Electrical Power and Energy Systems 141 (2022) 108163

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

# A parallel and asynchronous state estimation for coupled transmission-distribution networks

Yingqi Tang, Kunjie Tang, Shufeng Dong *

*College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China*

ARTICLE INFO

*Keywords:*
Asynchronous state estimation
Communication delays
Parallel mechanism
Coupled transmission-distribution networks
Sensitivity analysis

ABSTRACT

Considering the increasing scale of power systems and gradually coupling between transmission networks (TNs) and distribution networks (DNs), a parallel and asynchronous state estimation (par&async-SE) algorithm is proposed to deal with communication delays. First, the coupled transmission-distribution network state estimation problem is reformulated in line with engineering practice. Then, a TN broadcast transmitting strategy, a synchronous preprocessing technique, and a receive-compute-transmit parallel mechanism are applied to improve conventional asynchronous algorithm. Moreover, a 'key DN' concept is introduced and a sensitivity analysis method is proposed to realize key DNs selection. These improvements make the par&async-SE algorithm proposed not only have a higher convergence rate and efficiency but also be less sensitive to parameter settings. Numerical experiments demonstrate that the par&async-SE algorithm has the same accuracy as conventional algorithms. It can also achieve better convergence, reduce computation time, and be less sensitive to parameter settings than the synchronous state estimation algorithm and conventional asynchronous state estimation algorithm when communication delays exist, so that parameter tuning for this self-adaptive algorithm is not necessary in engineering practice. Besides, the par&async-SE algorithm also has bad data robustness when choosing appropriate methods to solve extended-TNSE and DNSE subproblems.

## 1. Introduction

Due to power grids expansion and increasing penetration of renewables, transmission networks (TNs) and distribution networks (DNs) are gradually coupled. However, these two networks are normally operated independently, which makes that models and data of energy management systems in transmission control centers [1] and distribution management systems in distribution control centers [2] cannot be fully shared. This situation leads to difficulties in global situation awareness, operation, dispatch and control [3]. Researches about coupled transmission-distribution (CTD) network analysis have been done in various areas, such as power flow calculation [4-7], optimal power flow [8-11], state estimation (SE) [12-14], economic dispatch [15,16], etc. Among them, SE is a fundamental tool in energy management systems (EMSs) and distribution management systems (DMSs) in practice. By estimating voltage magnitude and angle for all nodes given a set of measurements [13], SE can provide useful information for energy trading, economic dispatch, security analysis, etc. These applications can further ensure the security and stability of the power system.

Many researchers have obtained abundant achievements on the improvement of conventional SE [17-20]. When referring to the field of CTD network analysis, there are also some preliminary achievements on coupled transmission-distribution state estimation (CTDSE). A simple and direct way to deal with CTDSE problem is decoupling it into several SE problems and solving them separately, where exterior networks are considered as simplified equivalent networks for each local SE problem [21]. But in fact, TNs and DNs are coupled and the mutual influence between them cannot be ignored. Particularly, the penetration of distributed generations in DNs leads to bi-directional power flow between TNs and DNs. In this case, the conventional decoupled methods may lead to inaccurate solutions and power mismatches at boundaries. Therefore, it is necessary to carry out global SE for the CTD networks. However, data barriers and low computational efficiency may become new bottlenecks for actual practice, and some algorithms are proposed to solve the static, time-invariant SE for large-scale electric power systems [22]. To get more accurate solutions than the decoupled methods and to overcome bottlenecks of computational efficiency caused by the increasing scale of power systems along with data exchange between different operators, seeking an interactive method to solve SE for CTD network is an issue worth studying. In this way, solving CTDSE problem requires the cooperation of TNs and DNs. References [12] and [14]

* Corresponding author.
*E-mail address:* dongshufeng@zju.edu.cn (S. Dong).

https://doi.org/10.1016/j.ijepes.2022.108163
Received 11 December 2021; Received in revised form 30 January 2022; Accepted 15 February 2022
Available online 14 April 2022

**Nomenclature**

*A. Superscripts, Subscripts, and Embellishments*

| | |
|---|---|
| ET | Extended transmission network |
| T | Original transmission network |
| D | Distribution network |
| D*i* | The *i*-th distribution network |
| recv | The reception thread |
| cal | The calculation thread |
| send | The transmission thread |
| *k* | The *k*-th iteration |
| *i*, *j* | The *i*/*j*-th distribution network |
| (*a*) | Phase *a* |
| (*b*) | Phase *b* |
| (*c*) | Phase *c* |

