On Detailed Synchronous Generator Modeling for Massively Parallel Dynamic State Estimation

Hadis Karimipour Venkata Dinavahi
Department of Electrical & Computer Engineering
University of Alberta
Edmonton, Alberta T6G 2V4, Canada.
Email: hadis@ualberta.ca, dinavahi@ualberta.ca

Abstract— **Synchronous generators are normally represented in a simplified fashion to reduce computational complexity in dynamic state estimation (DSE). In this paper a dynamic state estimator for a sixth-order synchronous generator model was developed on the massively parallel graphic processing units (GPU) to provide detailed and accurate Extended Kalman Filter (EKF) based estimation of the generator states. The estimation results are compared with the time domain simulation results on the CPU to demonstrate the accuracy of the proposed method. Also a speed-up of 10.02 for a 5120 generator system is reported.**

Index Terms— **Dynamic state estimation, extended kalman filter, graphic processing units, large-scale systems, parallel programming, synchronous generator model.**

I. Introduction

Computational speed is a major concern for online dynamic state estimation (DSE) of power systems. Traditional state estimators overcame this problem by using simplified network modeling [1], [2], and by reducing the size of the problem [3]–[5] whose state was being estimated. Some approaches also focused on estimation accuracy by increasing either modeling or algorithmic complexity [6]–[8]. However, these methods were computationally onerous limiting their practical applicability to small scale systems. With widespread use of phasor measurement units (PMUs), the computational demands on DSE have increased. Accurate component modeling and large system sizes have become important challenges that need to be addressed.

Synchronous generator modeling in large-scale DSE needs particular attention. Detailed representation of the synchronous generator in online DSE would allow system operators to accurately assess system condition and take rapid control actions following major disturbances. There is limited research on the state estimation of the synchronous generator. For example, a dynamic state estimation method for second-order synchronous machine model, neglecting the field voltage dynamic equations, was proposed in [9]. An unscented Kalman filter based state estimation method for third-order generator model assuming the rotor angle as a measurable signal is represented in [10]. Also proposed in [6] is DSE with unknown inputs for fourth-order generator model considering the exciter output voltage as unknown input. Using the same generator model, [11] proposed an optimal state estimation of both generator internal dynamics and algebraic states. However, in these works the number of synchronous generators in the state estimation process was limited to a single generator which precluded the study of computational performance of large-scale systems.

The objective of this paper is to investigate the impact of detailed synchronous generator model on the accuracy and speed of state estimation using multi-core CPU and many-core GPU architecture for large-scale systems. An Extended Kalman Filter (EKF) based DSE utilizing graphic processing units (GPUs) is introduced where the synchronous machine is represented by a sixth-order dynamic model including the excitation system with automatic voltage regulator (AVR) and power system stabilizer (PSS) which increases the number of differential equations and hence the complexity of the model. The mathematical complexity and the substantial opportunity to exploit parallelism on large data sets has made such DSE uniquely suited for the GPU architecture. GPUs have been proven to be efficacious in large-scale electromagnetic transient and transient stability simulations [12], [13]. Fine grained parallelism on GPU by distributing tasks among individual compute unified device architecture (CUDA^{TM}) threads accelerated the synchronous generators state estimation in comparison to multi-thread CPU implementation. Case studies containing up to 5120 synchronous machines were utilized in this research, and the accuracy of the state estimation is verified by time domain simulation.

The organization of this paper is as follows. Section II describes generator modeling and the massively parallel DSE algorithm. Section III presents the experimental results and analysis, followed by conclusions in Section IV.

II. Extended Kalman Filter Based Parallel DSE

A. Synchronous Generator Model

In this work a sixth-order model of synchronous generator including AVR and PSS is used. The complete system representation is summarized here.

This work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

978-1-4799-5904-4/14/$31.00 ©2014 IEEE

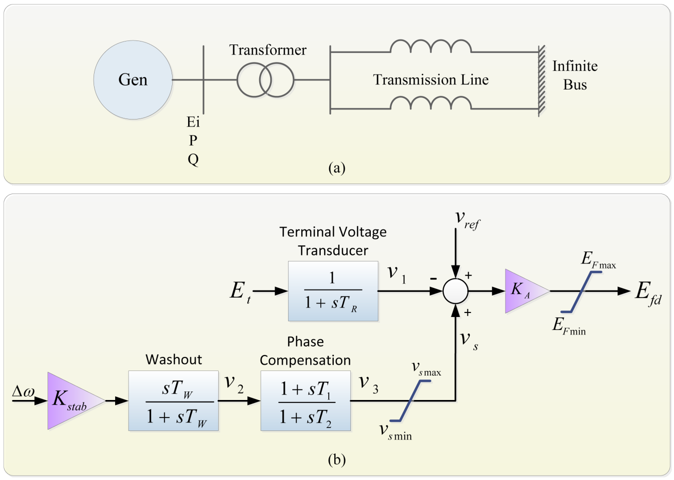

Fig. 1. Synchronous generator excitation system with AVR and PSS.

