IET Generation, Transmission & Distribution

Research Article

Parallel relaxation-based joint dynamic state estimation of large-scale power systems

ISSN 1751-8687
Received on 2nd July 2015
Revised on 9th September 2015
Accepted on 29th September 2015
doi: 10.1049/iet-gtd.2015.0808
www.ietdl.org

Hadis Karimipour, Venkata Dinavahi ✉

Department of Electrical and Computer Engineering, University of Alberta, Edmonton, Alberta T6G 2V4, Canada
✉ *E-mail: dinavahi@ece.ualberta.ca*

Abstract: Massive amounts of data generated in large-scale grids poses a formidable challenge for real-time monitoring of power systems. Dynamic state estimation which is a prerequisite for normal operation of power systems involves the time-constrained solution of a large set of equations which requires significant computational resources. In this study, an efficient and accurate relaxation-based parallel processing technique is proposed in the presence of phasor measurement units. A combination of different types of parallelism is used on both single and multiple graphic processing units to accelerate large-scale joint dynamic state estimation simulation. The estimation results for both generator and network states verify that proper massive-thread parallel programming makes the entire implementation scalable and efficient with high accuracy.

1 Introduction

Dynamic state estimation (DSE) [1] which is paramount for secure monitoring and control of a power system to ensure its stability and reliability, possesses the ability to predict the system states in advance and within a short time interval. By modelling the time varying nature of the system to provide a full dynamic view of its behaviour, DSE alleviates losses under drastic changes during load fluctuations or network switching. Continued growth in demand followed by system development and complex interconnections within the new smart grid paradigm has led to significant operational and control problems. These problems necessitate the need for major changes in computational resources for real-time action by system operators in energy control centres which are hard to achieve using traditional measurements provided by supervisory control and data acquisition system (SCADA) [2]. Advancements in high-performance computing (HPC) [3] such as graphic processing unit (GPU) technology along with the increasing deployment of high-speed time-synchronised phasor measurement units (PMUs) [4] in wide areas provide the opportunity for real-time monitoring of the *dynamic* states, e.g., generator rotor angles and generator speed, in addition to the *static* network states of voltage magnitudes and phase angles.

Conventional methods to improve the cycle time for state estimation include the following main strategies:

- *Complexity reduction* to alleviate the computational burden using reduced order model, lower dimensional measurement data or partial update of the Jacobian matrix [5–7].
- *Hierarchical two-level state estimation* which decomposes the whole system into several independent subsystems wherein each subsystem uses its own measurement set to estimate the states locally and then sends the estimated states to a centralised coordinator [8–10].
- *Distributed state estimation* where each subsystem can run its local state estimation and exchange data between the substations without the central coordinator [11–14].
- *Parallel state estimation* utilises distributed computation on multiple computational servers by decomposing the problem into several subproblems running the estimation process simultaneously in all the subsystems [15, 16].

Most of the approaches dealing with DSE try to improve the computational performance of the steady-state estimation process [17–20] which only provides a series of snapshots of system conditions, where the dynamic transition between the snapshots is overlooked. However, few researchers focused on the dynamic parameters of the synchronous generator, which plays a vital role in a power system [21–24].

Although the above approaches tried to improve the process of state estimation, they all have their own drawbacks. Performance degradation may occur in complexity reduction approaches due to the constraints of the number of measurements or due to neglecting the model details. The main issue with hierarchical methods is the communication overhead between subsystems and the delay caused by coordination stage. In addition, in both distributed and hierarchical methods the observability of the subsystems is the underlying assumption which may not be feasible in practice. Both distributed and parallel approaches accelerate the computational process; however, they mostly decompose the original system randomly neglecting the effect of subsystems on each other which may result in inaccurate estimation.

Moreover, in almost all of the aforementioned work, DSE has been performed for either a single machine or a small power system. In large-scale power systems with detailed modelling of grid components, a key issue in the design of dynamic state estimator based on dynamic models is the requirement for high-performance computation resources.

From computing source point-of-view supercomputers, multiprocessor networks and various types of parallel processing architectures such as multiple-instruction multiple-data, single-instruction multiple-data (SIMD), and distributed memory, have already been employed for power system analysis [25, 26]. All of these approaches accelerate the simulation process to some extent; however, they are limited by some important factors such as cost, system size, programmability, and communication issues.

The objective of this paper is to explore DSE using extended Kalman filter (EKF) in large-scale power systems utilising detailed synchronous generator modelling which complicates the estimation problem resulting in a high computational burden. Recently, GPUs have tremendously accelerated many HPC applications as well as power system transient stability, electromagnetic transients, and power flow analysis [27–30]. This paper proposes a heterogeneous parallel multi-GPU and multi-core CPU implementation of large-scale DSE based on an accurate and robust relaxation method to estimate both generator (dynamic) and network (static) states of the power system. Coarse-grained parallelism using relaxation method along with its implementation on a multi-GPU computational server where each of the individual GPUs has a fine-grained parallel architecture enables significant acceleration of the DSE process. Relaxation method had been previously used in other area such as simulation of very large-scale integration (VLSI) circuits [31], and power system transient stability simulation [32].

The first nontrivial task in relaxation-based joint DSE (RJDSE) is domain decomposition. There are two main approaches used for system decomposition in DSE. Approaches which randomly or geographically [11–13] partition the system are stymied by the computational load-balancing problem and may result in inaccuracy due to neglecting the effect of subsystems. Other methods which rely on the graph theory are highly complex for implementation on large-scale systems [33, 34]. In this work, based on the slow coherency method the original system is decomposed into subsystems in which tightly coupled variables are grouped together, along with equal distribution of computation load on multiple GPUs guaranteed by relaxation method. Also, bad data analysis can be run locally in parallel with the state estimation.

Since traditional measurement sets using SCADA cannot provide adequate data to capture the dynamic nature of the power systems, PMUs with higher refresh rate than SCADA are used in this paper. Considering rapid deployment of PMUs in power systems, it is expected that the entire network will be fully observable only using PMUs in the near future. Through careful PMU placement [35–37], it is possible to make the whole network observable using minimum number of PMUs to obtain a redundant enough measurement set. To reduce data transfer and for practicability optimal placement of PMUs is considered which also improves the security of the system.

In summary, the main contributions of the proposed approach are as follows:

- Parallel multi-GPU implementation of joint generator and network DSE.
- Application of relaxation method which distributes equal workload among all the processors and eliminates the need for all subsystem to be observable.
- Domain decomposition based on coherency method which reduces the effect of non-overlapping decomposition on the accuracy.
- Eliminating central coordinator which reduces the communication time between the subsystems.
- Distributed localised bad data analysis in parallel with state estimation.

The organisation of this paper is as follows. Section 2 provides formulation and state estimation model used in this work. Section 3 explains the proposed RJDSE method. The simulation results are provided in Section 4 followed by conclusion in Section 5.

2 Formulation and state-space model

Joint dynamic and static state estimation for a multi-machine power system can be mathematically described as follows

$$\begin{cases} \boldsymbol{f}(\dot{\boldsymbol{x}}, \boldsymbol{x}, t, \boldsymbol{u}) = \boldsymbol{0}, & x_{\mathrm{g}}(t_0) = x_{\mathrm{g}0} \\ \boldsymbol{g}(\boldsymbol{x}, t, \boldsymbol{u}) = \boldsymbol{0}, & \\ \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{Z}, \boldsymbol{\varepsilon}) = \boldsymbol{0}, & x_{\mathrm{n}}(t_0) = x_{\mathrm{n}0} \end{cases} \tag{1}$$

where $\boldsymbol{x}$ is the vector of state variable including $\boldsymbol{x}_{\mathrm{g}}$, the dynamic state of generator and, $\boldsymbol{x}_{\mathrm{n}}$, the static state of the network. x_0 is the initial values of state variables. $\boldsymbol{f}(.)$ describe the non-linear dynamic behaviour of the generators, $\boldsymbol{g}(.)$ model the output function, and $\boldsymbol{h}(.)$ is the non-linear function of network measurement. $\boldsymbol{u}$ and $\boldsymbol{Z}$ represent the output vector and network measurements vector, respectively. t represents the simulation time. For a system with m generators and n buses, there are $9 \times m + 2n$ elements in vector $\boldsymbol{x}$: 9 states per generator, and n voltage magnitudes, n phase angles. Bold notation refers to vectors and matrices.