*B. Sets, Variables and Parameters*

| | |
|---|---|
| $J(.)$ | Objective function of state estimation |
| $z$ | Measurement vector |
| $x$ | State vector |
| $h(.)$ | Vector of functions relating the measured quantities to the state variables |
| $R$ | Measurement error covariance matrix |
| $H$ | System Jacobian matrix |
| $r_i$ | The root bus of *i*-th DN |
| $b_i$ | The bus in TN which *i*-th DN is connected to |
| $l_{bi,ri}$ | The transmission line connecting $b_i$ and $r_i$ |
| $S_{ti}$ | The measurement of power flows through line $l_{bi,ri}$ to bus $r_i$ |
| $S_{fi}$ | The measurement of power flows through the line started from bus $b_i$ |
| $V_r$ | The measurement of voltage magnitude |
| $S_g^{eq}$,$S_d^{eq}$ | Two kinds of equivalent measurement |
| $n$ | Number of DNs |
| $k$ | Current count of iteration |
| $k_0$ | Maximum count of synchronous iteration |
| $K_0$ | A function for tuning $k_0$ |
| $s_i(.)$ | The iteration count of the TN when the updated data of *i*-th DN are last used |
| $S$ | The minimum count of updated data from DNs when extended-TNSE is proceeded |
| $\tau$ | The iteration count that the TN should use updated data of DN at least once |
| $\tau_i^*$ | The iteration count of the TN since the updated data of *i*-th DN are last used |
| $\overline{\tau}$ | The iteration count that the TN should use updated data of a non-key DN at least once |
| $u$ | Index of one key DN obtained through self-sensitivity analysis |
| $v$,$w$ | Index of two key DNs obtained through mutual-sensitivity analysis |
| $\Phi$ | A collection of DNs whose data can reach TN timely |
| $\varepsilon$ | Convergence tolerance |
| $\Theta$ | A mapping describing the iteration process |
| $\Theta_B$ | A composition mapping describing the fixed-point iteration process |
| $T$ | Processing thread of TN and DN |
| $\mu$ | Mean of normal distribution |
| $\sigma$ | Standard deviation of normal distribution |
| $N$ | Normal distribution |

introduce a formulation of CTDSE problem and a synchronous state estimation (sync-SE) algorithm for processing interactive global SE on CTD systems. The issue about dealing with boundary nodes is unclear, causing difficulties in algorithm designing and incapability of adapting to actual conditions.

Communication condition is another important issue in distributed SE. Reference [23] and [24] have taken communication conditions into consideration in the field of distributed optimal power flow problem, indicating that the communication delays of data transmitting between different system operators may have a great influence on the efficiency and convergence of the conventional algorithm. However, in some preliminary researches of distributed SE, the lack of quantitative analysis of communication delays is a shortcoming, such as reference [25]. Reference [26] assumes that the time for communication at each iteration is on the negligible nanosecond level, which cannot meet the actual operation requirement. Reference [27] supposes that when facing a loss of measurements caused by communication interruptions, the correlation coefficient among the measurements can be used to fill up missing values. In [28], the outputs from agents are considered to be timeout after a pre-set time by the coordinator, named as timeout strategy. Moreover, a weight decrease strategy reducing the weight of outputs from timeout agents is also introduced. These two strategies cause that updates from some agents cannot be used in time, resulting in a large calculation error. References [29] and [30] investigate the distributed dynamic state estimation considering packet losses, focusing on handling delays of the measurements in a certain system instead of large-scale coupled power systems. These studies are mainly interested in multi-area distributed analysis while the data barrier in CTD networks is not fully considered.

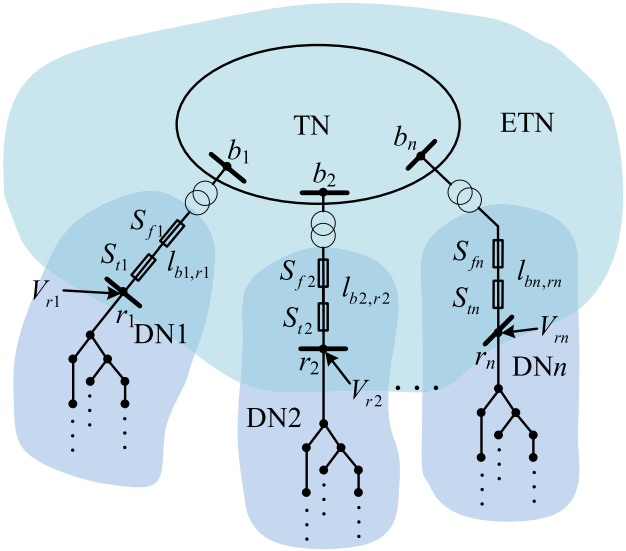


**Fig. 1.** A CTD network with measurements of boundary elements.

Please note that the communication delays mentioned in this paper mean the data transmitting delays between TNs and DNs, instead of the data acquiring delays of the measurements. The data collection rate of measurements is high enough indeed, but the data barrier in CTD systems is a bottleneck. It means that different system operators cannot exchange data timely with each other. When a SE analysis is triggered or executed periodically, all the data required need to be exchanged at once to ensure that the data used are latest. This exchange process may be affected by the communication delays.