Equations of motion and field voltage:

$$\begin{aligned} \dot{\delta}(t) &= \omega_R.\Delta\omega(t), \\ \dot{\Delta\omega}(t) &= \frac{1}{2H}[T_e(t) + T_m - D.\Delta\omega(t)], \\ \dot{e}'_q(t) &= \frac{1}{T'_{do}}[E_{fd}(t) - e'_q(t) - (x_d - x'_d)I_d(t)]. \end{aligned} \tag{1}$$

where δ, ω and e'_q represent the rotor angle, rotor speed and direct axis transient voltage, respectively. T_e stands for the electrical torque, T_m is the mechanical power, E_{fd} is the exciter output voltage and I_d is the direct axis current. Excitation system including an AVR and PSS:

$$\begin{aligned} \dot{v}_1(t) &= \frac{1}{T_R}[v_t(t) - v_1(t)], \\ \dot{v}_2(t) &= K_{stab}.\dot{\Delta\omega}(t) - \frac{1}{T_w}v_2(t), \\ \dot{v}_3(t) &= \frac{1}{T_2}[T_1\dot{v}_2(t) + v_2(t) - v_3(t)]. \end{aligned} \tag{2}$$

where v_1, v_2 and v_3 are excitation voltages. v_t represents the terminal voltage. Fig. 1 (b) shows ST1A type excitation system [14]. $\omega_R, H, D, T'_{do}, x_d, x_q, x'_d, x'_q, T_R, T_w, T_1, T_2$ and K_{stab} are constant parameters whose definition can be found in [15]. Neglecting the stator resistance R_a, electrical torque T_e will be equal to terminal active power P_t.

$$T_e \cong P_t = e_d i_d + e_q i_q. \tag{3}$$

Using the definition of d-axis and q-axis voltages (e_d, e_q) and currents (i_d, i_q) from [15], electrical torque can be expressed as:

$$T_e \cong \frac{v_t sin(\delta)}{x'_d}e'_q + \frac{v_t(x'_d - x_q)}{x'_d x_q}sin(\delta)cos(\delta). \tag{4}$$

B. Extended Kalman Filter Based Estimation

The sixth order state-space model can be rewritten as:

$$\begin{aligned} \dot{x}_1 &= \omega_R x_2, \\ \dot{x}_2 &= \frac{1}{2H}[T_e + T_m - Dx_2], \\ \dot{x}_3 &= \frac{1}{T'_{do}}[E_{fd} - \frac{x_d}{x'_d}x_3 + (\frac{x_d}{x'_d} - 1)v_t cos(x_1)], \\ \dot{x}_4 &= \frac{1}{T_R}[v_t - x_4], \\ \dot{x}_5 &= K_{stab}\dot{x}_2 - \frac{1}{T_w}x_5, \\ \dot{x}_6 &= \frac{1}{T_2}[T_1\dot{x}_5 + x_5 - x_6]. \end{aligned} \tag{5}$$

The measurement set M includes the electrical output power given as:

$$M = \frac{v_t sin(x_1)}{x'_d}x_3 + \frac{v_t(x'_d - x_q)}{x'_d x_q}sin(x_1)cos(x_1), \tag{6}$$

According to the aforementioned formulations the vector of state variables $\boldsymbol{x}$ for the synchronous generator is given as:

$$\boldsymbol{x} = [\delta, \Delta\omega, e'_q, v_1, v_2, v_3], \tag{7}$$

The general form of the dynamic model is represented as:

$$\begin{aligned} \dot{\boldsymbol{x}} &= f(\boldsymbol{x}, \Delta t, \boldsymbol{\upsilon}), \\ \boldsymbol{M} &= h(\boldsymbol{x}, \Delta t, \boldsymbol{\varepsilon}). \end{aligned} \tag{8}$$