The detailed ninth-order state-space model of a single generator including automatic voltage regulator (AVR) and power system stabiliser (PSS) used in this work which includes two windings on the d-axis (one excitation field and one damper) and two damper windings on the q-axis can be written as follows [38]

$$\begin{aligned} \dot{\boldsymbol{x}}_{\mathrm{g}1}(t) &= \omega_R \cdot \boldsymbol{x}_{\mathrm{g}2}(t), \\ \dot{\boldsymbol{x}}_{\mathrm{g}2}(t) &= \frac{1}{2\mathrm{S}}[T_{\mathrm{e}}(t) + T_{\mathrm{m}} - D \cdot \boldsymbol{x}_{\mathrm{g}2}(t)], \\ \dot{\boldsymbol{x}}_{\mathrm{g}3}(t) &= \omega_R \cdot [e_{\mathrm{f}d}(t) - R_{\mathrm{f}d} \cdot I_{\mathrm{f}d}(t)], \\ \dot{\boldsymbol{x}}_{\mathrm{g}4}(t) &= -\omega_R \cdot R_{1d} \cdot I_{1d}(t), \\ \dot{\boldsymbol{x}}_{\mathrm{g}5}(t) &= -\omega_R \cdot R_{1q} \cdot I_{1q}(t), \\ \dot{\boldsymbol{x}}_{\mathrm{g}6}(t) &= -\omega_R \cdot R_{2q} \cdot I_{2q}(t), \\ \dot{\boldsymbol{x}}_{\mathrm{g}7}(t) &= \frac{1}{T_R}[v_{\mathrm{t}} - x_{\mathrm{g}7}(t)], \\ \dot{\boldsymbol{x}}_{\mathrm{g}8}(t) &= \frac{1}{T_2}\left[T_1 K_{\mathrm{stab}} \cdot \dot{\boldsymbol{x}}_{\mathrm{g}2} - \boldsymbol{x}_{\mathrm{g}8} + \left(1 - \frac{T_1}{T_W}\right)\alpha\right], \\ \dot{\boldsymbol{x}}_{\mathrm{g}9}(t) &= \frac{1}{T_A}[\beta - \boldsymbol{x}_{\mathrm{g}9}]. \end{aligned} \tag{2}$$

T_{m} and T_{e} represent the mechanical input torque and the electrical output torque, respectively. $e_{\mathrm{f}d}$ and $I_{\mathrm{f}d}$ are field voltage and current. I_{1d}, I_{1q}, and I_{2q} describe d- and q-axis currents. For the whole system, according to the aforementioned formulations the $9 \times m$ vector of state variables $\boldsymbol{x}_{\mathrm{g}}$ of the synchronous generator is given as

$$\boldsymbol{x}_{\mathrm{g}} = [\boldsymbol{\delta}, \boldsymbol{\Delta\omega}, \boldsymbol{\psi}_{\mathrm{f}d}, \boldsymbol{\psi}_{1d}, \boldsymbol{\psi}_{1q}, \boldsymbol{\psi}_{2q}, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]^{\mathrm{T}}, \tag{3}$$

where $\boldsymbol{\delta}$ and $\boldsymbol{\Delta\omega}$ represent vectors of rotor speed and angle, respectively. $\boldsymbol{\psi}_{\mathrm{fd}}$, $\boldsymbol{\psi}_{\mathrm{1d}}$, $\boldsymbol{\psi}_{\mathrm{1q}}$, $\boldsymbol{\psi}_{2q}$ represent vectors of rotor flux linkages and $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, $\boldsymbol{v}_3$ are vectors of exciter voltages; $\boldsymbol{v}_{\mathrm{t}}$ represents the vector of terminal voltage which can be calculated as the network state. For a system with m generators, all of aforementioned vectors are $m \times 1$. Fig. 1 shows AC5A type excitation system [39].

The output electrical torque T_{e} of the machine can be written as

$$T_{\mathrm{e}} = [L''_{ad} - L''_{aq}]I_d I_q - \left[\frac{\psi_{\mathrm{f}d}}{L_{\mathrm{f}d}} + \frac{\psi_{d1}}{L_{d1}}\right]L''_{ad} I_q + \left[\frac{\psi_{q1}}{L_{q1}} + \frac{\psi_{q2}}{L_{q2}}\right]L''_{aq} I_d. \tag{4}$$

where ω_R, S, D, $R_{\mathrm{f}d}$, R_{1d}, R_{1q}, R_{2q}, $L_{\mathrm{f}d}$, L_{d1}, L_{q1}, L_{q2}, L''_{ad}, L''_{aq}, T_R, T_w, T_1, T_2, and K_{stab} are constant system parameters whose definition can be found in [38].

The continuous-time differential equations were discretised using the Trapezoidal integration method and then linearised resulting in

$$\boldsymbol{x}_{\mathrm{g}}^{t+\tau} - \boldsymbol{x}_{\mathrm{g}}^{t} = \frac{\tau}{2}[\boldsymbol{f}(\boldsymbol{x}_{\mathrm{g}}, t + \tau, \boldsymbol{u}) + \boldsymbol{f}(\boldsymbol{x}_{\mathrm{g}}, t, \boldsymbol{u})], \tag{5}$$

$$\boldsymbol{x}_{\mathrm{g}}^{t+\tau} = \boldsymbol{F}_{\boldsymbol{x}_{\mathrm{g}}^{t}} x_{\mathrm{g}}^{t} + \boldsymbol{\eta}^{t}, \boldsymbol{\eta}^{t} \sim N(0, \boldsymbol{Q}^{t}), \tag{6}$$

where $\boldsymbol{F}_{\boldsymbol{x}_{\mathrm{g}}^{t}} = \partial \boldsymbol{f}/\partial \boldsymbol{x}_{\mathrm{g}}^{t}|_{\boldsymbol{x}_{\mathrm{g}}^{t}}$ represents the 9×9 state transition between two time steps. $\boldsymbol{\tau}$ is the integration time-step. $\boldsymbol{\eta}$ and $\boldsymbol{Q}$ are 9×1 linearisation error and 9×9 error covariance matrix, respectively. The resulting linear algebraic equations are solved to obtain the generator states.

