SAND2017-6572C

Scalability of the Vector Quantization Approach for Fast QSTS Simulation

Jeremiah Deboever[1], Santiago Grijalva[1], Matthew J. Reno[2], Xiaochen Zhang[1], and Robert J. Broderick[2]

1. Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332, USA

2. Electric Power Systems Research, Sandia National Laboratories, Albuquerque, NM, 87123, USA

Abstract — Quasi-static time-series (QSTS) provides the necessary simulation fidelity on the impact that new energy resources would have on a specific distribution system feeder. However, this simulation can often take 10-120 hours for a single study, on traditional computers. The vector quantization approach proposed in [1] demonstrated very attractive computational time reduction for feeders with limited number of input time-series profiles and few distribution system voltage control elements. In this work, we expend on this algorithm and address this issue to model feeders of any complexity while maintaining a high computational time reduction. We demonstrate a 98.7% reduction on a real distribution system feeder with 2969-bus, 3 load/PV profiles, and 8 voltage regulating elements.

I. Introduction

Interconnection studies for distributed energy resources (DERs) on distribution networks are currently performed using scenario-based simulation that includes peak load, back feeding, and high PV output cases. Because this approach solves a limited set of static power flows scenarios, all potential impacts due to DERs will not always be captured. Moreover, scenario-based simulations are not able to capture time-dependent states in the system, e.g. voltage regulating tap changers. A draft of the IEEE standard IEEE P1547.7 D110 recently proposed a time-series simulation meant to capture any temporal effects for interconnection studies [2]. Quasi-static time-series (QSTS) simulation solves static power flows chronologically at a specific granularity, which allows controllable elements to be modeled with their control logic including delays. This is important to model in interconnection impact studies because intermittent resources can create excessive controller actions that will reduce the equipment life.

QSTS simulation has not yet been widely used for impact studies of DERs, and more specifically PV, because of its computational burden. A yearlong study at 1-second granularity is suggested to capture the fast variation of relevant circuit quantities due to PV and the seasonal changes in the power demand [3]. This represents solving 31.5 million chronological power flows. The computation time for large feeders can be between 10-120 hours on conventional computers. The ability to simulate multiple locations, sizes, advanced inverter settings, or feeder configurations is therefore limited by the speed of the QSTS simulation. This makes fast time-series approximations very desirable.

There are several challenges in reducing the computational time of QSTS simulations due to the discontinuous and nonlinear nature of the simulation [4]. Six of the most significant challenges are: 1) the sheer number of power flows requires significant computational power. 2) some distribution feeders can be very complex when considering unbalanced loads or various controller logics, and their performance cannot be predicted without prior knowledge. 3) the time dependence between time steps requires each time step to be solved chronologically. 4) deadbands in controller logics create multiple feasible power flow solutions within their limits. Models must consider the hysteresis of the controllers. 5) the interaction between controllable elements can be extremely challenging to model, especially when considering the previous two challenges. 6) an accurate analysis for extended time-horizon simulations can require extensive data for post simulation analysis.

Different approaches in speeding up a QSTS simulation have been discussed in the literature: variable time-step, circuit reduction, A-diakoptics method, and vector quantization. Variable time-step increases the granularity of the simulation as a mean of computational time reduction [5]. Circuit reduction reduces the number of buses in the circuit to speed up the power flow solver [6]. A-diakoptics method divides large circuits into sub-areas to solve them separately [7]. Vector quantization takes advantage of similar power flow solutions to alleviate solving the non-linear unbalanced three-phase power flow equations [1], [8], [9]. This paper focuses on the vector quantization method presented in [1]. The scalability aspects discussed in that publication are addressed in this paper. More specifically, the algorithm is adapted to simulate realistic size feeders with more complexity including: size of the feeder, multiple load/PV profile inputs, larger number of controllable elements, and new types of controller logics (e.g. advanced inverter controls). This paper is organized as follows. In Section II, a summary of the vector quantization algorithm proposed in [1] is presented. The scalability of the algorithm in terms of feeder size, number of profiles, number of controllable elements, and types of controller logics are addressed in Section III. In Section IV, a simulation of a realistic test feeder is demonstrated. Section V discusses the computational time reduction opportunity. The importance of this work for system planners and operators is presented in the conclusion.