where $\boldsymbol{\upsilon}$ and $\boldsymbol{\varepsilon}$ are system and measurements noises assuming normal distribution with zero mean, and 6×6 covariance $\boldsymbol{Q}$ and $\boldsymbol{R}$, respectively. The estimation time step is represented by Δt. The Trapezoidal integration method was used for discretization with respect to time. Then linearization of (8) resulted in a new set of algebraic equations:

$$\boldsymbol{x}_{(t+\Delta t)} = \boldsymbol{F}_t\boldsymbol{x}_t + \boldsymbol{\upsilon}_t, \ \ \boldsymbol{\upsilon}_t \sim N(0, \boldsymbol{Q}_t), \tag{9}$$

where $\boldsymbol{F}_t = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}|_{x_{(t-\Delta t)}}$ represents the 6×6 state transition between two time steps. Under normal operating conditions it is possible to adjust $\boldsymbol{F}_t$ such that $\boldsymbol{Q}_t$ remains constant. There are two main steps in the implementation of EKF: state prediction and state filtering.

1) State Prediction: Using the measurement and estimated states at the time instant t, the predicted value $\tilde{\boldsymbol{x}}_{t+\Delta t}$ can be formulated as:

$$\begin{aligned} \tilde{\boldsymbol{x}}_{(t+\Delta t)} &= \boldsymbol{F}_t\hat{\boldsymbol{x}}_t, \ \ (\boldsymbol{x}_t - \hat{\boldsymbol{x}}_t) \sim N(0, \boldsymbol{\rho}_t), \\ \tilde{\boldsymbol{\rho}}_{(t+\Delta t)} &= \boldsymbol{F}_t\boldsymbol{\rho}_t{\boldsymbol{F}_t}^T + \boldsymbol{L}_t\boldsymbol{Q}_t{\boldsymbol{L}_t}^T, \ \ (\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t) \sim N(0, \tilde{\boldsymbol{\rho}}_t), \end{aligned} \tag{10}$$

where $\boldsymbol{L}_t = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\upsilon}}|_{x_{(t-\Delta t)}}$ is a 6×6 matrix of partial derivative of f with respect to system noise. $\boldsymbol{\rho}$ and $\tilde{\boldsymbol{\rho}}$ are 6×6 error covariance matrices for estimated and predicted values, respectively. $\tilde{\boldsymbol{x}}$ represent the predicted state and $\hat{\boldsymbol{x}}$ is the estimated state.

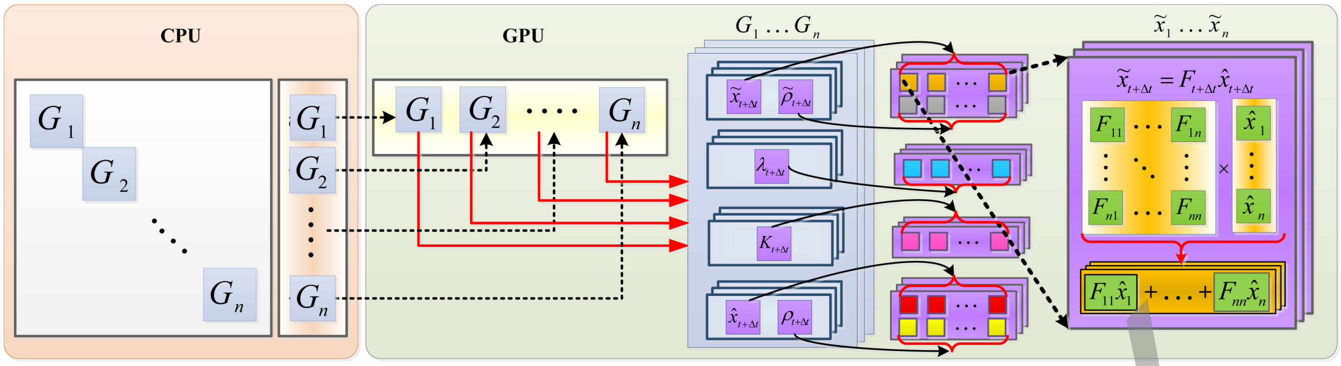

Fig. 2. GPU implementation of detailed generator DSE based on EKF.