The non-linear measurement function $\boldsymbol{\gamma}(.)$ can be written as

$$\boldsymbol{Z} = \boldsymbol{\gamma}(\boldsymbol{x}_n) + \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \sim N(0, \boldsymbol{R}), \tag{7}$$

where $\boldsymbol{\gamma}(.)$ and $\boldsymbol{\varepsilon}$, are the nonlinear network measurement functions,

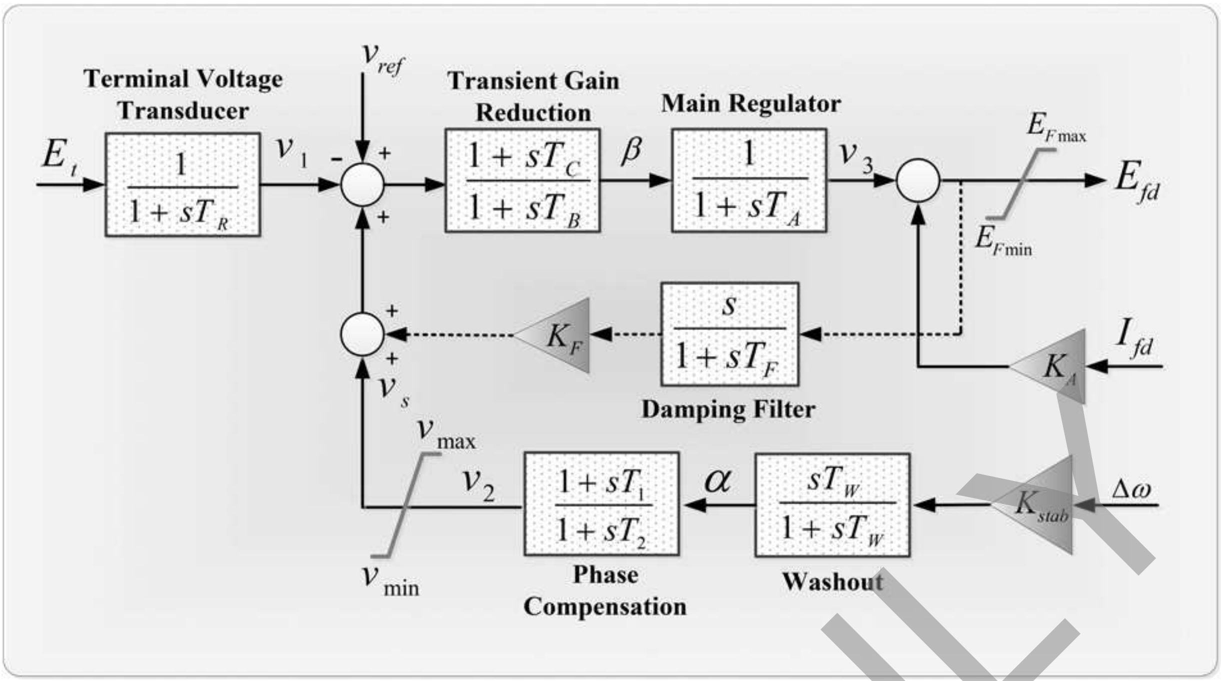

Fig. 1 *Synchronous generator excitation system with AVR and PSS*

and uncorrelated Gaussian network measurement errors, respectively. $\boldsymbol{x}_\text{n}$ represents the vector of network state including voltage magnitude ($\boldsymbol{v}$) and phase angle ($\boldsymbol{\phi}$). To minimise the weighted least-squares error of network state estimation the following objective function should be minimised [40]

$$\boldsymbol{J}(\boldsymbol{x}_\text{n}) = [\boldsymbol{Z} - \boldsymbol{\gamma}(\boldsymbol{x}_\text{n})]^\text{T}\boldsymbol{R}^{-1}[\boldsymbol{Z} - \boldsymbol{\gamma}(\boldsymbol{x}_\text{n})]. \tag{8}$$

In a system with n buses and l branch, $\boldsymbol{R}$ is the error covariance matrix of network measurement with the maximum size of $(2n+2l+1)\times(2n+2l+1)$. Also, $\boldsymbol{Z}$, $\boldsymbol{\gamma}(.)$, and $\boldsymbol{\varepsilon}$ have the maximum size of $(2n+2l+1)\times 1$. Satisfying first order optimality condition and substituting the first-order Taylor's expansion of $\boldsymbol{\gamma}(.)$ around $\boldsymbol{x}_{\text{n}0}$ in (8), the following equation can be solved iteratively to find the static state of the system.

$$\boldsymbol{\Gamma}^\text{T}(\boldsymbol{x}_\text{n})\boldsymbol{R}^{-1}\boldsymbol{\Gamma}(\boldsymbol{x}_\text{n})\boldsymbol{\Delta}(\boldsymbol{x}_\text{n}) = \boldsymbol{\Gamma}^\text{T}(\boldsymbol{x}_\text{n})R^{-1}[\boldsymbol{Z} - \boldsymbol{\gamma}(\boldsymbol{x}_\text{n})], \tag{9}$$

$$\boldsymbol{\Delta}(\boldsymbol{x}_\text{n}) = \boldsymbol{\lambda}(\boldsymbol{x}_\text{n}, \boldsymbol{\varepsilon}), \tag{10}$$

where $\boldsymbol{\Gamma} = \partial\boldsymbol{\gamma}/\partial\boldsymbol{x}_\text{n}$ is the Jacobian matrix with the maximum size of $(2n+2l+1)\times 2n$ and $\Delta(\boldsymbol{x}) = \boldsymbol{x}_\text{n} - \boldsymbol{x}_{\text{n}0}$ is the $2n\times 1$ static state mismatch vector.

For a single generator, using the measurement and estimated states at the time instant t, the predicted value $\tilde{\boldsymbol{x}}_\text{g}^{t+\tau}$ can be formulated as

$$\begin{aligned} \tilde{\boldsymbol{x}}_\text{g}^{t+\tau} &= \boldsymbol{F}_\text{t}\hat{\boldsymbol{x}}_\text{g}^t, \quad (\boldsymbol{x}_\text{g}^t - \hat{\boldsymbol{x}}_\text{g}^t) \sim N(0, \boldsymbol{\rho}_\text{t}), \\ \tilde{\boldsymbol{\rho}}_{(\text{t}+\tau)} &= \boldsymbol{F}_\text{t}\boldsymbol{\rho}_\text{t}\boldsymbol{F}_\text{t}^\text{T} + \boldsymbol{L}_\text{t}\boldsymbol{Q}_\text{t}\boldsymbol{L}_\text{t}^\text{T}, \quad (\boldsymbol{x}_\text{g}^t - \tilde{\boldsymbol{x}}_\text{g}^t) \sim N(0, \tilde{\boldsymbol{\rho}}_\text{t}), \end{aligned} \tag{11}$$

where $\boldsymbol{L}_\text{t} = \partial\boldsymbol{f}/\partial\boldsymbol{\eta}|_{\boldsymbol{x}_\text{g}^t}$, $\boldsymbol{\rho}$ and $\tilde{\boldsymbol{\rho}}$ are 9×9 error covariance matrices for estimated and predicted values, respectively. $\tilde{\boldsymbol{x}}_\text{g}$ represent the predicted state and $\hat{\boldsymbol{x}}_\text{g}$ is the estimated state.

The updated state through EKF can be written as

$$\hat{\boldsymbol{x}}_\text{g}^{t+\tau} = \tilde{\boldsymbol{x}}_\text{g}^{t+\tau} + \boldsymbol{K}^{\boldsymbol{a}}_{(\text{t}+\tau)}\boldsymbol{e}_{(\text{t}+\tau)} \tag{12}$$

where $\boldsymbol{K}^{\boldsymbol{a}}$ and $\boldsymbol{e}$ are 9×9 Kalman gain vector and 9×1 estimation error vector, respectively. The same formulation will stand for network state estimation as well.

3 Relaxation-based joint dynamic state estimation

3.1 Optimal PMU placement

Proper measurement set is an important factor, which affects both accuracy and speed of any estimation method. For higher accuracy and sample rates, PMUs are selected for this work. PMU measures voltage phasor of the bus and currents of all branches connected to it. It is not necessary to install PMU in all of the buses for full observability. Optimal PMU placement for full observability of the system can be written as a linear programming problem to minimise following objective function in a system with n buses:

$$\begin{aligned} \boldsymbol{J}(x) &= \sum_{i=1}^{n} \boldsymbol{I}_i \times \boldsymbol{p}_i, \\ \text{subject to: } &\sum_{j=1}^{n} c_{i,j} \times p_i \geq 1 \text{ at bus } i \end{aligned} \tag{13}$$

where $c_{i,j}$ is the element of connectivity matrix which is 1 if bus i and bus j are connected, and 0 otherwise. I_i is the cost of PMU installation at bus i and p_i is the binary variable equal to 1 or 0 depending on whether PMU is installed on bust i or not. In order to handle multiple contingency and to increase the reliability of the system in presence of bad data, backup conventional data are considered in proper location along with critical measurements identification [41, 42].