When the distributed mechanism is introduced, the algorithm may confront convergence issues. In [26], when the iteration count raises to 30000, the averaged relative error of all areas can be less than 10%. In

[31], the differences of bus phase angles between distributed and centralized algorithms are small enough only when the iteration count comes to a hundred level. In [25], after the iteration count reaches 30 to 50, the distributed algorithm is able to have the same precision as the centralized algorithm. Reference [32] provides an idea about conducting asynchronous distributed SE and an asynchronous convergence detecting method is proposed. These papers do not consider communication delays when dealing with distributed and convergence issues, leading to doubt about the applicability of methods in real-world operation.

To deal with the communication conditions and convergence issues in CTDSE, this paper proposes a parallel and asynchronous state estimation (par&async-SE) algorithm for CTD networks. Several cases are simulated to demonstrate the effectiveness of the algorithm proposed. This paper offers the following distinguishing contributions:

(i) It reformulates the CTDSE model to adapt it to practice. Some previous researches [12,33] divide the measurement sets into transmission, distribution and zero injection at boundary nodes. However, some measurements of power flow at tie lines between TN and DNs are usually equipped in actual engineering conditions. We considered the measurements of power flows at tie lines between TN and DNs, and Fig. 1 in the paper shows a CTD network with measurements of boundary elements equipped. The model of CTD networks is more in line with the actual situation. Line power flow measurements in the real world have the corresponding relation with the variables in the formula expressed.

(ii) It proposes a novel parallel and asynchronous state estimation method in the framework of asynchronous communication. Section 3 presents the conventional asynchronous method and implementation of the par&async-SE with several improvements based on the async-SE algorithm. The improvements include TN broadcast transmitting strategy, synchronous preprocessing technique, receive-compute-transmit parallel mechanism and they can help the algorithm achieve better convergence and reduce computation time. These improvements are described in Section 3.2 with Fig. 3 containing the limitations of conventional methods, principles and effects of each improvement.

(iii) It introduces a 'key distribution network' (key DN) concept and uses a sensitivity analysis method to obtain key DN, which decreases sensitivity to parameters of conventional async-SE. A self-adaptive parameter tuning strategy is also proposed based on the sensitivity analysis. The introduction of key DN is for identifying different impacts of DNs on the asynchronous algorithm. The key DN obtained in self-sensitivity analysis has the greatest impact on the whole calculation process. Thus, data updates from this key DN should be brought to the forefront in the extended-TNSE subproblem. The key DNs obtained in the mutual-sensitivity analysis have the second-largest impact on the whole problem and they should also be carefully considered.

Numerical experiments demonstrate that the par&async-SE algorithm can reduce computation time, achieve a higher convergence rate, and be less sensitive to parameter settings compared with sync-SE and conventional async-SE algorithms when facing communication delays. Also, the par&async-SE algorithm is robust to the occurrence of bad data and gross errors.

The rest of this paper is organized as follows: Section 2 presents the reformulation of the CTDSE problem and conventional synchronous algorithm. Section 3 presents the implementation of the par&async-SE with several improvements. Then, numerical experiments are presented in Section 4. Finally, conclusions are drawn in Section 5.

## 2. CTDSE problem reformulation and synchronous algorithm

This section presents the reformulation of CTDSE problem and the synchronous solution. In specific, Section 2.1 presents the reformulation of the problem. Section 2.2 presents the conventional sync-SE algorithm for the CTDSE problem in Section 2.1.

### 2.1. CTDSE problem reformulation

As to a specific power system, the weighted-least-square (WLS) solution [34,35] of SE can be obtained by solving the following optimization problem:

$$\min_{x} J(x) = [z - h(x)]^T R^{-1} [z - h(x)] \tag{1}$$

where $z$: vector of measured quantities; $x$: vector of state variables; $h(.)$: vector of functions relating the measured quantities to the state variables; $R$: diagonal covariance matrix and the reverse of it is playing as the weight factors.