2) State Filtering: This step updates the predicted values using the next set of measurements at the time instant $(t + \Delta t)$. The updated state through EKF can be written as:

$$\begin{aligned}
\hat{\boldsymbol{x}}_{(t+\Delta t)} &= \tilde{\boldsymbol{x}}_{(t+\Delta t)} + \boldsymbol{K}_{(t+\Delta t)}(\boldsymbol{M}_{(t+\Delta t)} - \boldsymbol{h}(\tilde{\boldsymbol{x}}_{(t+\Delta t)})),\\
\boldsymbol{K}_{(t+\Delta t)} &= \tilde{\boldsymbol{\rho}}_{(t+\Delta t)}\boldsymbol{H}^T{}_{(t+\Delta t)}\boldsymbol{\lambda}^{-1}{}_{(t+\Delta t)},\\
\boldsymbol{\lambda}_{(t+\Delta t)} &= [\boldsymbol{H}_{(t+\Delta t)}\tilde{\boldsymbol{\rho}}_{(t+\Delta t)}\boldsymbol{H}^T{}_{(t+\Delta t)} + \boldsymbol{M}_{(t+\Delta t)}\boldsymbol{R}\boldsymbol{M}^T{}_{(t+\Delta t)}],\\
\boldsymbol{\rho}_{(t+\Delta t)} &= \tilde{\boldsymbol{\rho}}_{(t+\Delta t)} - \boldsymbol{K}_{(t+\Delta t)}\boldsymbol{H}_{(t+\Delta t)}\tilde{\boldsymbol{\rho}}_{(t+\Delta t)},
\end{aligned} \tag{11}$$

where $\boldsymbol{H} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}|_{x_t}$ and $\boldsymbol{M} = \frac{\partial \boldsymbol{h}}{\partial \boldsymbol{\varepsilon}}|_{x_t}$ are 1×6 vectors. $\boldsymbol{K}$ is the 6×1 Kalman gain vector. Fig. 3 shows the overall massively parallel dynamic state estimation operation flowchart.

State estimation based on EKF includes several matrix-vector and matrix-matrix products which are computationally intensive jobs especially for large-scale systems. There are also several independent tasks in state prediction and state filtering stages which are good candidate for parallelization. Computations of $\tilde{\boldsymbol{x}}_{(t+\Delta t)}$ and $\tilde{\boldsymbol{\rho}}_{(t+\Delta t)}$ are independent of each other. Inside $\tilde{\boldsymbol{\rho}}_{(t+\Delta t)}$, calculation of $\boldsymbol{F}_t\boldsymbol{\rho}_t\boldsymbol{F}_t{}^T$ and $\boldsymbol{L}_t\boldsymbol{Q}_t\boldsymbol{L}_t{}^T$ can be done simultaneously. For state filtering it is necessary to first calculate $\boldsymbol{K}_{(t+\Delta t)}$ and $\boldsymbol{\lambda}_{(t+\Delta t)}$ sequentially. At the final step $\hat{\boldsymbol{x}}_{(t+\Delta t)}$ and $\boldsymbol{\rho}_{(t+\Delta t)}$ are independent, too. Defining individual CUDA kernel (functional program which generate a large number of threads for data parallelism) for each independent task, all of them can be run in parallel to reduce the computation time. Also inside each kernel all threads execute the same function in parallel. It should be considered that since the GPU's architecture is considerably different from that of a traditional CPU, it requires a completely different algorithmic approach for implementation. Fig. 2 explain the proposed approach for a system with n generators. For brevity only calculation of $\tilde{\boldsymbol{x}}_{(t+\Delta t)}$ is shown in detail, where on the GPU the states of all generators can be estimated simultaneously. Moreover, for each generator all the independent tasks can be run in parallel.

