and mode of the dataset converges to the mean of the dataset.

Fig. 7: Simulation Setup

Fig. 9: Simulation Setup

Fig. 10: Simulation Setup

VI. CONCLUSIONS AND FUTURE WORK

Conventional high-fidelity QSTS simulations can require considerable computation time. Although a number of methods like network reduction and dynamic time step have been proposed in literature, the reduction in computational time comes at the cost of reduced network detail or lower resolution of the obtained results. These methods therefore might not be able to accurately capture all corner cases that are essential to assess system state and overall accurate system operational metrics.

This paper presents proposes a Monte Carlo-based method for determining the initial time-dependent system states, such as voltage regulator and capacitor controller states, to enable accurate temporally parallelized QSTS simulation. The Monte Carlo-based method performs well to mitigate errors introduced due to QSTS parallelization. Additionally, the paper provides sensitivity analysis for a number of controllable parameters with in the Monte Carlo-based method.

Future work on this topic includes extending the sensitivity analysis to other controllable parameters like T_{off} and T_h . Additionally, the impact of choosing multi modal distribution over unimodal distribution fitting on the total simulation error will also be investigated. Further work on understanding the impact on varying the number of Monte Carlo samples, N , is also important as fewer samples would further increase computational speed. Finally, in the next stage of this work, the simulations will be carried on a real feeder to quantify the gains that can be made using the proposed method for QSTS simulation.

ACKNOWLEDGMENT

This work was supported by the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08-GO28308 with the National Renewable Energy Laboratory. Funding for the work was provided by the DOE Office of Energy Efficiency and Renewable Energy, Solar Systems Integration Office.

REFERENCES

[1] L. Miller, L. Cibulka, M. Brown, and A. von Meier, "White paper: Electric Distribution System Simulation and Analysis Tools Status and

Research Gaps for Integration of Renewables and Electric Vehicles," California Institute for Energy and Environment, Tech. Rep. CEC-XXX-2013-XXX, 2013. [Online]. Available: http://uc-ciee.org/downloads/Task_1_1_White_Paper.pdf

[2] 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, no. 3, pp. 951–957, aug 2007. [Online]. Available: <http://ieeexplore.ieee.org/document/4282010/>

[3] B. A. Mather, "Quasi-static time-series test feeder for PV integration analysis on distribution systems," in *2012 IEEE Power and Energy Society General Meeting*. IEEE, jul 2012, pp. 1–8. [Online]. Available: <http://ieeexplore.ieee.org/document/6345414/>

[4] M. J. Reno and R. J. Broderick, "Predetermined time-step solver for rapid quasi-static time series (QSTS) of distribution systems," in *2017 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, apr 2017, pp. 1–5. [Online]. Available: <http://ieeexplore.ieee.org/document/8086045/>

[5] J. Deboever, X. Zhang, M. J. Reno, and ..., "Challenges in reducing the computational time of QSTS simulations for distribution system analysis," *Sandia National ...*, 2017. [Online]. Available: https://www.researchgate.net/profile/Matthew_Reno/publication

[6] 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*. Chicago: IEEE, 2017, p. 6. [Online]. Available: https://www.researchgate.net/profile/Matthew_Reno/publication

[7] 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 *2013 IEEE Power Energy Society General Meeting*, July 2013, pp. 1–5.

[8] B. Mather, "Fast Determination of Distribution-Connected PV Impacts Using a Variable Time-Step Quasi-Static Time-Series Approach," in *Conference: Presented at the 2017 IEEE 44th Photovoltaic Specialists Conference (PVSC), 25-30 June 2017, Washington, DC*. Golden: National Renewable Energy Laboratory (NREL), Golden, CO (United States), 2017, p. 6.

[9] Y. Jiang, N. Acharya, and Y. Pan, "Model reduction for fast assessment of grid impact of high penetration PV," in *2017 19th International Conference on Intelligent System Application to Power Systems (ISAP)*. IEEE, sep 2017, pp. 1–6. [Online]. Available: <http://ieeexplore.ieee.org/document/8071384/>

[10] A. P. Reiman, T. E. McDermott, M. Akcakaya, and G. F. Reed, "Electric Power Distribution System Model Simplification Using Segment Substitution," *IEEE Transactions on Power Systems*, pp. 1–1, 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/8047317/>

[11] Z. K. Pecanek, V. R. Disfani, M. J. Reno, and J. Kleissl, "Multiphase distribution feeder reduction," *IEEE Transactions on Power Systems*, pp. 1–1, 2017. [Online]. Available: <http://ieeexplore.ieee.org/document/7976321/>

[12] A. Nagarajan, B. Palmintier, F. Ding, B. Mather, and M. Baggu, "Improving advanced inverter control convergence in distribution power flow," in *2016 North American Power Symposium (NAPS)*. IEEE, sep 2016, pp. 1–5. [Online]. Available: <http://ieeexplore.ieee.org/document/7747887/>

[13] R. Hunsberger and B. Mather, "Temporal decomposition of a distribution system quasi-static time-series simulation," in *2017 IEEE Power Energy Society General Meeting, Chicago, IL, Jun. 16-20, 2017*, June 2017, pp. 1–5.

[14] Vapnik, Vladimir Naumovich and Vapnik, Vlamimir, *Statistical learning theory*. Wiley New York, 1998, vol. 1.

[15] E. Jones, T. Oliphant, and P. Peterson, "{SciPy}: open source scientific tools for {Python}." 2014.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[17] W. H. Kersting, "Radial distribution test feeders," in *2001 IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No.01CH37194)*, vol. 2, 2001, pp. 908–912 vol.2.

[18] R. C. Dugan, "Reference guide: The open distribution system simulator (opends)," *Electric Power Research Institute, Inc*, vol. 7, 2012.

[19] G. Van Rossum *et al.*, "Python Programming Language," in *USENIX Annual Technical Conference*, vol. 41, 2007, p. 36.