As shown in Fig. 1, a CTD network contains one TN and $n$ DNs, and model sharing between TN and DNs is not feasible. The root bus of DN $i$ is denoted as $r_i$, and the bus in TN which DN $i$ is connected to is denoted as $b_i$. The transmission line connecting buses $b_i$ and $r_i$ is denoted as $l_{bi,ri}$. Based on the decoupled model in reference [12] and [33], along with some equivalent measurements introduced by this paper, the whole CTD network can be decoupled into an extended transmission network (extended-TN) and $n$ DNs. The measurement of power flows through line $l_{bi,ri}$ to bus $r_i$ is denoted as $S_{ti}$ and that flows through the line started from bus $b_i$ is denoted as $S_{fi}$. The measurement of voltage magnitude of bus $r_i$ is denoted as $V_{ri}$. By introducing two kinds of equivalent measurement named $S_{gi}^{eq}$ and $S_{di}^{eq}$, the optimization problem (1) can be decoupled into (4) and (5). Meanwhile, the equivalent measurements are set as equation (2) and (3):

$$S_{gi}^{eq} = S_{ti} \tag{2}$$

$$S_{di}^{eq} = -S_{fi} \tag{3}$$

$$\min_{x_{ET}} J_{ET}(x_{ET}) = [z_{ET} - h_{ET}(x_{ET})]^T R_{ET}^{-1} [z_{ET} - h_{ET}(x_{ET})] \tag{4}$$

$$\min_{x_D} J_D(x_D) = [z_D - h_D(x_D)]^T R_D^{-1} [z_D - h_D(x_D)] \tag{5}$$

where subscript $ET$, $T$, and $D$ represent variables in extended-TN, original TN and DN;$z_{ET} = \begin{bmatrix} z_T & S_d^{eq} & V_r \end{bmatrix}^T$; $z_D$ contains measurements $S_{gi}^{eq}$ and $V_{ri}$ of $r_i$; $x_{ET}$ contains state variables of $r_i$.

The optimality conditions of (4), (5) are equations (6) and (7).

$$H_{ET}^T(x_{ET}) R_{ET}^{-1} [z_{ET} - h_{ET}(x_{ET})] = 0 \tag{6}$$

$$H_D^T(x_D) R_D^{-1} [z_D - h_D(x_D)] = 0 \tag{7}$$

where $H_{ET}$, $H_D$ are the corresponding Jacobian matrices of $h(.)$.

Equation (6) describes the extended transmission network state estimation (extended-TNSE) subproblem, while equation (7) describes the distribution network state estimation (DNSE) subproblem. In the decoupled form of CTDSE problem, $S_{gi}^{eq}$ and $S_{di}^{eq}$ are regarded as equivalent power injections of bus $r_i$, and can be used to solve extended-TNSE and DNSE subproblem. Two mappings can be established to describe the

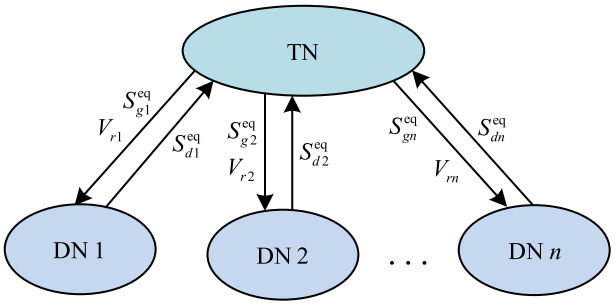


**Fig. 2.** Data flow in sync-SE algorithm.

back-and-forth iterations between TN and DNs:

$$\left[S_{g1}^{eq},\ldots,S_{gi}^{eq},\ldots,S_{gn}^{eq};V_{r1},\ldots,V_{ri},\ldots V_{rn}\right]=\Theta_T[S_{d1}^{eq},\ldots,S_{di}^{eq},\ldots,S_{dn}^{eq}] \tag{8}$$

$$[S_{di}^{eq}]=\Theta_{Di}\left[S_{gi}^{eq};V_{ri}\right] \tag{9}$$

where equation (8) describes the iteration process of extended-TNSE subproblem while equation (9) describes the iteration process of DNSE subproblem. After being abstracted from equations (6) and (7), equations (8) and (9) are describing a general solution process for SE, including WLS SE, weighted-least-absolute-value (WLAV) SE and other SE methods. Given the equivalent load $S_{di}^{eq}$ of $i$-th DN, the extended-TNSE subproblem can be solved by TN, and the equivalent power injection $S_{gi}^{eq}$ along with the voltage magnitude $V_{ri}$ of bus $r_i$ in $i$-th DN can be obtained for a new iteration. Given the equivalent power injection $S_{gi}^{eq}$ along with the voltage magnitude $V_{ri}$ of bus $r_i$, each DN can solve the DNSE subproblem, and the equivalent load $S_{di}^{eq}$ of $i$-th DN can be also acquired for a new iteration. After each back-and-forth iteration, the values of $S_g^{eq}$,$S_d^{eq}$ and $V_r$ are updated.