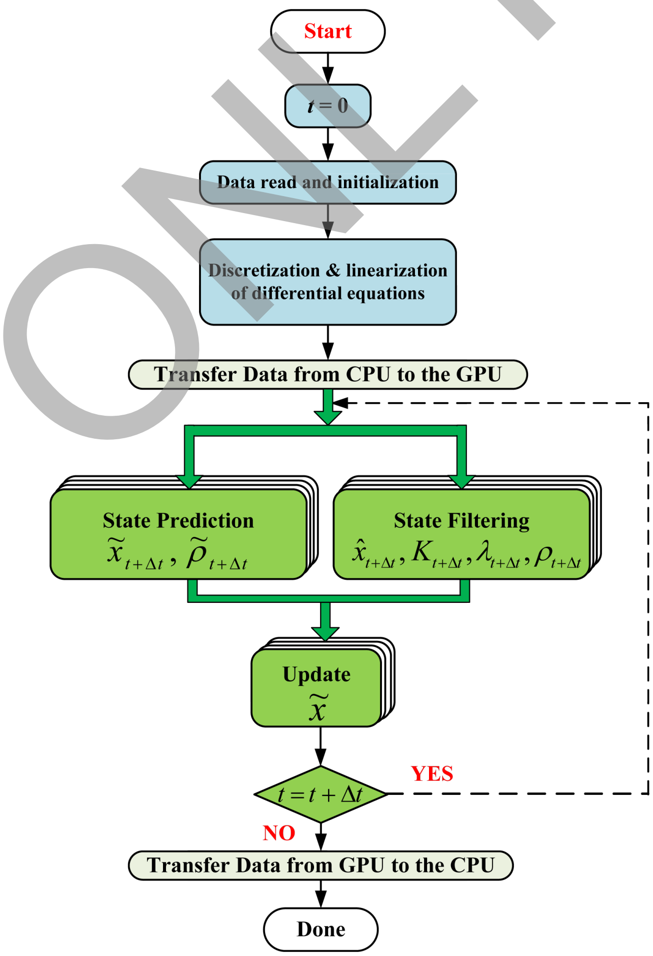
Start
$t = 0$
Data read and initialization
Discretization & linearization of differential equations
Transfer Data from CPU to the GPU
State Prediction
$\tilde{x}_{t+\Delta t}, \tilde{\rho}_{t+\Delta t}$
State Filtering
$\hat{x}_{t+\Delta t}, K_{t+\Delta t}, \lambda_{t+\Delta t}, \rho_{t+\Delta t}$
Update
$\tilde{x}$
$t = t + \Delta t$
YES
NO
Transfer Data from GPU to the CPU
Done

Fig. 3. Massively parallel dynamic state estimation operation flowchart.

III. Case Study and Discussion

To evaluate accuracy and efficiency of the parallel DSE algorithms, the results of state estimation were compared with time-domain simulation in Matlab® and multi-thread CPU-based estimation algorithm. Since the matrices are highly sparse, all matrices and vectors are stored in compressed row format to reduce the computational burden. Large-scale test power systems were constructed to explore the efficiency of the GPU-based program. Case 1 is the IEEE 39-bus system which was duplicated and interconnected to create large-scale systems. It is assumed that PMUs are installed at all generator buses, so that the states of each generator are estimated locally and in par-

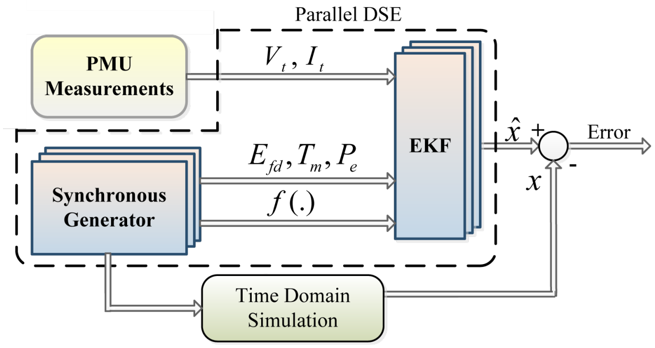

Fig. 4. Extended Kalman filter based parallel DSE block diagram.

allel. The refresh rate of PMU measurements is considered as 60/s which is equal to the simulation time-step of 16.6 ms. The signals T_m, E_{fd}, v_t and the measurement signal T_e are used as the input for the GPU-based algorithm. The block diagram of the proposed EKF based parallel DSE is shown in Fig. 4.

A. Hardware Setup

The hardware used in this work is one unit of Tesla^{TM}S2050 GPU from NVIDIA®. It has 448 cores which deliver up to 515 Gigaflops of double-precision peak performance. This device contains 14 streaming multiprocessors (SMs), each with 32 streaming processors (SPs), an instruction unit, and on-chip memory [16]. CUDA version 5.0 with compute capability 2.0 is used for programming. The CPU is the quad-core AMD® Phenom^{TM} II with 3.2 GHz core clock and 12 GB memory, running 64-bit Windows 7® operating system. OpenMP® standard and C++ was used for developing the multi-thread CPU program.

B. Accuracy and Speed-up Analysis of the GPU-based DSE

Table I shows a summary of the results obtained using double precision (64-bit) format for both CPU and GPU codes as the system size increased. The execution time for GPU-based program (T_{Ex}^{GPU}) is the total of execution and communication time. Massively parallel implementation of the DSE on GPU is done using the state-space model presented in (5) and (6). Performance of the proposed method, was evaluated for different case studies. The estimated states for generator number 1 in Case 1 are shown in Fig. 5 and Fig. 6. It is clear from the results that the proposed approach can accurately track the system dynamics. The small differences compared to the time domain simulation results are justifiable considering the fact that the order of block execution in each GPU grid is undefined in kernel definition. Therefore, it leads to slightly different results if different CUDA blocks perform calculations on overlapping portions of data. Small oscillations in the results are due to using corrupted measurements with Gaussian noise.