3.2 Domain decomposition

Domain decomposition refers to any technique that divides a system of equations into several subsets that can be solved individually using conventional numerical methods. The preliminary step for applying RJDSE on GPU is to partition the system into interdependent subsystems while making the dependency between any two subsystems weak enough to ignore their interconnection. It is already shown that the rate of convergence in relaxation method is highly dependent on the method of partitioning [43]. In power system analysis after a large disturbance in the system some generators lose their synchronism with the network. Considering the coherency characteristic which reflects the level of dependency between generators, the proposed method partitions the system into several areas in which generators are in step together or coherent. This type of decomposition is independent of size of the system, disturbance, level of complexity and location of disturbance [44]. For efficient parallelisation, load balancing is also considered in domain decomposition. Fig. 2 shows a power system decomposed into J subsystems. Decomposition based on the coherency approach and equal load work criteria, divides the full set of equation into several independent function running in separate GPUs (functional parallelism).

3.3 Relaxation

Relaxation facilitates the parallel solution of the small subsystems, and the exchange of computational data between them. Each subsystem is solved for its local variables, while other subsystems are considered constant or relaxed during the time-step. At the end of each time-step the global variables of all subsystems are exchanged for the next time-step. Using parallelism inherent in the relaxation method, the proposed RJDSE offers a coarse grained parallelisation as a top level algorithm which should be implemented before using a numerical method for solving the system of equations. Utilising equal load distribution among subsystems RJDSE reduces the complexity resulting in faster computations which also makes it efficient for implementation in a multi-GPU architecture.

3.4 Hierarchy of parallelism for RJDSE

At the data-parallel level equations are expressed based on the unique SIMD-based architecture of GPU (fine-grained parallelism). Instead of using single element values vectors or matrices of them are used. In addition, all matrix–matrix and matrix–vector which include many independent *for* loops are implemented in a fully parallel manner. CUDA which is the general-purpose programming model for the GPU hardware was used in this work. The entire simulation code was written in C++ integrated with CUDA using CUBLAS and CUSPARSE libraries [45, 46] using double precision floating point.

After partitioning the system, the following set of equations can describe the dynamics of each subsystem

$$\begin{cases} \boldsymbol{f}(\boldsymbol{x}_1^t,\ \ldots,\boldsymbol{x}_i^{t+\tau},\ \ldots,\boldsymbol{x}_J^t,\dot{\boldsymbol{x}}_1^t,\ \ldots,\dot{\boldsymbol{x}}_i^{t+\tau},\ \ldots,\dot{\boldsymbol{x}}_J^t,t, \\ \boldsymbol{u}_1^t,\ \ldots,\boldsymbol{u}_i^{t+\tau},\ \ldots,\boldsymbol{u}_J^t)=\boldsymbol{0} \\ \boldsymbol{g}(\boldsymbol{x}_1^t,\ \ldots,\boldsymbol{x}_i^{t+\tau},\ \ldots,\boldsymbol{x}_J^t,t,\boldsymbol{u}_1^t,\ \ldots,\boldsymbol{u}_i^{t+\tau},\ \ldots,\boldsymbol{u}_J^t)=\boldsymbol{0} \\ \boldsymbol{h}(\boldsymbol{x}_1^t,\ \ldots,\boldsymbol{x}_i^{t+\tau},\ \ldots,\boldsymbol{x}_J^t,\boldsymbol{Z}_1^t,\ \ldots,\boldsymbol{Z}_i^{t+\tau},\ \ldots,\boldsymbol{Z}_J^t,\boldsymbol{\varepsilon})=\boldsymbol{0} \end{cases} \tag{14}$$

where indices L and G stand for local and global variables, respectively. Equation (14) is solved in parallel and iteratively for all subsystems. After each iteration, the global state variables are exchanged and updated between all interconnected subsystems. To calibrate the results, an overall loop based on the Gauss–Jacobi algorithm is applied outside the solutions of subsystems. Since the Gauss–Jacobi algorithm only uses the previously computed values for the solution of each subsystem, all computations of subsystems can be processed in parallel.

In summary, fine-grained parallelism is performed inside the functional parallelism, and functional parallelism is a subset of coarse-grained parallelism (Fig. 3). In the best-case scenario, coarse-grained parallelism by dividing the system into J subsystems reduce the execution time to τ/J; using functional parallelism and J_1 independent tasks results in execution time of $\tau/J.J_1$; and finally utilising J_0 independent matrix–matrix, matrix–vector, and other type of fine-grained parallelism execution time can be further reduced to $\tau/J.J_1.J_0$. However, in reality the speedup is less than this considering the different costs of parallelisation and data transfer to GPU. The algorithm starts at the top level with Gauss–Jacobi iteration. By functional parallelism, almost equal tasks are assigned to each GPU. The iteration starts at the same time and in parallel inside all GPUs. Inside each iteration, fine-grained parallelism is used to accelerate the process. After each iteration, only estimated states of boundary buses are exchanged. If the Gauss–Jacobi algorithm convergence is not satisfied, then new iterations will be performed. Since the subsystems are fully independent of each other, after each time-step the results of network estimation will be transferred to the generator state estimation model. The same procedure is performed for generator dynamic state estimation. While network estimation is working on estimation for the time-step $t+\tau$, generator state estimation is working for the time-step t, simultaneously. Thus, the network estimation is always one step ahead of the generator estimation. The results will be checked for bad data identification (BDI) after each time-step of network estimation. In case of bad data, only the subsystem affected by bad data will repeat state estimation instead of the whole system. Fig. 4 shows the complete flowchart of the RJDSE algorithm implementation on a multi-GPU architecture controlled by η CPU threads.

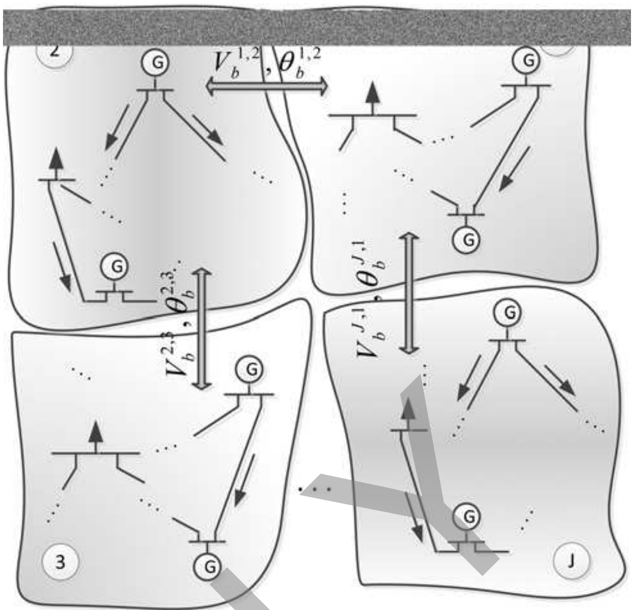
2
G
$V_b^{1,2},\theta_b^{1,2}$
G
G
G
$V_b^{2,3},\theta_b^{2,3}$
$V_b^{J,1},\theta_b^{J,1}$
G
G
G
G
3
J

Fig. 2 *Original power system decomposed into J subsystems for RJDSE implementation*