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.

II. VECTOR QUANTIZATION APPROACH

The vector quantization algorithm proposed in [1] groups similar power flow solutions in clusters to avoid the iterative power flow solver. Each time the power flow equations are solved, the solution is stored in a solution space for consequent time steps. If searching for a previously computed power flow solution is faster than solving the power flow equations, significant computational time reduction can be attained. This section briefly describes the vector quantization algorithm but additional details can be found in [1].

At each time step in the QSTS simulation, the algorithm cycles through two objectives: determining the power flow solution and determining whether an action is taken by the controllable elements on the feeder. The power flow solution depends on two sets of factors: the power injections on the feeder and the previous states of controllable elements. For PV interconnection studies, the power injections at time *t* is defined as the vector $\boldsymbol{u}_t$ where $\boldsymbol{d}_t$ is a vector of the different load profiles and $\boldsymbol{p}_{pv,t}$ is a vector of the PV output profiles. The previous states of controllable elements at time *t* defined as $\boldsymbol{l}_t$ includes the states of any elements that affect the voltage on the feeder, for instance any voltage regulating tap changers $\boldsymbol{r}_t$ or capacitor banks $\boldsymbol{c}_t$. The factors affecting the power flow solution can then be defined as vector $\boldsymbol{h}_t$.

$$\boldsymbol{u}_t \coloneqq [\boldsymbol{d}_t, \boldsymbol{p}_{pv,t}] \quad (1)$$

$$\boldsymbol{l}_t \coloneqq [\boldsymbol{r}_t, \boldsymbol{c}_t] \quad (2)$$

$$\boldsymbol{h}_t \coloneqq [\boldsymbol{u}_t,\ \boldsymbol{l}_t] \quad (3)$$

The power flow solution $\boldsymbol{g}_t$ is a vector of the bus voltage magnitudes and angles.

$$\boldsymbol{g}_t \coloneqq [\,|\boldsymbol{v}_t|\,, \boldsymbol{\theta}_t\,] \quad (4)$$

From the power flow solution $\boldsymbol{g}_t$ and the vector $\boldsymbol{h}_t$, any time-series values on the feeder can be analyzed without having to run the time-series simulation. A state vector $\boldsymbol{x}_t \coloneqq [\boldsymbol{h}_t, \boldsymbol{g}_t]$ can be stored in a state matrix $\boldsymbol{X}$ for post-simulation analysis.

The objective of the vector quantization algorithm is to bypass the iterative AC power flow solver by reassigning a similar solution if it has already been computed. At the beginning of the time step, a quantization logic is used to determine whether a similar (as defined by quantization clustering) power flow solution has been computed. A flow chart of the algorithm is presented in Fig. 1.

The quantization logic used to determine whether a solution exists must be faster than solving the power flow equations to make the algorithm attractive. The logic presented in [1] uses a matrix indexing method as shown in Algorithm 1. Matrix $\boldsymbol{S}$ is the solution space in which solutions are appended each time a new solution is computed. Matrix $\boldsymbol{M}$ is an indexing matrix used to determine whether or not a solution exists. Each value in vector $\boldsymbol{h}_t$ is associated to a unique dimension in the matrix $\boldsymbol{M}$. Thus, for an n-dimensional matrix $\boldsymbol{M}$, $indx = \boldsymbol{M}([\boldsymbol{h}_t])$ is the unique value at the location $\boldsymbol{M}(h_{t,1}, h_{t,2}, \ldots, h_{t,n})$ that is associated with a power flow solution.

