Efficient Large-Scale Simulation of Integrated Transmission and Distribution Systems

Mohammad Asif Iqbal Khan, *Student Member, IEEE,* Sumit Paudyal, *Member, IEEE,* Sukumar Kamalasadan, *Senior Member, IEEE*

***Abstract*—The changing paradigm of power systems imposes emerging challenges for power system design, planning, and operation. The paradigm shift of power systems has been reflected in modeling and simulation methods used for power system analysis as well. Simulation frameworks are urged to continuously follow growing demand for large-scale system simulation, simulation of fast system dynamics, and high-fidelity models. There is a tendency toward performing joint simulations of systems that have been studied separately in the past. For instance, a joint simulation including detailed models of transmission and distribution systems is performed nowadays to analyze interdependencies and unforeseen interactions between integrated systems, particularly in case of scenarios with high level of penetration of distributed generation (DG). Additionally, envisioned complexity of future power systems requires simulation frameworks that enable interdisciplinary and multi-domain studies. Multi-physics simulation the approach is utilized for the holistic analysis of the city district energy systems that represent hybrid energy systems. Studies on architectures in smart grids require a simulation framework that enables joint analysis of power systems and information and communication technology systems. To maintain secure operations of transmission and distribution systems, it is important to have a better understanding of the coupling of transmission and distribution systems.**

***Index Terms*—Transmission system, distribution system, power flow, graphic processing unit (GPU), constant jacobian method.**

I. Introduction

THE modernization of the future grid includes bi-directional power flow in the distribution grids, which makes the steady-state interaction of distribution system (DS) and transmission system (TS) of utmost importance for the reliable operations of interconnected power grids [1]–[3]. Although the impact of variable energy resources at the distribution system have been studied in the recent works by the authors in [4], [5]; the impact of such integration at the bulk power system needs to be studied together with transmission system [6]. Additionally, the penetration of intermittent sources motivates us to develop computationally efficient power flow simulation tools that integrate grids at different voltage levels for real time analysis at finer time resolution using low-cost infrastructures [7], [8].

The transmission and distribution (T&D) model and solution methods in literature follow decoupled approaches, where TS and DS are decoupled at interface buses and solved independently using off-the-shelf tools. The work in literature lacks comprehensive component modeling (particularly on the distribution side), scalability, and real-time performance of integrated T&D systems for steady-state interactions in TS-DS.

This work is supported in part by National Science Foundation grant ECCS-2001732. M. A. I. Khan and S. Paudyal are with Florida International University, Miami, FL, USA; S. Kamalasadan is with the University of North Carolina at Charlotte; Emails: {mkhan103, spaudyal}@fiu.edu, skamalas@uncc.edu; Corresponding Author: M. A. I. Khan, mkhan103@fiu.edu

As the solve time is critical for real time performance, the development of computationally efficient models and methods are crucial to simulate large-scale integrated T&D systems. Additionally, integrated T&D simulation has scalability concerns, and hence, a working example of large-scale integrated T&D simulation is largely missing in the literature. Since the simulation of large-scale power systems consists of solving thousands of non-linear equations, which is computationally burdensome, graphics processing units (GPU) are used in literature to enhance the computational efficiency as GPUs can overcome the limitation of LU-based direct solvers by employing parallelization [9]–[11]. By exploiting the sparse vector/matrix operations, the massively parallel architecture of GPUs are utilized in a hybrid GPUâcentral processing unit (CPU) computing environment in [12], [13] to further reduce the solve time. GPU can aid in achieving additional layer of parallelism with better coalesced memory accesses in solving massive number of tasks in achieving higher speed up in simulation compared to traditional platforms [14], [15]. Additionally, use of pre-conditioner have been found effective for a GPU based platform [16]. Heterogeneous CPU-GPU hybrid platform with generalized minimum residual (GMRES), bi-conjugate gradient based sequential, and parallelly iterative power flow solver can provide better parallelism, simulation efficiency and convergence as demonstrated in [17], [18].

The power flow solution algorithm is critical for speed up and its realization in hardware as it requires parallelizing capability to implement on GPU platforms. Forward-Backward algorithm shows promising results with possibility of parallelizing for radial feeders [19]. Newton-Raphson (NR) based method shows greater speed-up in GPU compared to other methods for transmission system, while the absolute solve time of NR-based method still would be slow as system size grows [20] and a combination of constant jacobian approach with traditional NR in presence of adaptive jacobian update is proven to be efficient and accurate for transmission and distribution system separately [21]–[23]. It is shown in [24], using multi-frontal method that the efficiency of solving sparse linear equations of NR method can be improved by using GPU based linear algebra library. GPUs parallel programming library coupled with efficient power flow solving algorithm has found to provide real time analysis of large-scale power systems and at least two orders of magnitude faster than the CPU counterpart [25], [26].