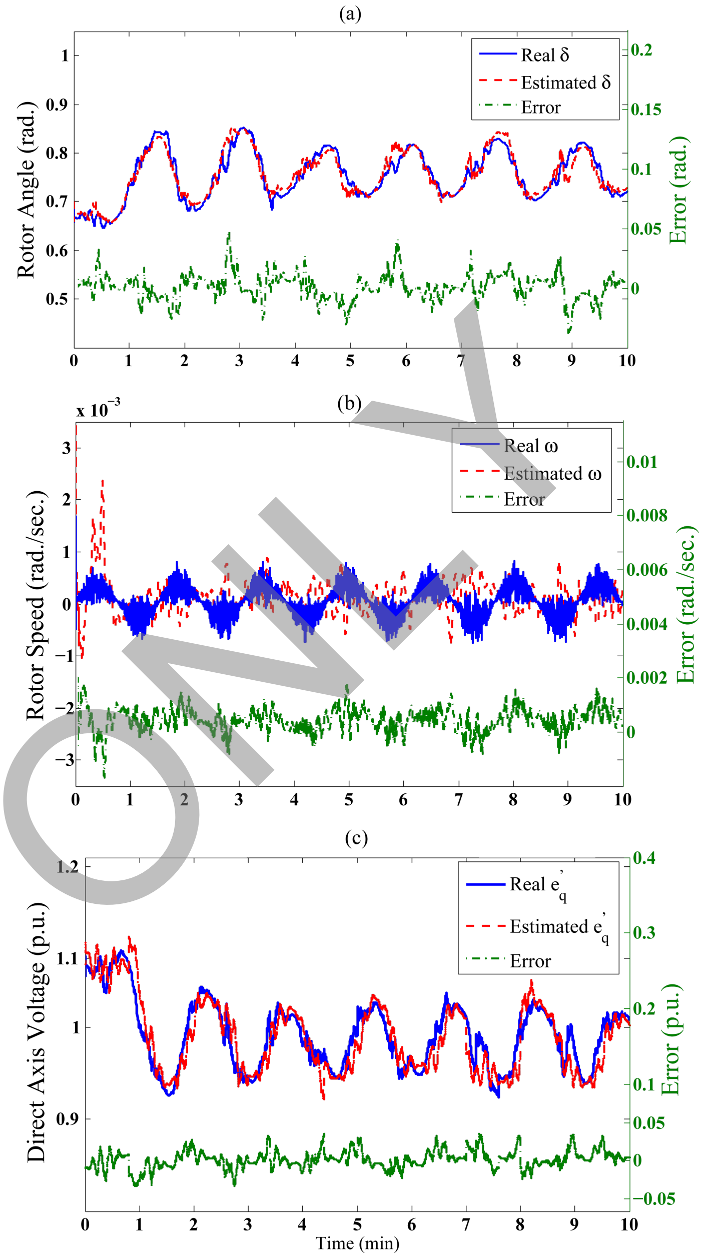

Fig. 5. Estimated states: (a) rotor angle, (b) rotor speed, (c) direct axis voltage.

For the computational speed-up comparison, two separate simulation codes were provided: one in C++ as multi-thread program, and the other an integration of C++ and CUDA in GPU. The objective was to see how the execution time for the CPU and the GPU increases as the number of generators increased. As shown in Fig. 7, the advantage of utilizing GPU for parallelization is highlighted as the size of the system increasing. The reason is that for small size of data the communication overhead between the CPU and GPU supersedes the execution time on the CPU. Since the programming structure is one of the most important factors which affects the processing time, it is possible to obtain faster results with a different programming style. However, the speed-up shown in Table I would

TABLE I
Summary of Model Complexity and Execution Time Under Normal Operation Condition