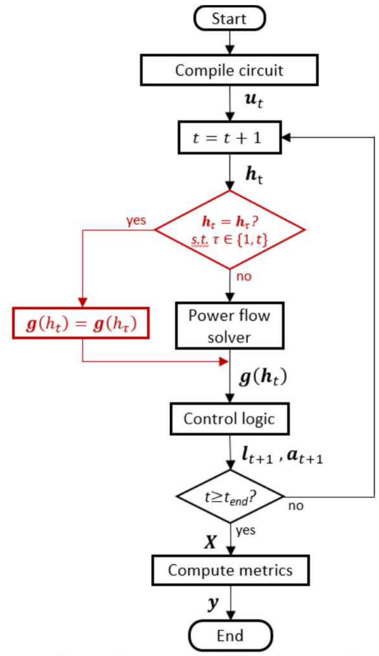

Fig. 1. QSTS flow diagram with proposed quantization algorithm (in red) [1] .

Algorithm 1: Quantization logic using matrix indexing [1]

$indx = \boldsymbol{M}([\boldsymbol{h}_t]),$
$if\ indx \neq 0\ :$
  $\boldsymbol{g}_t = \boldsymbol{S}(indx, :)$
$else:$
  $Solve\ power\ flow\ equations\ with\ \boldsymbol{h}_t,$
  $indx = size(\boldsymbol{S}, 1) + 1,$
  $\boldsymbol{S}(indx, :) = \boldsymbol{g}_t,$
  $\boldsymbol{M}([\boldsymbol{h}_t]) = indx,$
end

This matrix indexing logic runs quickly if the two matrices are preallocated in the memory. However, the indexing matrix $\boldsymbol{M}$ can become extremely large considering that it has the same number of dimensions as the length of vector $\boldsymbol{h}_t$, which is the total number of input time-series profiles and the number of controllable elements. The length of each dimension of $\boldsymbol{M}$ depends on the number of unique combinations of vector $\boldsymbol{h}_t$, which can be decreased using quantization. However, quantization introduces inherently an error in the data that can affect the accuracy of the simulation. Namely, metrics reported by the QSTS simulation, such as controller actions, under-/over-voltage, power loss, or constraint violations, can be falsely reported when quantization clusters are large and cover a wide range of slightly similar power flow solutions. Because the algorithm requires the indexing matrix $\boldsymbol{M}$ to be preallocated in the memory, a finite number of load and PV profiles and controllable elements can be modeled based on the available memory for this multi-dimensional matrix. The formulation of this algorithm in [1] limited the combined number of profile and controllable elements to 6. These issues are addressed in the following section.

III. Scalability of the Algorithm

A. Size of the feeder

The test circuit simulated in [1] only has 13 buses and thus, is not representative of a realistic distribution feeder. Although it has been demonstrated in the literature that an accurate model of a large feeder can be created based on a small number of buses [6], the vector quantization algorithm scales well with the size of the feeder. As the number of nodes in the circuit increases, the computational time to solve an individual power flow increases proportionally. The computational time of the proposed algorithm is a function of the number power flows computed ($N_{pf\ solved}$), the controller logic (T_{CL}), and an overhead time associated with the implementation of the algorithm itself (T_{VQ}). The average computational time to solve a power flow can be estimated from the computational time of QSTS simulation with a brute force approach ($T_{brute\ force}$).

$$T_{comp.} = \left(\frac{T_{brute\ force}}{31.5e^6}\right) N_{pf\ solved} + T_{CL} + T_{VQ} \qquad (5)$$

Because the quantization algorithm only computes power flows for a subset of the total number of time steps, the overall computational time of the vector quantization will increase much slower than if the power flow at each time step was computed. Furthermore, the overhead computational time associated with the control logic and the logic required to implement the algorithm is independent of the number of nodes in the circuit (i.e. the size of $\boldsymbol{h_t}$ and $\boldsymbol{M}$ in line 1 in Algorithm 1 do not change based on the number of nodes in the circuit). Therefore, as the feeder size increases, this overhead computational time becomes negligible compared to the computational time of the power flow solver, making the computational time reduction equal to the ratio of the number of power flows computed over the total number of time steps. For instance, a simulation with 31,500 unique computed power flows would reduce the computational time of a QSTS simulation by about a thousand.