Though the aforementioned works demonstrate promising results for fast power flow computation with the aid of GPUs, no work in the literature has demonstrated performance in solving power flow solutions for scalable integrated T&D systems using the GPU platform. This works intends to extend the ideas to solve an integrated T&D simulation in CPU-GPU platform for real-time TS-DS interactions.

The remainder of the paper is structured as following.

Section II discusses the power flow solution methods for transmission and distribution systems. Section III discusses the basic GPU architecture and Decoupled Approach with CPU-GPU Implementation. Section IV presents and discusses the case studies and simulation results. Section V discusses the main conclusions drawn from this work.

II. Power Flow Solution Methods

In this Section, we briefly explain the two power flow solution algorithms used for the transmission and distribution systems in addition to development of power flow models, decoupled approach, and the comprehensive component models. There are derivative-free methods to solve distribution system power flow, which are demonstrated to be efficient for large-scale systems and are suitable for GPU implementation. However, the main computational bottleneck would be from the power flow of transmission side, which we overcome using Constant Jacobian Method.

A. Decoupled Approach for solving Integrated T&D Model

Consider an N-bus transmission system, where the first m buses have lumped loads, and buses $m+1$ through N have distribution feeders connected downstream the buses. In a decoupled approach, power flow model of the transmission system is solved in sequence frame. Consider a R-node distribution system connected to an arbitrary j-bus of the transmission network. Distribution power flow analysis can be formulated similar to the transmission system, except the distribution systems are modeled in phase frame of reference. The solution procedure of mathematical models of the decoupled approach can be explained using Fig. 1. In the Fig. 1, an integrated T&D system is shown where distribution feeders are connected at nodes i and j. In the decoupled approach, the transmission and distribution systems are solved iteratively and based on some information exchange (Please see [6] for details on the mathematical modeling).

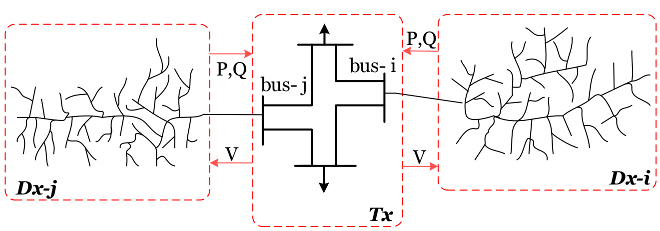

Fig. 1. A T&D system showing boundaries and information exchange for solving transmission and distribution integrated simulation using a decoupled approach

B. Constant Jacobian Method

Being a derivative based method that requires minimal effort on the initial guess and excellent convergence for transmission level power flow, NR is typically used over others [27], [28]. However, NR is slow for large systems. In regular NR method, the power flow solution is obtained by solving iteratively the following set of equations with mismatch vector $[\Delta x]$, Jacobian matrix $[J]$, and unknown vector $[\Delta y]$.

$$\underbrace{\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix}_k}_{[\Delta x]} = \underbrace{\begin{bmatrix} J_1 & J_2 \\ J_3 & J_4 \end{bmatrix}_k}_{[J]} \underbrace{\begin{bmatrix} \Delta \delta \\ \Delta V \end{bmatrix}_k}_{[\Delta y]} \tag{1}$$

where $J_1 = \frac{\partial P}{\partial \delta}$, $J_2 = \frac{\partial P}{\partial V}$, $J_3 = \frac{\partial Q}{\partial \delta}$, $J_4 = \frac{\partial Q}{\partial V}$. P, Q, V, δ are the active, reactive power, voltage magnitude and angle respectively. The k-th iteration of NR update equations can be written as,

$$[x_{k+1}] = [x_k] + [\Delta x_k], \tag{2}$$
$$[\Delta x_k] = [J_k]^{-1} [\Delta y_k]. \tag{3}$$