From the $k$-th iteration to $(k+1)$-th iteration, a composition mapping can be defined as equation (10):

$$\left[S_{g,k+1}^{eq};V_{r,k+1}\right]=\Theta_B\left[S_{g,k}^{eq};V_{r,k}\right] \tag{10}$$

where $\Theta_B=\Theta_T(\Theta_{D1}(\cdot),\Theta_{D2}(\cdot),\ldots,\Theta_{Dn}(\cdot))$; $S_{g,k}^{eq}$ and $V_{r,k}$ are the values of $S_g^{eq}$ and $V_r$ in $k$-th iteration.

Compared to reference [12] and [14], the CTDSE problem reformulated in this paper is more suitable for engineering practice. In the model proposed, the power flow measurements of transmission lines connected to root bus $r_i$ of DN $i$ are treated as bus injection measurements, which is not clear in previous researches. Furthermore, different models can be built according to actual conditions of TN and DN, and the extended-TNSE and DNSE subproblems can be solved by different methods. For instance, single-phase balanced modeling is often used for TN, while three-phase unbalanced modeling is often used for DN. Several appropriate algorithms can be used to solve these models by TN and DN respectively.

### 2.2. Synchronous state estimation algorithm

The data flow in sync-SE algorithm among TN and DNs is shown in Fig. 2. In the conventional sync-SE [12,14], the TN needs to wait for values of $S_d^{eq}$ from all DNs to arrive. The start time of TN's computation is determined by the last value it receives. If any DN has low computational efficiency or bad communication condition, then the solution process may face efficiency problems.

## 3. Par&async-SE algorithm

This section presents the par&async-SE algorithm proposed in this paper. In specific, Section 3.1 introduces the conventional async-SE algorithm. Section 3.2 presents the novel par&async-SE algorithm with several improvements for conventional async-SE. Section 3.3 presents the detailed implementation of par&async-SE algorithm. Section 3.4 performs convergence analysis on the algorithm proposed.

### 3.1. Conventional asynchronous method for CTDSE problem

The iterative solution of CTDSE can be abstracted as a fixed-point problem described by equation (10), and the asynchronous iterative method can also be used to obtain the solution [36]. An asynchronous iteration corresponding to the operator $\Theta_B$ and starting with a given pair