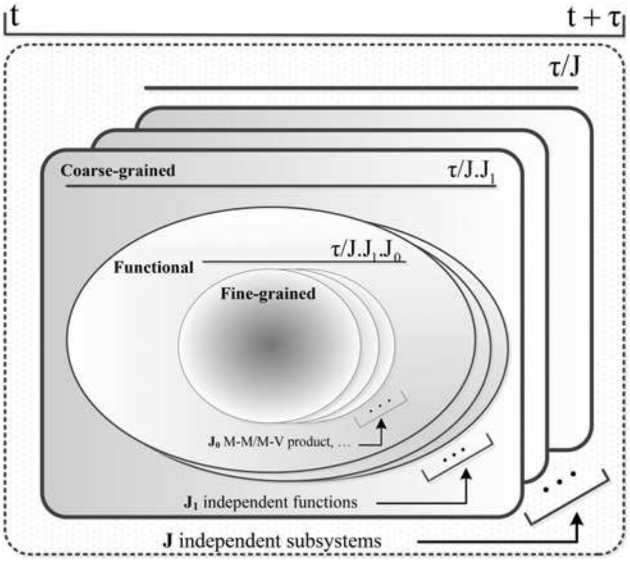

Fig. 3 *Hierarchy of parallelism in RJDSE, τ: integration time-step, t: simulation time*

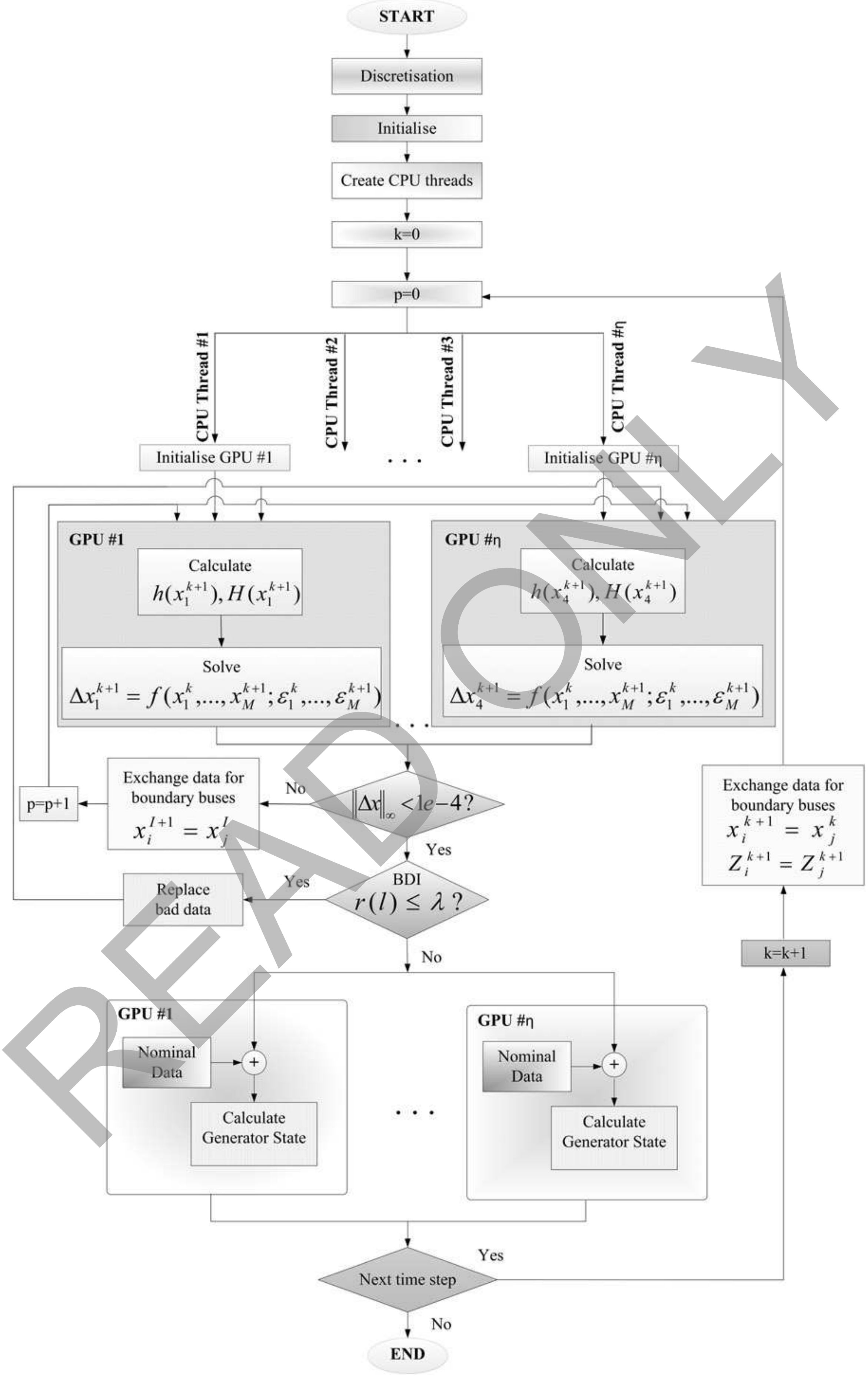

Fig. 4 *Flowchart of RJDSE implementation on multi-GPU architecture with η CPU threads*

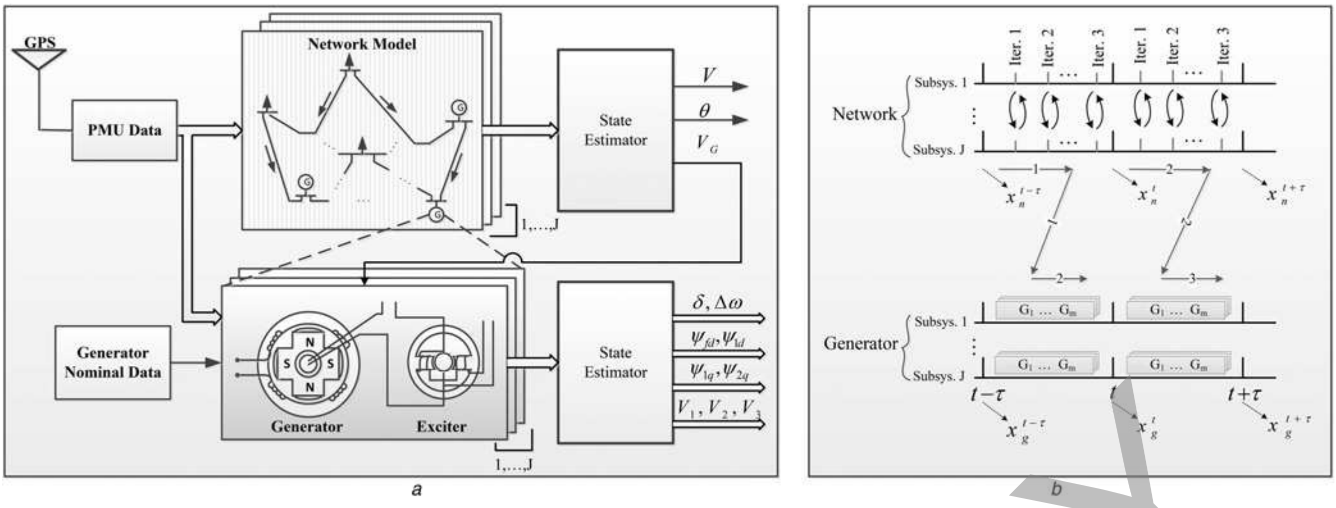

Fig. 5 *Schemes represent*

a Overall block diagram of the proposed RJDSE method
b Time progression of RJDSE on GPU

4 Large-scale DSE case study

The results of relaxation-based joint state estimation implementation on multiple GPUs are demonstrated in this section. The accuracy of the simulation has been verified using the PSS/E® software.