Computational burden of NR depends on the size of Jacobian sub-matrices $[J1]$, $[J2]$, $[J3]$, $[J4]$ in (1) and factorization of the full Jacobian matrix at every iteration. For large system (thousands of buses) the burden is huge as the weight of the matrix increases in a cubic rate of matrix size N. Though the factorization operation of the matrix can be reduced to $N^{1.5}$ from N^3 using sparse matrices techniques, still around 85% time and computational resources are consumed by these factorization and remains the most critical issue in solving large-scale power systems with NR method [29]. Therefore, in this work, Constant Jacobian method is used to reduce the computational burden and to achieve faster solve time. In the Constant Jacobian method, the Jacobian matrix is pre-calculated and kept constant throughout the iterative process, which is proven to be fairly accurate, and computationally less burdensome. This also makes the GPU implementation much simpler as avoid large matrix inversion every iteration. The update equation can be written as,

$$[x_{k+1}] = [x_k] + [\Delta x_k], \tag{4}$$
$$[\Delta x_k] = [\bar{J}]^{-1} [\Delta y_k], \tag{5}$$

where $[\bar{J}]$ is the Jacobian matrix which is kept constant through the NR update process, and is equal to the Jacobian matrix evaluate at the initial starting solution of NR.

C. Z-Bus Method

On the distribution system side, derivative free methods are available. We adopt Implicit Z-bus method, which is derivative free, and hence, simple to implement and has shown efficient computational performance [30]. Implicit Z-bus methods involve the solution of a linear set of equations for constant current injections and an iterative method for representing non-linear constant power load models [30].

The basic of Implicit Z-bus method follows ohm's law in a circuit and in the three-phase frame can be represented as,

$$\underbrace{\begin{bmatrix} i_n \\ i_s \end{bmatrix}}_{[i]} = \underbrace{\begin{bmatrix} Y_{NN} & Y_{NS} \\ Y_{SN} & Y_{SS} \end{bmatrix}}_{[Y]} \underbrace{\begin{bmatrix} v_n \\ v_s \end{bmatrix}}_{[v]} \tag{6}$$

where $[Y]$ is the network admittance that consists of four sub-matrices of Y_{NN}, Y_{SN}, Y_{NS} and Y_{SS}. Subscript S represents slack buses and N represents non-slack buses. $[i]$ is the nodal current injection vector, and $[v]$ is the nodal voltage vector. The voltage update equation of Z-bus is given as,

$$v_n = {Y_{NN}}^{-1} i_n - {Y_{NN}}^{-1} Y_{NS}\ v_s. \tag{7}$$

III. Simulation Tool, GPU and Decoupled Approach with CPU-GPU Implementation

In this section, discuss the basics of the GPU architecture and the GPU language CUDA. Additionally, the major flow of the power flow method for the integrated T&D system in the hybrid CPU-GPU environment is discussed.

A. GPU Architecture and CUDA

In this work, we are using a NVIDIA GPU and the algorithm is based on Compute Unified Device Architecture (CUDA) based application program interfaces (API) which is a simpler device management when using NVIDIA GPUs [31]–[33]. CUDA is a parallel computing platform and programming model that follows data-parallel model of computation. In the heterogeneous CPU-GPU hybrid CUDA computing environment, the CPU works as a host, processing the sequential (non-iterative) tasks due to its hierarchy in control logic and memory power, and GPU provides the acceleration through accessing the global memory and parallelizing the assigned tasks into multiple SMs through simple logical functions called the kernel which can work in iterative manner [20]. The typical execution model of a CUDA program is shown in Fig. 2 where CPU, the host, initiates the simulation process and transfers the calculation instruction and parameters to the GPU by initiating kernel calls which in turn allocates memory thread block containing grid of threads to perform the large stream of calculation using parallel functions [13].

B. CPU-GPU Hybrid Implementation

The developed models and algorithms are implemented on CPU and CPU-GPU hybrid computing environment. A flow chart showing the simulation process and data flow between the CPU-GPU hybrid simulation architecture is presented in Fig. 3. The CPU does the prepossessing and initial non-iterative steps of power flow process. That is using the available system data, Y-bus of transmission and distribution systems are formulated in CPU. The initialization of unknown vectors and initial Jacobain for TS are also evaluated in CPU. Then, the data is transferred from CPU to GPU. In the GPU, the iterative steps of power flow calculations are carried out using designated functions for TS and DS. Once the TS power flow converges, the DS sub-station voltage is updated and the DS function operates till it converges. Once, the DS function converges, the TS system load data is updated using the DS power flow and this process continues until the global integrated T&D mismatch is lower than specified tolerance and then, the final Power Flow results are transferred back to CPU from GPU for visualization.