B. Number of Load/PV Profiles

In the modified IEEE 13-bus test circuit discussed in [1], the indexing matrix $\boldsymbol{M}$ would be 6-dimensional with the two input time-series profiles, the three voltage regulators, and the switching capacitor bank. If the two profiles are quantized into 100 clusters, the regulators have 33 tap positions, and the capacitor bank has two states, the indexing matrix $\boldsymbol{M}$ would have 719 million entries. Obviously, additional input time-series profiles for each type of customer or different PV locations cannot be done with the current algorithm. Each additional input time-series increased the size of $\boldsymbol{M}$ exponentially. The proposed method to address this issue is not to treat profiles as individual dimensions in the indexing matrix $\boldsymbol{M}$ but as a single dimension representing a 'scenario' of profiles. A time-series vector $\boldsymbol{indx_u}$ can be created where each entry is the index of the first time that combination of profiles was experienced. The value of each profile at a specific time step t can easily be determined with $\boldsymbol{u}([\boldsymbol{indx_u}])$.

After quantizing the profiles, the time-series vector $\boldsymbol{indx_u}$ can be created representing the 'scenario' of profiles at each time-step. The vector defining unique power flow solutions becomes:

$$\boldsymbol{h_t} = \left[indx_{\boldsymbol{u},t}, \boldsymbol{l_t}\right] \qquad (6)$$

By pre-processing the profiles, the number of dimensions for profiles in the indexing matrix is limited to one no matter how many input time-series profiles are considered. Moreover, the size of the indexing matrix is also reduced since not every unique vector $\boldsymbol{u_t}$ is experienced over the time horizon. This method is only possible because the dataset for the profiles is known prior to the time-series simulation. With this indexing method, the vector quantization algorithm is not affected by the number of profiles considered in the simulation, whether it is individual load profiles based on Advanced Metering Infrastructure (AMI) data or multiple PV profiles based on scale and location.

C. Number of Controllable elements

A similar approach to the one discussed above can be used to reduce the memory requirement for the controllable elements in the indexing matrix. However, unlike the profiles, the states of those elements are not known prior to the time-series simulation. Thus, the entire space must be created for the current indexing method to work. From analyzing the simulation results on the modified IEEE 13-bus test circuit from [1], one can find that this space is extremely sparse as shown in Fig. 2. To take advantage of this sparsity, a controllable element matrix $\boldsymbol{L}$ can be created with each dimension representing the state of an element. This matrix is initially a zero-matrix that is populated with indices as the simulation progresses. Each time a set of controllable element states are experienced for the first time, a column is appended to the indexing matrix $\boldsymbol{M}$ and an index referring to the column number is stored in matrix $\boldsymbol{L}$. The following two equalities are then implemented before line 1 of Algorithm 1.

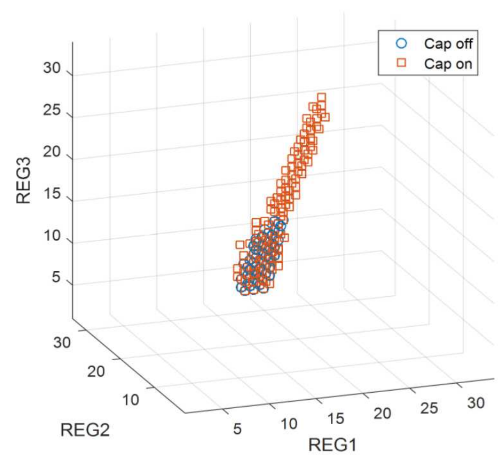

Fig. 2. Regulator tap position and capacitor status combinations that are experienced in a yearlong QSTS simulation for the modified IEEE 13-bus test circuit