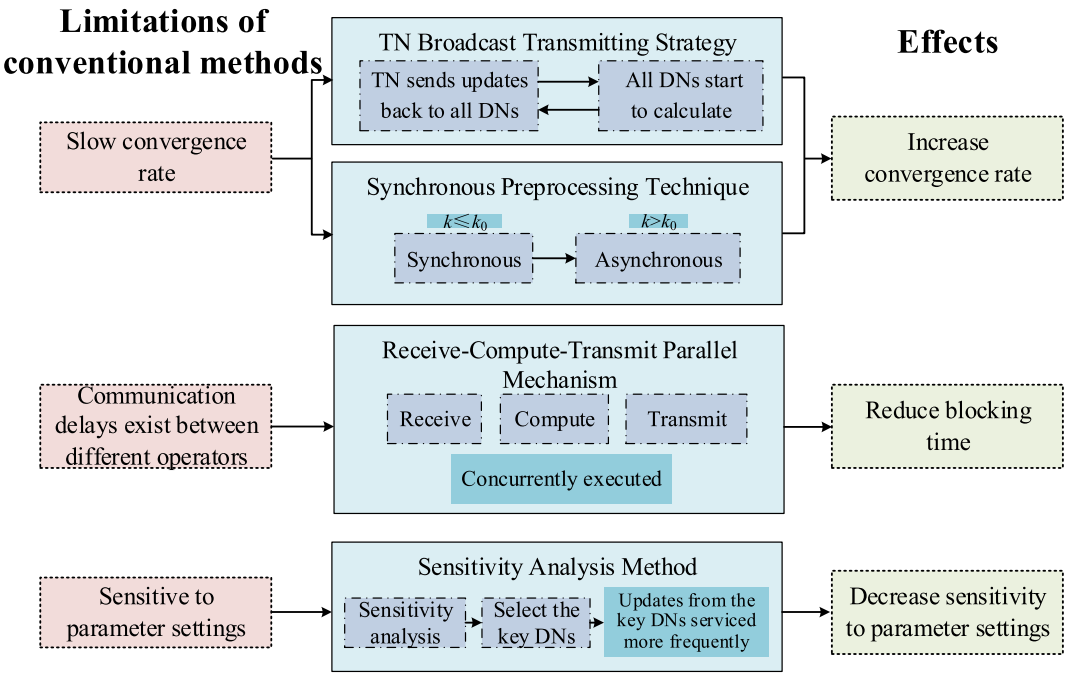


**Fig. 3.** Components of par&async-SE algorithm.

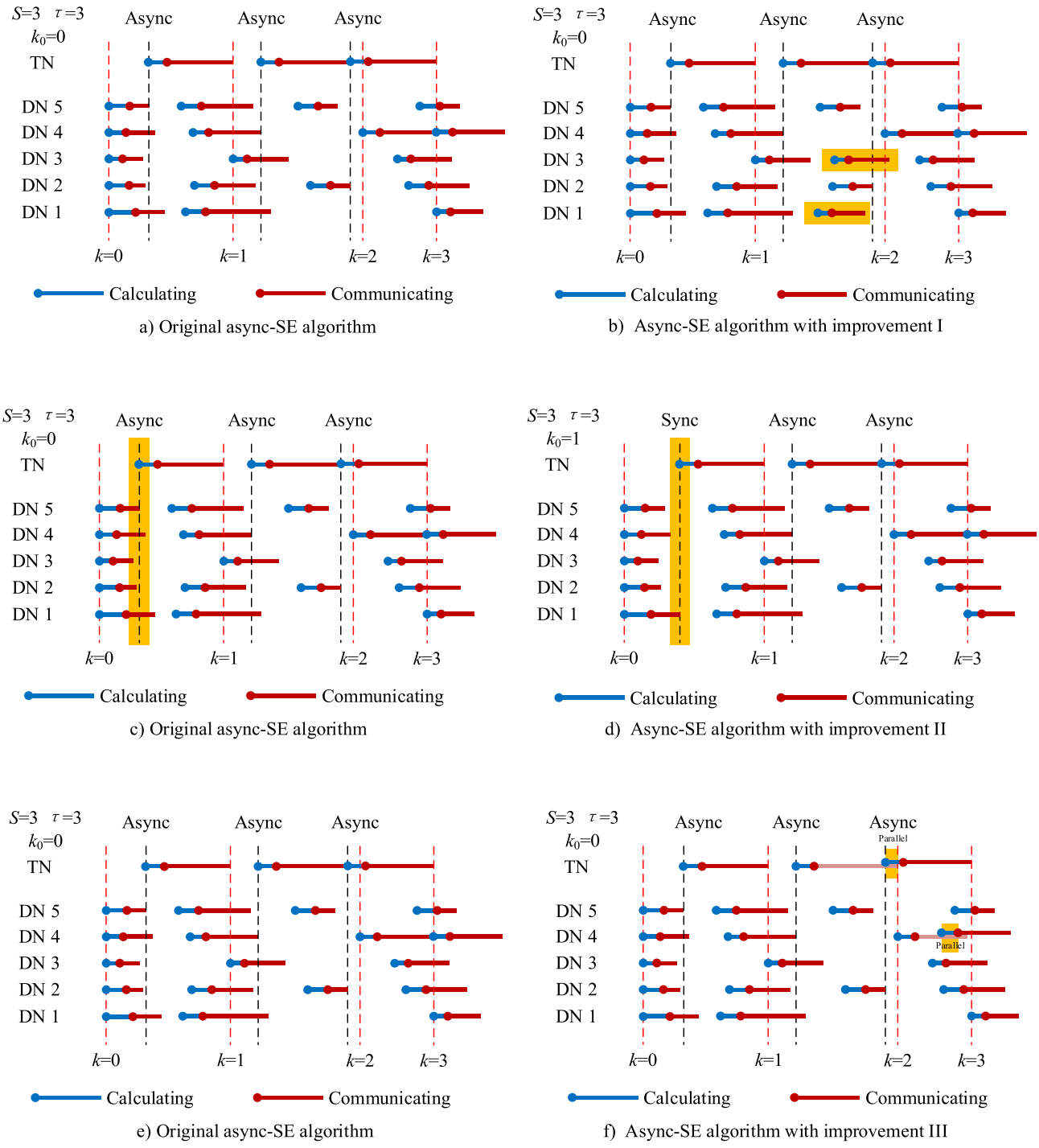


**Fig. 4.** Illustration of the improvements I, II and III.

of vector $\left[S_{g,0}^{eq}; V_{r,0}\right]$ is a sequence $\left[S_{g,k}^{eq}; V_{r,k}\right]$, $k = 0, 1\ldots$, of pairs of vectors defined recursively by:

$$\left[S_{gi,k+1}^{eq}; V_{ri,k+1}\right] = \begin{cases} \left[S_{gi,k}^{eq}; V_{ri,k}\right] & i \notin \Phi_k \\ \Theta_B\left[S_{gi,k}^{eq}; V_{ri,k}\right] & i \in \Phi_k \end{cases} \tag{11}$$

where $\Phi_k$ is a collection of DNs whose data can reach TN timely.

Equation (11) presents the basic iteration form of conventional async-SE algorithm, which is proposed by modifying the sync-SE algorithm [9,37]. To clarify the form of async-SE algorithm in extended-TNSE and DNSE subproblem, equation (11) is decoupled:

$$\left[S_{g,k+1}^{eq}; V_{r,k+1}\right] = \Theta_T\left[S_{d1,s_1(k)}^{eq}, \ldots, S_{di,s_i(k)}^{eq}, \ldots, S_{dn,s_n(k)}^{eq}\right] \tag{12}$$

$$\left[S_{di,s_i(k)}^{eq}\right] = \Theta_{Di}\left[S_{gi,s_i(k)}^{eq}; V_{ri,s_i(k)}\right] \tag{13}$$

where $s_i(k)$ is considered as a function of $k$, tends to infinity as $k$ tends to infinity. Equation (12) and (13) describes the asynchronous iteration form of extended-TNSE and DNSE subproblem respectively.

In the conventional async-SE algorithm, parameters $S$ and $\tau$ are introduced to increase the convergence rate. The following constraints should be satisfied:

$$\begin{cases} k+1-s_i(k) \leqslant \tau \\ \sum_{i=1}^{n} sgn(s_i(k)-k) \leqslant -n+S \\ s_i(k) \leqslant k \end{cases} \tag{14}$$

Parameter $S$ indicates that the TN will not proceed extended-TNSE until receiving updated data from $S$ DNs, shown as the second formula in (14). To ensure convergence, the TN should use updated data of each DN at least once in $\tau$ iterations. Each DN has a variable named $\tau_i^* = k+1-s_i(k)$ to record the iteration of TN since its updated data are last used, shown as the first formula in (14). When $S = n$ or $\tau = 1$, the async-SE degrades to sync-SE. The third formula in (14) is determined by the asynchronous mechanism, which means that the data coming from DNs used by TN is latest ($s_i(k) = k$) or outdated ($s_i(k) < k$).

### 3.2. Several improvements for conventional Async-SE

To further increase the convergence rate and speed up the iteration process, several improvements are applied to conventional async-SE to form the parallel and asynchronous state estimation (par&async-SE) algorithm, as shown in Fig. 3.

I. TN Broadcast Transmitting Strategy

In (14), the second constraint is modified to:

$$\sum_{i=1}^{n} sgn(s_i(k)-k) = -n+S \tag{15}$$

In conventional async-SE, $s_i(k) \leq k$ when $i \in \Phi_k$. This causes that in a certain iteration of the extended-TNSE process, the TN may use outdated data from DNs. These data from DNs may be detrimental to convergence. By setting $s_i(k)$ equal to $k$ when $i \in \Phi_k$, the TN may send the newly calculating results to all DNs in each iteration and all DNs start their calculating thread when they receive data from TN. The TN broadcast transmitting strategy can ensure the data used in the extended-TNSE timely.

II. Synchronous Preprocessing Technique

In (14), two constraints are added forming:

$$\begin{cases} s_i(k) = k, i = 1, 2, \ldots, n, k \leqslant k_0 \\ s_i(k) \geqslant k_0, i = 1, 2, \ldots, n, k > k_0 \end{cases} \tag{16}$$

where $k_0 = K_0\left[S_{g,0}^{eq}; V_{r,0}\right]$, is a non-negative integer.

Synchronous and asynchronous iterations are combined in this technique. For an initial value of $\left[S_{g,0}^{eq}; V_{r,0}\right]$, the TN and DNs synchronize with each other at the beginning of back-and-forth iteration. There exists a parameter $k_0$ which can let $\left[S_{g,k0}^{eq}; V_{r,k0}\right]$ fall into the hypercube where the algorithm can converge. After $k_0$-th iteration, the asynchronous iteration is in place of synchronous iteration.