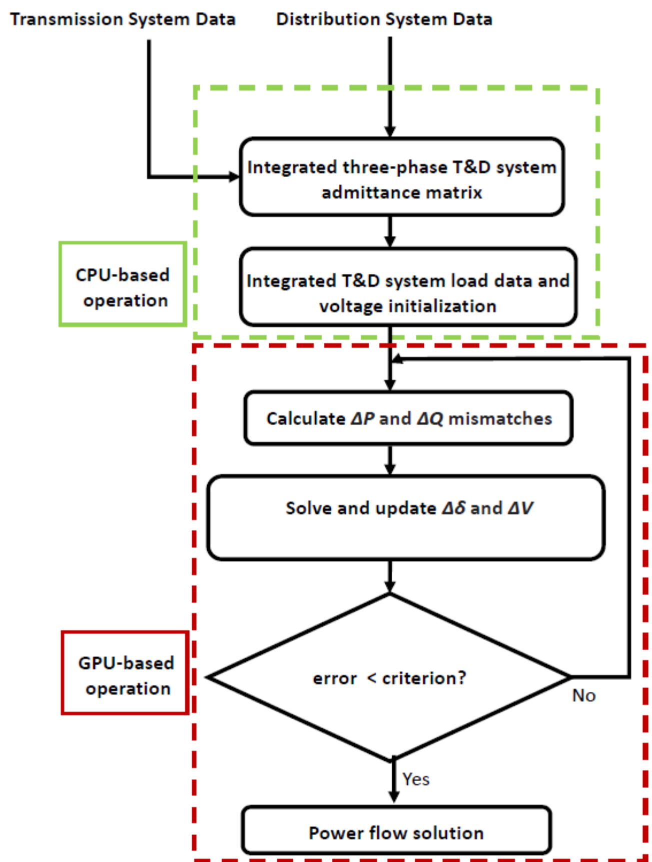

Fig. 3. Flowchart of Unified T&D Simulation in hybrid CPU-GPU environment

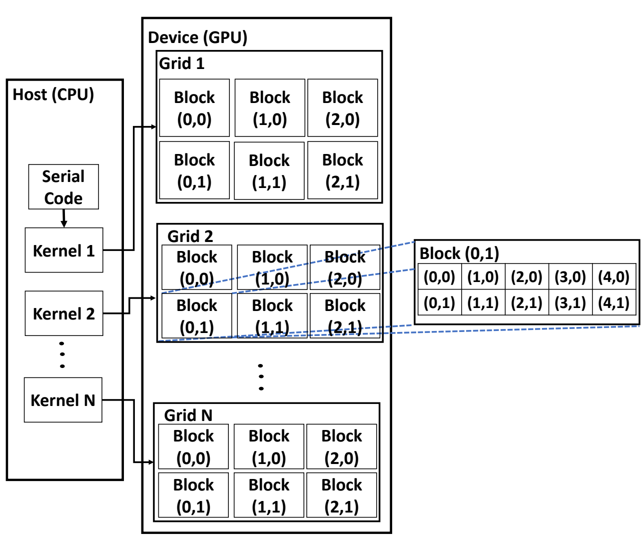

Fig. 2. Schematic illustration of CUDA programming elements.

C. Evaluation Metrics

We adopt the following metrics to measure and evaluate the performance of the CPU and CPU-GPU based solvers.

- Solve Time (t): The elapsed time in solving the cases in CPU and CPU-GPU hybrid simulation are recording the timestamp using *cudaEventRecord()* function that returns the elapsed time between the *start* and *stop* of the kernel calls in seconds with a resolution of half a microsecond [34].
- Effective Bandwidth: The effective bandwidth is calculated using the recorded timing and accessing the details of programming data structure. For the integrated T&D system, the effective bandwidth is calculated using the formula below and is represented in units of GB/s.

$$BW_{tnd} = \frac{(I_B + O_B) \times iter}{t} \tag{8}$$

where I_B and O_B are the input and output data to and from the GPU and $iter$ is the number of computational iterations.

IV. Numerical Studies

The simulation models are run on CPU and GPU using Microsoft Visual Studio Integrated development environment

(IDE) (version 2017 community). The CPU operating environment comprises 64 bit 8th Generation IntelÂ® Coreâ¢ i7 6 core and 12 threads (6C12T) Processors running at 2.20 GHz, and Bus Speed of 8 GT/s. The maximum memory size is 64 GB in a maximum of 2 channels with a maximum memory bandwidth of 41.8 GB/s. The GPU is an NVIDIA GeForce GTX 1050 Ti (4 GB, DDR5) with 768 CUDA cores with a Processor clock of 1.392 GHz, a Graphics Clock of 1.290 GHz, and a Memory Bandwidth of 112 GB/s connected through a 128-bit wide memory PCIe. We have tested the efficacy of the proposed architecture in the test system ranging from small to large scale integrated T&D systems presented below.