4.1 Test system preparation

Several case studies were built and modelled on PSS/E® using the IEEE 39 bus benchmark system. Larger test systems were constructed by duplicating the 39-bus system and adding interconnections to explore the efficiency of the proposed method. The initial condition for generator states was chosen based on the steady-state conditions.

To generate PMU data, the rectangular components of voltages and currents were obtained by sampling at regular time intervals, and then a Gaussian noise was added to each component. It is assumed that the PMU refresh rate is 30/s. Since the total vector error of PMU measurements should be less than 1% under steady-state conditions [47], both amplitude and phase angles are randomly distorted and the error of resultant vector is checked with the threshold of 1% until satisfied.

Given a PMU at a bus, it is assumed that the bus voltage phasor and all current phasors along lines connected to that bus will be available. For all case studies, PMUs are located with the considerations given Section 3.1. For example, for Case 1 test system PMUs are located at buses {2, 6, 9, 10, 13, 14, 17, 19, 22, 23, 25, 29, 34}. The error covariance matrices considered for the simulations are $\boldsymbol{R} = (0.1)^2 * I(n*n)$ and $\boldsymbol{Q} = (0.05)^2 * I(n*n)$, where $\boldsymbol{I}$ represents the identity matrix, and $\boldsymbol{n}$ is the number of states. The convergence threshold is considered 1×10^{-4} for all of the states. Fig. 5*a* shows the overall block diagram of the proposed method.

4.2 Implementation of RJDSE on multiple GPUs

Case 1 which is the IEEE 39-bus system has been partitioned into four subdomains satisfying both computational load balancing and coherency characteristic of the generators. Similarly, all the large test cases are partitioned. The simulation starts by initialisation on the CPU. After that, the measurement set corresponding to each subsystem was transferred to GPU assigned for that specific subsystem. All the subsystems start the simulation at the same time. After each iteration, boundary data was exchanged among subsystems. Once network state was converged, the results were used for generator state estimation. Network and generator state estimation was running simultaneously (Fig. 5*b*).

4.3 Accuracy analysis and BDI

Accuracy of the proposed method was evaluated under both normal and emergency conditions. A temporary three-phase fault is considered at $t = 3$ s which is cleared after 100 ms. The normalised Euclidian norm (NEN) of the state estimation is defined as a factor to evaluate the accuracy using:

$$\boldsymbol{x}_{\text{NEN}} = \frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|}{\sqrt{\dim(\boldsymbol{x})}}, \qquad (15)$$

where $\boldsymbol{x}_{\text{NEN}}$ is the normalised Euclidian norm of the estimation error, and $\dim(\boldsymbol{x})$ is the dimension of vector $\boldsymbol{x}$. $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ are vector of true states and estimated states, respectively. Results of simulation on all case studies are shown in Fig. 6. Also, the estimated states for Case 1 are shown in Fig. 7. As shown the maximum of the average errors for all case studies are less than 0.001 p.u. In summary, all the simulation results proves that proposed method is able to accurately capture the dynamic behaviour of the system. There are small differences compared to PSS/E (actual) results which are due to the fact that the order of block execution in each GPU grid is undefined in the kernel definition. Therefore, it leads to slightly different results if different CUDA blocks perform calculations on overlapping portions of data.

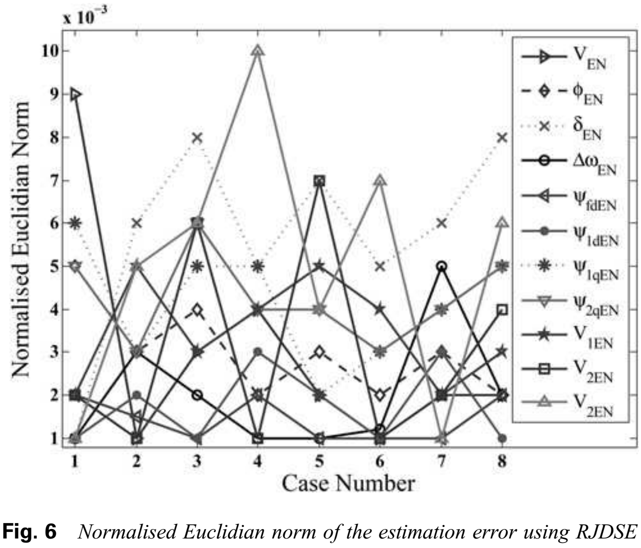

Fig. 6 *Normalised Euclidian norm of the estimation error using RJDSE*

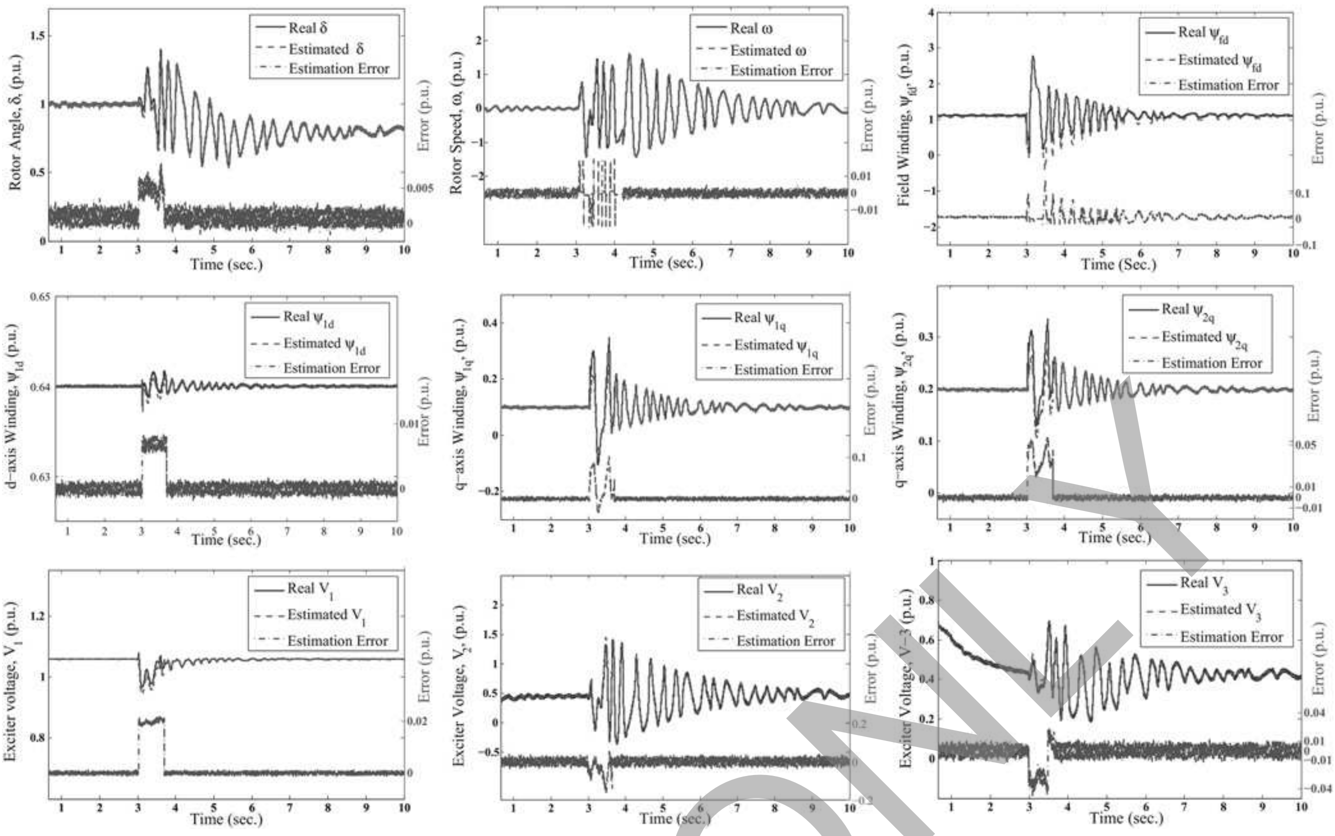

Fig. 7 *Generator state estimation and error of estimation in RJDSE*

The simulation results were followed by BDI process employing normalised residuals test as follows [48]

$$r_i^{\mathrm{N}} = \frac{|r_i|}{s_{ii}} \le \kappa, \tag{16}$$

where r_i^{N} is the largest normalised residual and σ_{ii} is the standard deviation of the *i*th component of the residual vector. The largest normalised residuals with threshold of $\kappa = 3$ were considered as bad data. Here the bad data refers to measurements with gross errors. Once bad data is identified corresponding measurement was updated by deducting gross error $(R_{ii}/\sigma_{ii} r_i)$ from bad data. Using the updated measurements state estimation was repeated only for the subsystems which were affected by bad data. For large-scale systems, this localisation of bad data can save lots of time which can in turn accelerate the state estimation process.