$$indx_L = \boldsymbol{L}([\boldsymbol{l}_t]) \quad (7)$$

$$\boldsymbol{h}_t = [indx_{\boldsymbol{u}}, indx_L] \quad (8)$$

At the end of the yearlong simulation, matrix $\boldsymbol{M}$ is a two-dimensional matrix with a size of the number of profile scenarios by the number of controllable element states experienced. This formulation takes advantage of the sparsity of the space due to the correlation between controllable elements and possibly reduce the pre-allocated space for the controllable elements in the indexing matrix. In the 13-bus test case referenced above, the three phases are relatively balanced, explaining why the voltage regulator tap positions of each phase are closely correlated. Not every entry in matrix $\boldsymbol{L}$ needs to have a value referring to the indexing matrix $\boldsymbol{M}$.

This matrix decomposition reduces the requirement to preallocate large datasets in memory. This concept can be further expended as the number of controllable elements increases, which increases the size of $\boldsymbol{L}$ but also its sparsity. Furthermore, the computational time of the algorithm will not be affected since Eq. (7) is only performed when a controller action is performed.

D. Type of controllers

Distribution feeders can have a wide variety of devices with control logic: energy storage systems, smart inverters, regulators, capacitor banks, etc. The number of devices implemented on a feeder can rapidly grow as some of them are often installed with other DERs (e.g. advanced inverters with PV systems). In fact, the need to determine the settings for these new types of controls is an important application of QSTS. For example, in order to determine the optimal settings for energy storage controls [10] or advanced inverter controls [11], potentially thousands of QSTS simulations need to be performed with different settings to fully study any potential interactions between controls and potential benefits.

One of the advantages of the proposed quantization algorithm is that it is robust to any type of control logic. Generally speaking, controllers operate based on a specific signal, whether it is time, voltage/current magnitudes, power factor, price, etc. The type of signal will dictate how the control logic is implemented in the quantization algorithm. For this purpose, the controllers can be grouped into three categories.

First, most controllers are dependent on a signal derived from the power flow solution (i.e. voltage/current magnitude, power factor,…) or have an hysteresis that prohibits them to be computed ahead of the time-series simulation. For instance, capacitor banks can operate based on a voltage signal and have delays to reduce excessive operation. To accurately model these types of controllers in a QSTS simulation, their logic is implemented at the end of each time step. States from the power flow solution can be used as control signals and controller states (i.e. equipment states or delay accumulators) can model the hysteresis within the controllers. In the vector quantization algorithm, these equipment states (e.g. tap position) are referred to in the matrix $\boldsymbol{L}$ since they affect the power flow solutions. Because the algorithm goes through each time-step, delay accumulators can easily keep track of any expiring delays within the controllers as the simulation progress through time. As discussed in III.C, the algorithm is not limited by the number of these controllers in the feeder model.

Second, the controllers without delays and no hysteresis will always have the same outcome for a given power flow solution and system state. Thus, their control logic can be implemented separately into the power flow module, with vector quantization only storing the solutions. Because each power flow is defined by a unique vector $\boldsymbol{u}$, the algorithm does not need to go through that control logic each time a power flow has already been computed. For example, this subtle difference is especially advantageous for feeders with multiple advanced inverters with volt/var capabilities programmed to control the power factor of their output based on the voltage magnitude of the system. In a QSTS simulation, this is simulated with a control logic iteratively changing the output of the inverter and re-computing the power flow until it converges. Under a brute-force approach to the QSTS simulation, this would be done at each time step, which will impact the computational time. However, the proposed algorithm reduces the number of times it goes through this type of controller logic since only the final converged solution is stored by the vector quantization algorithm.

Third, any controllers operating solely on a signal known ahead of the QSTS simulation (i.e. price, time, etc.) and does not depend on any states from the power flow solution could be preprocessed with the other input time-series profiles as a scenario (see subsection III.B). For instance, the power output of an energy storage system or the charging of an electric vehicle can be based on a predetermined schedule. In that case, the power output profile is time-dependent and is treated similar to a PV or load profile in the input vector $\boldsymbol{u}$.