A. Case 1: Small Test System

The models, solution algorithms, and hybrid CPU-GPU implementation are first tested on a small system that comprises of the IEEE 14 bus Transmission System (TS) and 123-bus Distribution System (DS) as shown in Fig. 4 referred as the 14-123 bus system. The IEEE 14-bus test system comprises 14 buses and 20 lines hosting 5 generators, 3 transformers, 4 capacitor banks, and 11 loads with a net load of 260 MW (75 MVAr) which altogether is an ideal and simple approximation of the American Electric Power system. The IEEE 123-node test system operates at a nominal voltage of 4.16 kV and is characterized by three-phase overhead and underground lines, unbalanced loading with constant current, impedance, and power, four voltage regulators, shunt capacitor banks, and multiple switches with a net-connected load of 3.7 MW (1.3 MVAr). First, the transmission system and distribution system power flow convergences are independently ensured. Then, the coupled T&D model is solved using a CPU and a hybrid CPU-GPU platform using the decoupled approach. Fig. 5 shows the node voltage magnitudes and angles obtained from the simulation. The comparison of the same solution to its counterpart from CPU reveals very minimal error as seen from Fig. 6 and shows to outperform CPU based approach.

B. Case 2: Large Test System

A large sized system is created by combining 2383-bus Polish transmission system (TS) and 2522 nodes distribution system (DS) that resulted in total of 6,200 nodes. The Polish network contains in total of 327 synchronous generators. The generator buses comprise 322 shunt loads and there are 1,503 active and reactive loads measuring a net-connected load of 24,558 MW (8,144 MVAr). For the large-scale distribution case studies, a 2522-bus unbalanced distribution system is used, which is extracted from the MV-side of the IEEE 8500-node test feeder [35]. The test system has total of 3,817 single-phase nodes among which 1,413 are a load nodes. The IEEE standard load data for this feeder is modified and a load of 6.32 MW and PV of 6 MW are connected. The coupled T&D system is shown in Fig. 7 and is referred as 2383-2522 bus system.

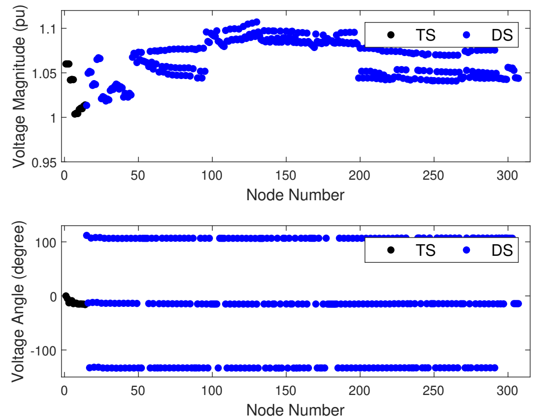

Fig. 5. Voltage magnitude and angle of the integrated small scale T&D system (14 bus transmission and 123 bus distribution system).

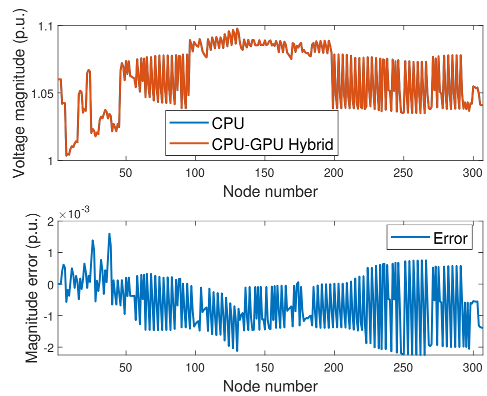

Fig. 6. Error in Voltage magnitude comparison between CPU and CPU-GPU solutions

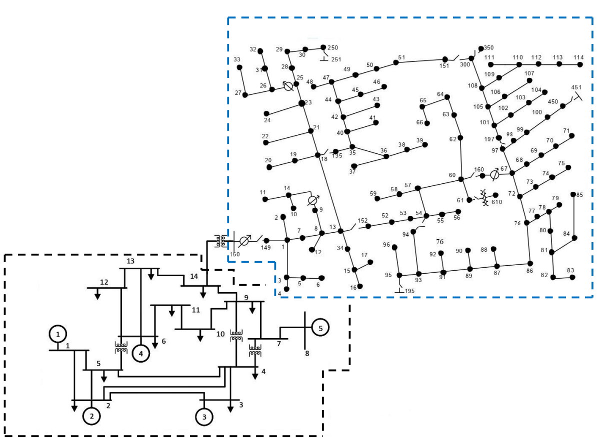

Fig. 4. Schematic of integrated T&D system (14 bus transmission and 123 bus distribution system).