4.4 Complexity analysis and speedup

To demonstrate the performance of the proposed RJDSE in terms of speed-up, the test systems described in Table 1 were used to perform simulation on Tesla™S2050 server from NVIDIA® with 4 GPUs, and 448 cores in each unit of GPU. This device contains 14 streaming multiprocessors, each with 32 streaming processors, an instruction unit, and on-chip memory [33]. CUDA version 5.0 with compute capability 2.0 is used for programming. The CPU is the quad-core Intel® Xeon™ E5-2620 with 2.0 GHz core clock and 32 GB memory, running 64-bit Windows 7® operating system.

Generally, when a system with *N* buses is partitioned into *M* subsystems, each subdomain has approximately *N*/*M* buses. Assume that solving a linear system with iterative method has complexity of $O(N^{\alpha})$ where $\alpha \ge 1$. Using the domain decomposition technique, the complexity of solving each subsystem is $O((N/M)^{\alpha})$ which results in the complexity of $O((N)^{\alpha}/(M)^{\alpha-1})$ for the entire system. It may not be realistic to expect the same speedup in practice; however, the results clearly indicate the advantages of domain decomposition in accelerating DSE. As can be seen from Table 1 and Fig. 8 the percentage of required execution times increases faster in single-GPU ($T_{\mathrm{S.GPU}}^{\mathrm{Ex.}}$) simulation compared with multi-GPU ($T_{\mathrm{M.GPU}}^{\mathrm{Ex.}}$) implementation

Table 1 Execution time in CPU-only, single-GPU, and multi-GPU implementation

Case	No. of buses	No. of gen.	$T_{\mathrm{CPU}}^{\mathrm{Ex.}}$, s	$T_{\mathrm{S.GPU}}^{\mathrm{Ex.}}$, s	$T_{\mathrm{M.GPU}}^{\mathrm{Ex.}}$, s
1	39	10	0.6	0.4	0.4
2	78	20	1.9	1.1	0.9
3	156	40	4.8	2.16	1.45
4	312	80	21.1	6.9	4.3
5	624	160	45.2	12.6	7.3
6	1248	320	95.3	28.9	15.3
7	2496	640	290	43.1	22.2
8	4992	1280	475	62.8	30.6

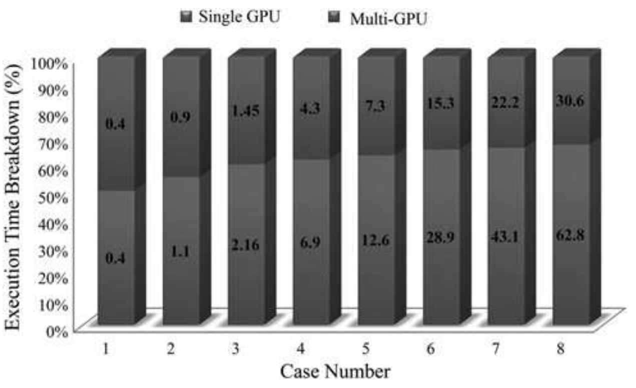

Fig. 8 *Percentage of execution time for varying test cases on single- and multi-GPU simulators*

which shows the higher complexity of this method. The comparison between CPU-only execution time ($T_{\mathrm{CPU}}^{\mathrm{Ex.}}$) and GPU execution time also verify that exploiting parallelism using GPUs can results in significant acceleration in the DSE process. It should also be noted that a better PMU placement by reducing the size of the problem and data transfer may also result in faster state estimation.

5 Conclusions

This paper presented a parallel relaxation-based joint state estimation approach for DSE of power systems. Using the traditional SE method, the size and computational cost of the simulator is usually prohibitive, especially for simulating large-scale systems. As the size of the system increases the amount of data collected in the network grows exponentially which leads to a slow and computationally expensive state estimation process. The motivation behind this work is to test the feasibility of a fully parallel method that could alleviate these limitations.

The proposed methodology: (i) uses only local measurements for DSE in each subsystem which reduce the size of the problem and leads to major reduction in data communication; (ii) takes advantage of massive multi-GPU parallelisation which reduces the execution time significantly; (iii) can be applied to any dynamic system regardless of complication and the type of states or parameters that are needed to be estimated; (iv) localises the effects of bad data to subsystems; (v) does not require either local observability or a central coordinator.

Result comparisons verified the accuracy and efficiency of the proposed method. In addition, the performance of the slow coherency method as the partitioning tool was analysed, and it was concluded that for different fault locations in the system, results derived from this method had lower amounts of error.

The proposed method is general and extensible to any number of GPUs connected in a cluster. Results show that more GPUs can reduce expected computation time.

6 Acknowledgment

Financial support from the Natural Science and Engineering Research Council of Canada (NSERC) is gratefully acknowledged.

7 References

1 Valverde, G., Terzija, V.: 'Unscented kalman filter for power system dynamic state estimation', *IET Gener. Transm. Distrib.*, 2011, **5**, (1), pp. 29–37

2 Queiroz, C., Mahmood, A., Tari, Z.: 'SCADASim—a framework for building SCADA simulations', *IEEE Trans. Smart Grid*, 2011, **2**, (4), pp. 589–597

3 Green, R.C., Wang, L., Alam, M.: 'Applications and trends of high performance computing for electric power systems: focusing on smart grid', *IEEE Trans. Smart Grid*, 2013, **4**, (2), pp. 922–931

4 Chakrabarti, S., Kyriakides, E., Ledwich, G., *et al.*: 'Inclusion of PMU current phasor measurements in a power system state estimator', *IET Gener. Transm. Distrib.*, 2010, **4**, (10), pp. 1104–1115

5 Korres, G.N., Contaxis, G.C.: 'Application of a reduced model to a distributed state estimator'. Proc. IEEE Power and Engineering Society, 2000, vol. 2, pp. 999–1004

6 Zhang, J., Welch, G., Bishop, G.: 'LoDiM: a novel power system state estimation method with dynamic measurement selection'. Proc. IEEE Power and Engineering Society, July 2011, pp. 1–7

7 Seshadri Sravan Kumar, V., Dhadbanjan, T.: 'State estimation in power systems using linear model infinity norm-based trust region approach', *IET Gener. Transm. Distrib.*, 2013, **7**, (5), pp. 500–510

8 Zhao, L., Abur, A.: 'Multiarea state estimation using synchronized phasor measurements', *IEEE Trans. Power Syst.*, 2005, **20**, (2), pp. 611–617

9 Gómez-Expósito, A., de la Villa Jaén, A.: 'Two-level state estimation with local measurement pre-processing', *IEEE Trans. Power Syst.*, 2009, **24**, (2), pp. 676–684