Not only has the number of controllable elements with hysteresis been addressed in III.C, the proposed vector quantization algorithm is robust to different types of controllers, since most would fit in one of the three previously stated categories.

IV. SIMULATION

The scalability of the algorithm is demonstrated using an actual distribution feeder with 2969 buses (5469 nodes) and 8 controllable elements – 4 three-phase switching capacitor banks, 3 single-phase line voltage regulators, and a three-phase substation load tap changer (Fig. 3). This realistic distribution feeder was modeled in OpenDSS through a MATLAB interface

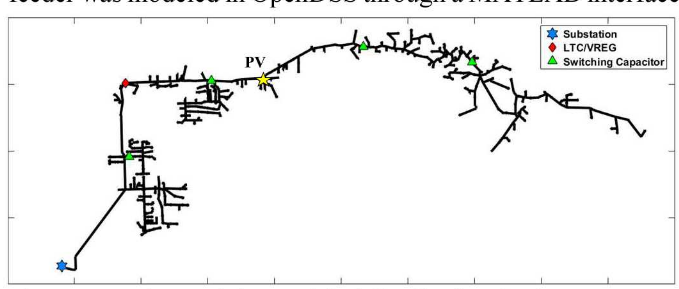

Fig. 3. Topology of the modeled distribution feeder.

and simulated on a Window 10 computer with 32GB of memory and a 3.50GHz processor.

A three-phase centralized PV system is modeled in the middle of the feeder with an irradiance profile similar to [1]. The loads are categorized to simulate two types of customers: residential (single-phase) and commercial customers (three-phase). The 1131 residential customers are mostly on the laterals and account for 4.2 MW of peak power. The 317 commercial customers account for 1.7 MW. The two load profiles are show in boxplots grouped by hour in Fig. 4 and Fig. 5.

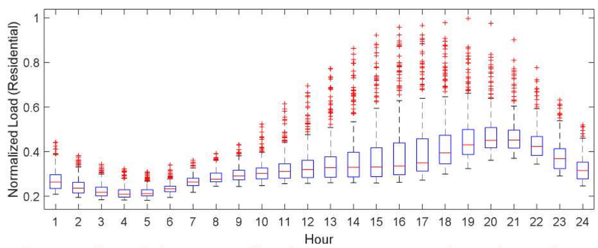

Fig. 4. Boxplot of the normalized power averaged per hour for a yearlong residential load profile.

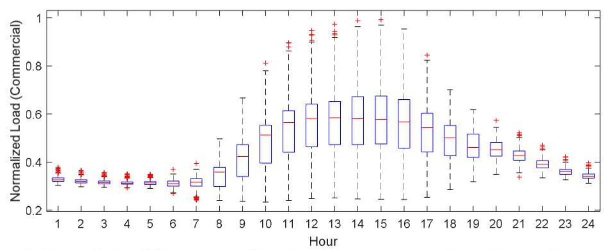

Fig. 5. Boxplot of the normalized power averaged per hour for a yearlong commercial load profile.

As expected, the residential profile has a tendency to peak around 7p-8p while the commercial profile has a higher load during business hours. When both profiles are used to model the load on a distribution feeder, not every combination of load multiplier will be experienced (i.e. commercial load at 1 p.u. and residential load at 0.2 p.u.). Thus, this space is relatively sparse and can be illustrated with a heat map of the reoccurrence of each combination (Fig. 6.).

The white pixels in the heat map illustrate a scenario of profiles that is never experienced in a yearlong simulation. Similar heat maps can be draw between profiles if more are considered. This illustrates the advantage of the formulation in section III.B. Each of the three yearlong profiles have a resolution of 1 second and a precision of 10^{-5} p.u. Without vector quantization, they have a total of 31,448,790 unique scenarios and a power flow is computed for every time step when considering the controllable element states. The number of computed power flows can be reduced through vector quantization, which ultimately reduces the precision of the profiles and clusters values together (TABLE I).