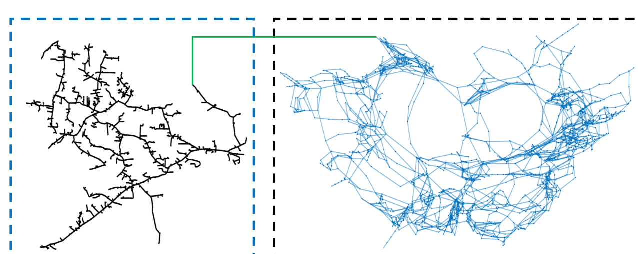

Fig. 7. Topology of the integrated large scale T&D system (2383 bus transmission and 2522 bus distribution) .

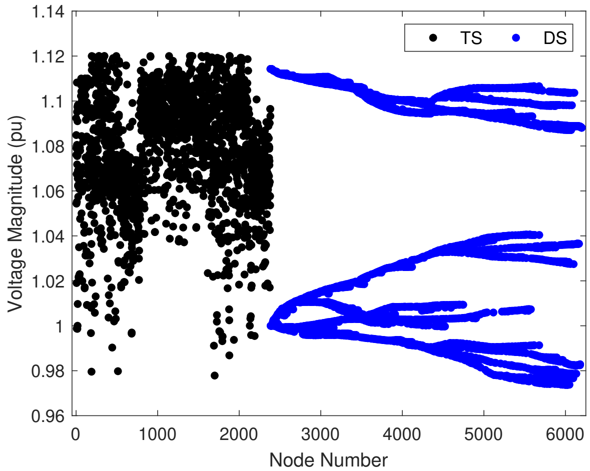

Fig. 8. Voltage magnitude of the integrated T&D system on the large scale system with 2383 bus transmission and 2522 bus distribution.

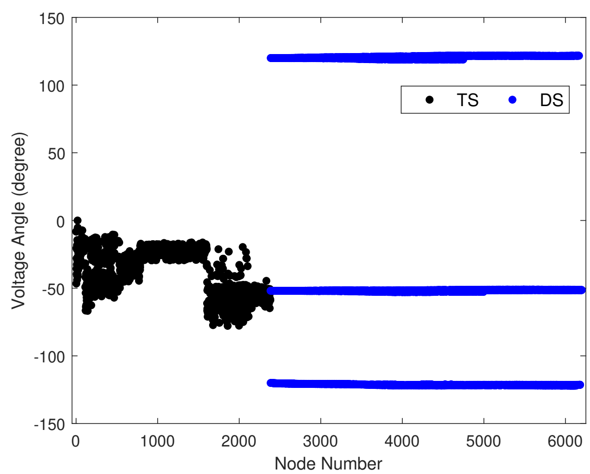

Fig. 9. Voltage angle of the integrated T&D system on the large scale system with 2383 bus transmission and 2522 bus distribution.

The resulting 6200-node system (Fig. 7) is then solved in a CPU-based and CPU-GPU hybrid platform to compare the performance of voltage magnitude and angle solution on the transmission as well as distribution part of the circuit. The load flow voltage magnitude and angle obtained from the CPU-GPU hybrid platform are shown in Fig. 8 and Fig. 9, respectively.

C. Comparative Analyses

Next, we analyze the solution accuracy, solve time, and bandwidth usage of CPU-GPU platform. The solve time of the simulations from both CPU and CPU-GPU hybrid environment are recorded and compared in Table I. For the small test system (i.e., the integrated system of 14-123 bus shows minimal improvement in the hybrid environment that provides only 1.15 times faster solution compared to the CPU version solve time of 3.40 s to 2.96 s by the hybrid solver. However, the computing performance of the hybrid environment are significant as the network size grows larger. For the large scale system with 6,200 nodes, the hybrid solver provides 6.67 faster solution in 114.7 s compared to 765.57 s by the CPU based approach.