10 Risso, M., Rubiales, A.J., Lotito, P.A.: 'Hybrid method for power system state estimation', *IEEE Trans. Power Appl. Syst.*, 2015, **9**, (7), pp. 636–643

11 Jiang, W., Vittal, V., Heydt, G.T.: 'Diakoptic state estimation using phasor measurement units', *IEEE Trans. Power Syst.*, 2008, **23**, (4), pp. 1580–1589

12 Korres, G.N.: 'A distributed multiarea state estimation', *IEEE Trans. Power Syst.*, 2011, **26**, (1), pp. 73–84

13 Xie, L., Choi, D.H., Kar, S., *et al.*: 'Fully distributed state estimation for wide-area monitoring systems', *IEEE Trans. Smart Grid*, 2012, **3**, (3), pp. 1154–1169

14 Kekatos, V., Giannakis, G.: 'Distributed robust power system state estimation', *IEEE Trans. Power Syst.*, 2013, **28**, (2), pp. 1617–1626

15 Abur, A., Tapadiy, P.: 'Parallel state estimation using multiprocessors', *Electr. Power Syst. Res.*, 1990, **18**, (1), pp. 67–73

16 Falcao, D.M., Wu, F.F., Murphy, L.: 'Parallel and distributed state estimation', *IEEE Trans. Power Syst.*, 1995, **10**, (2), pp. 724–730

17 Mandal, J.K., Sinha, A.K., Roy, L.: 'Incorporating nonlinearity of measurement function in power system dynamic state estimation', *IET Gener. Transm. Distrib.*, 1995, **142**, (3), pp. 289–296

18 Shih, K.R., Huang, S.J.: 'Application of a robust algorithm for dynamic state estimation of a power system', *IEEE Trans. Power Syst.*, 2002, **17**, (1), pp. 141–147

19 Caro, E., Conejo, A.J.: 'State estimation via mathematical programming: a comparison of different estimation algorithms', *IET Gener. Transm. Distrib.*, 2012, **6**, (6), pp. 545–553

20 Aminifar, F., Shahidehpour, M., Fotuhi-Firuzabad, M., *et al.*: 'Power system dynamic state estimation with synchronized phasor measurements', *IEEE Trans. Instrum. Meas.*, 2014, **63**, (2), pp. 352–363

21 Ghahremani, E., Kamwa, I.: 'Dynamic state estimation in power system by applying the extended Kalman filter with unknown inputs to phasor measurements', *IEEE Trans. Power Syst.*, 2011, **11**, (4), pp. 655–661

22 Wang, S., Gao, W., Meliopoulos, A.P.S.: 'An alternative method for power system dynamic state estimation based on unscented transform', *IEEE Trans. Power Syst.*, 2012, **27**, (2), pp. 942–950

23 Zhang, J., Welch, G., Bishop, G., *et al.*: 'A two-stage Kalman filter approach for robust and real-time power system state estimation', *IEEE Trans. Sust. Energy*, 2014, **5**, (2), pp. 629–636

24 Zhou, N., Meng, D., Huang, Z., *et al.*: 'Dynamic state estimation of a synchronous machine using PMU data: a comparative study', *IEEE Trans. Smart Grid*, 2015, **6**, (1), pp. 450–460

25 Happ, H.H., Pottle, C., Wirgau, K.A.: 'An assessment of computer technology for large scale power system simulation'. Proc. Power Industry Computer Application Conf., May 1979, pp. 316–324

26 Brasch, F.M., Van Ness, J.E., Kang, S.C.: 'Simulation of a multiprocessor network for power system problems', *IEEE Trans. Power Appl. Syst.*, 1982, **101**, (2), pp. 295–301

27 Jalili-Marandi, V., Dinavahi, V.: 'SIMD-based large-scale transient stability simulation on the graphics processing unit', *IEEE Trans. Power Syst.*, 2010, **25**, (3), pp. 1589–1599

28 Jalili-Marandi, V., Zhou, Z., Dinavahi, V.: 'Large-scale transient stability simulation of electrical power systems on parallel GPUs', *IEEE Trans. Parall. Distrib. Syst.*, 2012, **23**, (7), pp. 1255–1266

29 Zhou, Z., Dinavahi, V.: 'Parallel massive-thread electromagnetic transient simulation on G- PU', *IEEE Trans. Power Deliv.*, 2014, **29**, (3), pp. 1045–1053

30 Li, Z., Donde, V.D., Tournier, J.C., *et al.*: 'On limitations of traditional multi-core and potential of many-core processing architectures for sparse linear solvers used in large-scale power system applications'. Proc. IEEE Power and Engineering Society, July 2011, pp. 1–8

31 White, J.K., Sangiovani-Vincentelli, A.L.: 'Relaxation techniques for the simulation of VLSI circuits' (Kluwer, Boston, MA, 1987)

32 Crow, M.L., Ilic, M.: 'The parallel implementation of the waveform relaxation method for transient stability simulations', *IEEE Trans. Power Syst.*, 1990, **5**, (3), pp. 922–932

33 Xie, L., Choi, D.H., Kar, S.: 'Cooperative distributed state estimation: local observability relaxed'. Proc. IEEE Power and Engineering Society, July 2011, pp. 1–11

34 Weng, Y., Li, Q., Negi, R., *et al.*: 'Distributed algorithm for SDP state estimation'. Proc. Innovative Smart Grid Technologies, February 2013, pp. 1–6

35 Caro, E., Singh, R., Pal, B.C., *et al.*: 'Participation factor approach for phasor measurement unit placement in power system state estimation', *IET Gener. Transm. Distrib.*, 2012, **6**, (9), pp. 922–929

36 Korres, G.N., Manousakis, N.M.: 'State estimation and observability analysis for phasor measurement unit measured systems', *IET Gener. Transm. Distrib.*, 2012, **6**, (9), pp. 902–913

37 Azizi, S., Gharehpetian, G.B., Dobakhshari, A.S.: 'Optimal integration of phasor measurement units in power systems considering conventional measurements', *IEEE Trans. Smart Grid*, 2013, **4**, (2), pp. 1113–1121

38 Kundur, P.: 'Power system stability and control' (McGraw-Hill, 1994)

39 IEEE Std 421.5–2005: 'IEEE recommended practice for excitation system models for power system stability studies'. Proc. IEEE Power Engineering Society, 2006, pp. 1–85

40 Abur, A., Gómez-Expósito, A.: 'Power system state estimation theory and implementation' (Marcel Dekker, Inc., 2004) pp. 20–25

41 Azizi, S., Dobakhshari, A.S., Nezam Sarmadi, S.A., *et al.*: 'Optimal PMU placement by an equivalent linear formulation for exhaustive search', *IEEE Trans. Smart Grid*, 2012, **3**, (1), pp. 174–182

42 Gómez-Expósito, A., Abur, A.: 'Generalized observability analysis and measurement classification', *IEEE Trans. Power Syst.*, 1998, **13**, (3), pp. 1090–1096

43 Leondes, C.T.: 'Control and dynamic systems v42: analysis and control system' (Academic press Inc., 1991)

44 You, H., Vittal, V., Wang, X.: 'Slow coherency-based islanding', *IEEE Trans. Power Syst.*, 2004, **19**, (1), pp. 483–491

45 NVIDIA: 'CUSPARSE library', NVIDIA Developer, February 2013

46 NVIDIA: 'CUBLAS library', NVIDIA Developer, August 2014

47 IEEE Std C37.118.1: 'IEEE standard for synchrophasor measurements for power systems', December 2011, pp. 1–61

48 Monticelli, A., Garcia, A.: 'Reliable bad data processing for real-time state estimation', *IEEE Trans. Power Appl. Syst.*, 1983, **102**, (5), pp. 1126–1139