Without the updated algorithm discussed in this paper, the indexing matrix ***M*** used in [1] for the 200-cluster case would have 1.5 x 10^{14} entries (200^3 x 33^4 x 2^4) making it impossible to be preallocated on a conventional computer. With the proposed

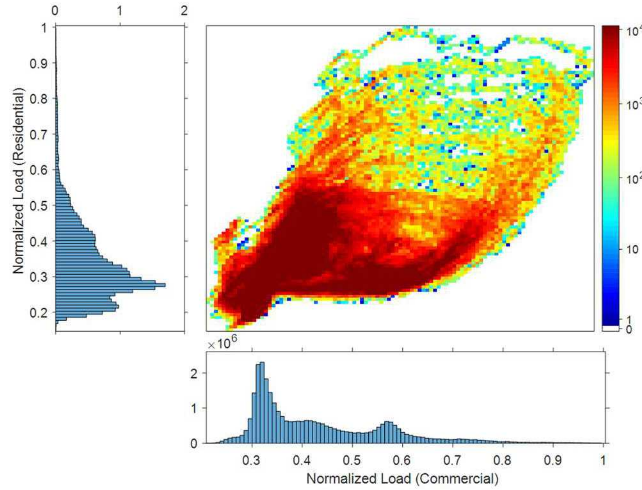

Fig. 6. Heat map of the number of time steps that have the same scenario (combination of multiplier values) for a yearlong profile with 1-second resolution. Note that the profiles were clustered for illustrative purposes.

algorithm, the indexing matrix had only 3.3 x 10^9 entries (863931 x 3829), which is equivalent to 13.2 GB. The novel implementation of the indexing matrix reduces the memory required to 0.002% of the original size for this larger circuit.

TABLE I. COMPARISON OF RUNNING A QSTS SIMULATION WITH A BRUTE FORCE APPROACH OR WITH THE PROPOSED VECTOR QUANTIZATION (VQ) ALGORITHM.

	# of scenarios	**# of computed pf**
Brute-force	31,448,790	31,499,600 (100%)
VQ (200 clusters)	863,931	1,722,978 (5.5%)
VQ (100 clusters)	218,187	716,443 (2.3%)
VQ (50 blusters)	43,743	253,870 (0.8%)

By reducing the number of computed power flows through vector quantization, the computational time of the simulation is drastically reduced with minimal impact on the accuracy of the simulation. In TABLE II, the computational time and accuracy is presented for each of the vector quantization cases.

TABLE II. QSTS SIMULATION RESULTS SHOWING THE ACCURACY AND COMPUTATIONAL TIME OF DIFFERENT VECTOR QUANTIZATION (VQ) CASES.

	LTC	**REG1**	**REG2**	**REG3**	**Time [hrs]**
Brute-force	2072	9031	12656	10783	36.1
VQ (200 clusters)	+0.5%	+2.3%	+2.5%	+2.4%	2.62
VQ (100 clusters)	+1.6%	+7.2%	+5.4%	+5.5%	1.17
VQ (50 blusters)	+2.8%	+5.4%	+5.1%	+5.0%	0.47

These results demonstrate the effectiveness of the proposed vector quantization algorithm to run a QSTS simulation of a real distribution feeder and address the robustness of the algorithm. Note that the computational time reported for the vector quantization algorithm includes time-consuming calls through a COM interface that would not exist if implemented directing in OpenDSS or other commercial software such as CYME. (refer to [1] for more information).

V. Computational Time Reduction Opportunity

As previously discussed, each power flow solution is defined by a unique vector ***h*** based on the time-series profiles and the states of controllable elements with hysteresis. Through vector quantization, the number of unique power flows computed over the time horizon can be reduced to improve the computational time of the simulation. For instance, the precision of load profiles can be reduced to increase the number of repeated values. However, this approximation introduces an error in the simulation that can impact the accuracy of the results. Thus, not all indices should be quantized equally depending on their impact on the power flow solution and the type of study conducted. Each profile may have a different impact on the feeder depending on the location, size, etc. of the element it is associated to. For example, the profile associated with a large centralized PV system may impact the operation of a voltage regulator, while the profile of a smaller system downstream may not. This can also be true for the profiles modeling different types of customers.