TABLE I
BANDWIDTH AND SPEED-UP OF CPU-GPU PLATFORM

	14-123	**2383-2522**
Data (MB)	2.31	1,521.60
CPU Sim. Time (s)	3.41	765.57
CPU-GPU Sim.Time (s)	2.96	114.70
Speed-Up	**1.15**	**6.67**
BW (GB/s)	**0.04**	**1.02**

V. CONCLUSION

With distribution grids becoming active systems, TSO-DSO in modernized power grid requires efficient models and solution methods for steady-state interactions of large-scale integrated T&D system. Hence, a need arises of a comprehensive mathematical model (including three-phase circuit components of distribution feeders) to study T&D interactions. In this work, we formulated and tested a detailed model of transmission and distribution systems overcoming the shortcoming of existing models in the literature while showing promising outcomes in the scalability. Using the computational advantages of Constant Jacobian method on the transmission side and derivative free Z-bus method on the distribution side, the developed comprehensive integrated model that is been proven to be efficient for T&D system. Furthermore, the model is implemented on a low-cost CPU-GPU hybrid simulation environment in achieving faster and memory efficient performance. From the simulation results and analyses, we have found that CPU-GPU based implementation outperforms CPU based solver.

.

REFERENCES

[1] Q. Huang and V. Vittal, "Integrated transmission and distribution system power flow and dynamic simulation using mixed three-sequence/three-phase modeling," *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3704–3714, Sep. 2017.

[2] R. Huang, R. Fan, J. Daily, A. Fisher, and J. Fuller, "Open-source framework for power system transmission and distribution dynamics co-simulation," *IET Generation, Transmission Distribution*, vol. 11, no. 12, pp. 3152–3162, 2017.

[3] K. Balasubramaniam and S. Abhyankar, "A combined transmission and distribution system co-simulation framework for assessing the impact of volt/var control on transmission system," in *Proc. IEEE Power Energy Society General Meeting*, July 2017, pp. 1–5.

[4] M. A. I. Khan, S. Paudyal, and M. Almassalkhi, "Performance evaluation of network-admissible demand dispatch in multi-phase distribution grids."

[5] F. B. dos Reis, K. Duwadi, R. Foutney, R. Tonkoski, T. M. Hansen, M. A. I. Khan, and S. Paudyal, "Impact of residential load models for overvoltage prevention studies in pv-rich lv grids," in *2019 IEEE Milan PowerTech*, 2019, pp. 1–6.

[6] M. A. I. Khan, A. Suresh, S. Paudyal, and S. Kamalasadan, "Decoupled and unified approaches for solving transmission and distribution co-simulations," in *Proc. North American Power Symposium (NAPS)*, 2019, pp. 1–6.

[7] M. J. Reno, J. Deboever, and B. Mather, "Motivation and requirements for quasi-static time series (QSTS) for distribution system analysis," in *Proc. IEEE Power Energy Society General Meeting*, 2017, pp. 1–5.

[8] D. F. Rodriguez, D. L. Alvarez, D. Gomez, J. M. Gers, and S. Rivera, "Low-cost analysis of load flow computing using embedded computer empowered by gpu," in *2021 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2021, pp. 1–5.

[9] X. Li, F. Li, H. Yuan, H. Cui, and Q. Hu, "GPU-based fast decoupled power flow with preconditioned iterative solver and inexact newton method," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2695–2703, July 2017.

[10] R. Gnanavignesh and U. J. Shenoy, "Parallel sparse lu factorization of power flow jacobian using gpu," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, 2019, pp. 1857–1862.

[11] A. D.Vasquez and T. Sousa, "A parallel processing approach to stability analysis considering transmission and distribution systems," in *2019 IEEE Milan PowerTech*, 2019, pp. 1–6.

[12] T. Huang, Q. Guo, and H. Sun, "A distributed computing platform supporting power system security knowledge discovery based on online simulation," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1513–1524, May 2017.

[13] V. Roberge, M. Tarbouchi, and F. Okou, "Parallel power flow on graphics processing units for concurrent evaluation of many networks," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1639–1648, July 2017.

[14] G. Zhou, Y. Feng, R. Bo, L. Chien, X. Zhang, Y. Lang, Y. Jia, and Z. Chen, "Gpu-accelerated batch-acpf solution for n-1 static security analysis," *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1406–1416, May 2017.

[15] G. Zhou, R. Bo, L. Chien, X. Zhang, F. Shi, C. Xu, and Y. Feng, "Gpu-based batch lu-factorization solver for concurrent analysis of massive power flows," *IEEE Transactions on Power Systems*, vol. 32, no. 6, pp. 4975–4977, Nov 2017.