Case	No. of generators	No. of Buses	No. of states	State transition F(x)	T_{Ex}^{CPU}	T_{Ex}^{GPU} Comp.	T_{Ex}^{GPU} Comm.	Speed-up
1	10	39	60	60 × 60	6ms	4.3ms	2.7ms	0.85
2	20	78	120	120 × 120	13ms	5.8ms	3.2ms	1.4
3	40	156	240	240 × 240	23ms	9.8ms	5.2ms	1.53
4	80	312	480	480 × 480	52ms	23.4ms	8.6ms	1.73
5	160	624	690	960 × 960	2.1s	37.9ms	13.1ms	2.5
6	320	1248	1920	1920 × 1920	4.15s	44.4ms	25.2ms	3.5
7	640	2496	3840	3840 × 3840	8.67s	55.9ms	46.1ms	5.1
8	1280	4992	7680	7680 × 7680	17.63s	1.48s	1.05s	6.96
9	2560	9984	15360	15360 × 15360	36.96s	2.84s	1.32s	8.88
10	5120	19968	30720	30720 × 30720	81.3s	5.99s	2.13s	10.02

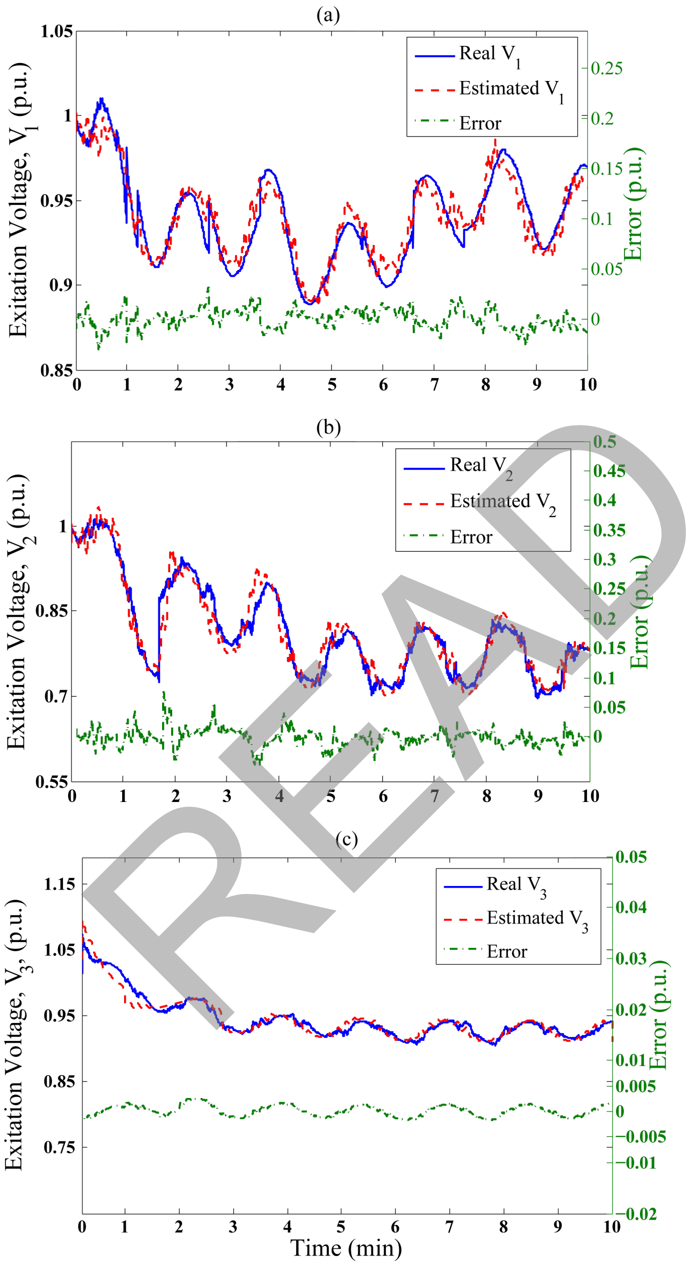

Fig. 6. Estimated states: excitation voltage (a) v_1, (b) v_2, (c) v_3.

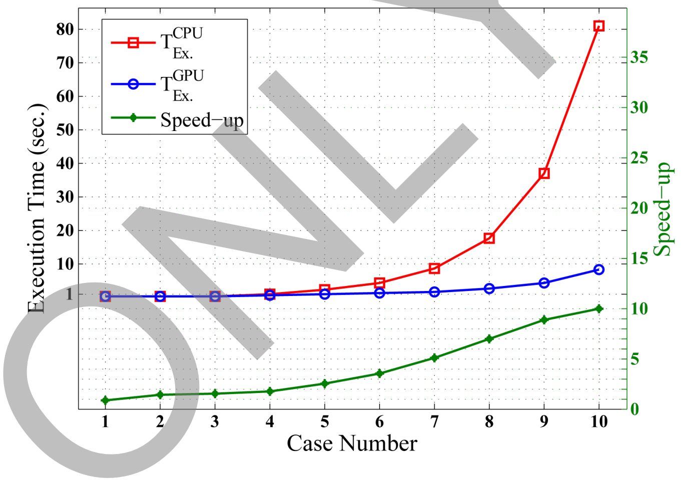

Fig. 7. Execution time ($T_{Ex.}$) and speed-up (S_p) comparisons of CPU-based and GPU-based DSE.

still be valid for increasing system sizes although with a slightly different value.