Various metrics can be derived from the QSTS simulation results: controller actions, voltage extremes, losses, etc. The accuracy of each is affected differently by the vector quantization. Depending on which metric a simulation should report, the vector quantization might be different and will vary for each specific feeder. Thus, a vector quantization strategy should be used to minimize the introduced error while providing an attractive computational time reduction. This topic is outside the scope of this paper but will be explored in future research.

VI. Conclusion

In this paper, the vector quantization algorithm proposed in [1] is adapted to model realistic size, complex feeders. The indexing method proposed can simulate large feeders with multiple profiles and controllable elements, a shortcoming that the previous algorithm could not do. Furthermore, this paper discussed the robustness of the algorithm capable of simulating feeders with various types of controllable elements. The new algorithm is tested with a model of an actual distribution feeder that includes 2969 buses, 3 profiles, and 8 controllable elements. This algorithm provides an opportunity for computational time reduction of QSTS simulations through vector quantization.

The capability to run a realistic distribution system feeder in a 93-99% reduction of the time on a conventional computer eliminates the main limitation to the wide use of QSTS simulation by the industry. Being able to simulate the operation of an actual distribution feeder with controllable elements on them can provide great insights for system planners especially as DER penetration increases. On the other hand, QSTS simulations without their computational burden could become valuable for other applications such as hosting capacity analysis or real time operation.

With an algorithm that can model a wide variety of feeder complexities, future work includes formulating a vector quantization strategy capable of optimally reducing the computational time with minimal losses in accuracy. In addition, this fast time-series approximation algorithm will ultimately be implemented in software such as CYME, OpenDSS, or GridLab-D.

VII. Acknowledgement

This research was supported by the DOE SunShot Initiative, under agreement 30691. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and 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.

VIII. References

[1] 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," *Sol. Energy*, p. Under Revision.
[2] IEEE P1547.7 D110, "Draft Guide to Conducting Distribution Impact Studies for Distributed Resource Interconnection," 2013.
[3] M. J. Reno, J. Deboever, and B. A. Mather, "Motivation and Requirements for Quasi-Static Time Series (QSTS) for Distribution System Analysis," in *IEEE PES General Meeting (2017)*.
[4] J. Deboever, X. Zhang, M. J. Reno, R. J. Broderick, and S. Grijalva, "Challenges in reducing computational time of QSTS simulations for distribution system analysis," SAND2017-5743, Albuquerque, NM, 2017.
[5] 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.
[6] M. J. Reno, K. Coogan, R. Broderick, and S. Grijalva, "Reduction of distribution feeders for simplified PV impact studies," in *IEEE Photovoltaic Specialists Conference*, 2013, pp. 2337–2342.
[7] D. Montenegro, G. A. Ramos, and S. Bacha, "A-Diakoptics for the Multicore Sequential-Time Simulation of Microgrids Within Large Distribution Systems," *IEEE Trans. Smart Grid*, pp. 1–9, 2015.
[8] C. D. López, "Thesis: Shortening time-series power flow simulations for cost-benefit analysis of LV network operation with PV feed-in," Uppsala Universitet, 2015.
[9] A. Pagnetti and G. Delille, "A simple and efficient method for fast analysis of renewable generation connection to active distribution networks," *Electr. Power Syst. Res.*, vol. 125, pp. 133–140, 2015.
[10] M. J. Reno, M. Lave, J. E. Quiroz, and R. J. Broderick, "PV Ramp Rate Smoothing Using Energy Storage to Mititgate Increased Voltage regulator Tapping," *IEEE Photovolt. Spec. Conf.*, 2016.
[11] J. Seuss, M. J. Reno, R. J. Broderick, and S. Grijalva, "Analysis of PV Advanced Inverter Functions and Setpoints under Time Series Simulation," SAND2016-4856, Albuquerque, NM, 2016.