[16] X. Li and F. Li, "Gpu-based two-step preconditioning for conjugate gradient method in power flow," in *Proc. IEEE Power Energy Society General Meeting*, July 2015, pp. 1–5.

[17] X. Liu, H. Wang, and S. X. . Tan, "Parallel power grid analysis using preconditioned gmres solver on CPU-GPU platforms," in *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2013, pp. 561–568.

[18] N. Garcia, "Parallel power flow solutions using a biconjugate gradient algorithm and a newton method: A GPU-based approach," in *Proc. IEEE PES General Meeting*, July 2010, pp. 1–4.

[19] D. Ablakovic, I. Dzafic, and S. Kecici, "Parallelization of radial three-phase distribution power flow using GPU," in *Proc. 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, Oct 2012, pp. 1–7.

[20] C. Guo, B. Jiang, H. Yuan, Z. Yang, L. Wang, and S. Ren, "Performance comparisons of parallel power flow solvers on GPU system," in *Proc. IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, Aug 2012, pp. 232–239.

[21] A. Semlyen and F. de Leon, "Quasi-newton power flow using partial jacobian updates," *IEEE Transactions on Power Systems*, vol. 16, no. 3, pp. 332–339, 2001.

[22] S. Wang, G. Liu, J. Xu, Y. Lang, Z. Yang, and C. Dou, "Power flow calculation of three-phase distribution network based on constant jacobian matrix and newton method," in *2014 China International Conference on Electricity Distribution (CICED)*, 2014, pp. 381–384.

[23] L. Zeng, S. G. Alawneh, and S. A. Arefifar, "Gpu-based sparse power flow studies with modified newtonâs method," *IEEE Access*, vol. 9, pp. 153 226–153 239, 2021.

[24] Xue Li, Fangxing Li, and J. M. Clark, "Exploration of multifrontal method with gpu in power flow computation," in *Proc. IEEE Power Energy Society General Meeting*, July 2013, pp. 1–5.

[25] Z. Wang, A. Chakrabortty, C. T. Kelley, X. Feng, and P. Franzon, "Improved numerical methodologies on power system dynamic simulation using gpu implementation," in *Proc. IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Feb 2019, pp. 1–5.

[26] G. Zhou, R. Bo, L. Chien, X. Zhang, S. Yang, and D. Su, "GPU-accelerated algorithm for on-line probabilistic power flow," in *Proc. IEEE Power Energy Society General Meeting (PESGM)*, Aug 2018, pp. 1–1.

[27] K. Birt, J. Graffy, J. McDonald, and A. El-Abiad, "Three phase load flow program," *IEEE Transactions on Power Apparatus and Systems*, vol. 95, no. 1, pp. 59–65, 1976.

[28] M. Dai, M. N. Marwali, J.-W. Jung, and A. Keyhani, "Power flow control of a single distributed generation unit," *IEEE Transactions on Power Electronics*, vol. 23, no. 1, pp. 343–352, 2008.

[29] F. Milano, *Power Flow Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 61–101.

[30] T. Chen, M. Chen, K. Hwang, P. Kotas, and E. A. Chebli, "Distribution system power flow analysis-a rigid approach," *IEEE Transactions on Power Delivery*, vol. 6, no. 3, pp. 1146–1152, July 1991.

[31] J. Fang, A. L. Varbanescu, and H. Sips, "A comprehensive performance comparison of CUDA and OpenCL," in *Proc. International Conference on Parallel Processing*, 2011, pp. 216–225.

[32] C.-L. Su, P.-Y. Chen, C.-C. Lan, L.-S. Huang, and K.-H. Wu, "Overview and comparison of OpenCL and CUDA technology for GPGPU," in *Proc. IEEE Asia Pacific Conference on Circuits and Systems*, 2012, pp. 448–451.

[33] D. B. Kirk and W.-m. W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, ser. Applications of GPU Computing Series. Burlington, MA: Morgan Kaufmann Publishers, 2010.

[34] *Professional CUDA C Programming*, 1st ed. GBR: Wrox Press Ltd., 2014.

[35] K. P. Schneider, B. A. Mather, B. C. Pal, C.-W. Ten, G. J. Shirek, H. Zhu, J. C. Fuller, J. L. R. Pereira, L. F. Ochoa, L. R. de Araujo, R. C. Dugan, S. Matthias, S. Paudyal, T. E. McDermott, and W. Kersting, "Analytic considerations and design basis for the ieee distribution test feeders," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3181–3188, 2018.