IV. Conclusion

The increasing size and complexity of modern power systems has made state estimation a slow and computationally expensive process. To alleviate this, most of the existing DSE approaches neglect the detailed model of the generator. This paper is an effort to implement DSE of detailed synchronous generator on GPU-based parallel computing platform by massive-threading programming. Numerical experiments in this work proved that successful parallelization reduces the execution time; 10.02 times faster state estimation for a system with 5120 generators is reported. For future work, the proposed generator state estimation approach will be combined with the network model for a comprehensive DSE.

References

[1] G. N. Korres, and G. C. Contaxis, “Application of a reduced model to a distributed state estimator”, *Proc. of IEEE PES,* vol. 2, pp. 999-1004, 2000.

[2] J. Zhang, G. Welch, G. Bishop, “LoDiM: a novel power system state estimation method with dynamic measurement selection”, *Proc. of IEEE PES,* pp.1-7, July 2011.

[3] T. Van Cutsem and M. Ribbens-Pavella, “Critical survey of hierarchical methods for state estimation of electrical power systems”, *IEEE Trans. on Power App. and Syst.,* vol. 102, no. 10, pp. 3415-3424, Oct. 1983.

[4] A. Gómez-Expósito and A. de la Villa Jaén, "Two-level state estimation with local measurement pre-processing", *IEEE Trans. on Power Syst.,* vol. 24, no. 2, pp. 676-684, May 2009.

[5] G. Valverde, S. Chakrabarti, E. Kyriakides, and V. Terzija, "A constrained formulation for hybrid state estimation", *IEEE Trans. on Power Syst.,* vol. 26, no. 3, pp. 1102-1109, Aug. 2011.

[6] E. Ghahremani, I. Kamwa, "Dynamic state estimation in power system by applying the extended kalman filter with unknown inputs to phasor measurements", *IEEE Trans. on power sys.,* vol. 26, no. 4, pp. 2556-2566, 2011.

[7] G. G. Rigatos, "A derivative-free kalman filtering approach to state estimation-based control of nonlinear systems", *IEEE Trans. on Ind. Elec.,* vol. 59, no. 10, pp. 3987-3997, Oct. 2012.

[8] S. Wang, W. Gao, A. P. S. Meliopoulos, "An alternative method for power system dynamic state estimation based on unscented transform", *IEEE Trans. on Power Syst.,* vol. 27, no. 2, pp. 942-950, May 2012.

[9] Z. Huang, K. Schneider, J. Neplocha, "Feasibility studies of applying Kalman filter techniques to power system dynamic state estimation", *Proc. Int. Power Engineering Conf.,* pp. 376C382, 2007.

[10] L. Lin, Linawati, L. Jasa, E. Ambikairajah, "A hybrid state estimation scheme for power system", *Proc. IEEE Circuits and Systems Conf.,* vol. 1, pp. 555-558, 2002.

[11] E. Farantatos, G. K. Stefopoulos, G. J. Cokkinides, A. P. Meliopoulos, "PMU-based dynamic state estimation electric power systems", *Proc. of PES,* pp. 1C8, 2009.

[12] Z. Zhou, V. Dinavahi, "Parallel massive-thread electromagnetic transient simulation on GPU", *IEEE Trans. on Power Del.,* vol. 29, no. 3, pp. 1045-1053, June 2014.

[13] V. Jalili-Marandi, Z. Zhou, V. Dinavahi, "Large-scale transient stability simulation of electrical power systems on parallel GPUs", *IEEE Trans. on Parallel and Dist. Syst.,* vol. 23, no. 7, pp. 1255-1266, July 2012.

[14] IEEE Std 421.5-2005, "IEEE recommended practice for excitation system models for power system stability studies", *IEEE Power Eng. Soc.,* pp. 1-85, 2006.

[15] P. Kundur, "Power System Stability and Control", *New York: McGraw Hill,* 1994.

[16] NVIDIA, "NVIDIA Tesla: a unified graphics and computing architecture", *NVIDIA CUDA C Programming Guide 4.0.,* 2013.