III. Receive-Compute-Transmit Parallel Mechanism

The reception, calculation and transmission thread of TN are defined as $T_{T,recv}$, $T_{T,cal}$ and $T_{T,send}$, respectively. Those of DN $i$ are defined as $T_{Di,recv}$, $T_{Di,cal}$ and $T_{Di,send}$. These threads can be parallel and have the advantage of concurrent execution. For example, $T_{T,send}$ of the current iteration and $T_{T,cal}$ of next iteration can be executed concurrently. Likewise, $T_{Di,send}$ and $T_{Di,cal}$ can be conducted at the same time when a new value of $\left[S_{g,k}^{eq}; V_{r,k}\right]$ arrives.

$$\begin{cases} k+1-s_i(k) \leqslant \tau \\ \sum_{i=1}^{n} sgn(s_i(k)-k) = -n+S \\ s_i(k) = k, k \leqslant k_0, i = 1, 2, \ldots, n \\ k_0 \leqslant s_i(k) \leqslant k, k > k_0, i = 1, 2, \ldots, n \end{cases} \tag{17}$$

Considering all of the improvements above, constraints should be satisfied as shown in equation (17). Fig. 4 shows how the improvements work respectively. In Fig. 4, $k$ is the iteration count as well as the clock of TN and the horizontal axis is a continuous timeline. With $n = 5$, $S = 3$, $\tau = 3$, and $k_0 = 1$, the first iteration is processed synchronously, which is shown in subfigure c) and d) corresponding to improvement II. After the calculation and communication threads of TN are completed, the iteration count $k$ becomes 1. The TN keeps polling data updates from DNs. After a certain period, updates from DN 2, DN 4 and DN 5 arrive and another extended-TNSE process is executed. Then the iteration count $k$ becomes 2. The TN sends the updated data back to all DNs and they process DNSE by themselves. Among them, DNSE executed by DN 1 and DN 3 is supplementary compared to conventional async-SE, as shown in subfigure a) and b) in yellow relating to the improvement I. When it comes to the third iteration, transmission thread $T_{T,send}$ of the TN hasn't finished and the number of data updates from DNs reaches $S$. Then $T_{T,cal}$ of next iteration is executed simultaneously. Similarly, when the next update from TN arrives and DN 4 hasn't completed its thread $T_{D4,send}$, the $T_{D4,cal}$ of next iteration can be executed concurrently. This is shown in subfigures e) and f) in Fig. 4 and the thread parallelization concerning improvement III is highlighted in yellow.

IV. Sensitivity Analysis Method for Key DNs Selection

The sensitivity index can reflect the sensitivity of equivalent node injection power of bus $r_i$ to the state variable of bus $r_i$. Also, it can show how far DN $i$ is affected when equivalent node injection power of bus $r_j$ in DN $j$ changes. Self-sensitivity of a certain DN $i$ in $k$-th iteration is defined as follow:

$$\left|\frac{\Delta S_{di,k}^{\mathrm{eq}}}{\Delta V_{ri,k}}\right| = \frac{\left|S_{di,k+1}^{\mathrm{eq}} - S_{di,k}^{\mathrm{eq}}\right|}{\left|V_{ri,k+1} - V_{ri,k}\right|}, i = 1, \ldots, n \tag{18}$$

Maximum self-sensitivity of a certain DN $i$ in $k$ iterations is defined as follow:

$$\left|\frac{\Delta S_{di}^{\mathrm{eq}}}{\Delta V_{ri}}\right| = \max_{k}\left(\left|\frac{\Delta S_{di,k}^{\mathrm{eq}}}{\Delta V_{ri,k}}\right|\right) \tag{19}$$

Through self-sensitivity analysis, one key DN can be obtained whose index is recorded as $u$:

$$u = \underset{i}{\mathrm{argmax}}\left(\left|\frac{\Delta S_{di}^{\mathrm{eq}}}{\Delta V_{ri}}\right|\right) \tag{20}$$

Similarly, mutual-sensitivity of DN $i$ to DN $j$ in $k$-th iteration is defined as follow:

$$\left|\frac{\Delta S_{di,k}^{\mathrm{eq}}}{\Delta S_{dj,k}^{\mathrm{eq}}}\right| = \frac{\left|S_{di,k\ +\ 1}^{\mathrm{eq}} - S_{di,k}^{\mathrm{eq}}\right|}{\left|S_{dj,k\ +\ 1}^{\mathrm{eq}} - S_{dj,k}^{\mathrm{eq}}\right|}, i, j = 1, \ldots, n, i \neq j \tag{21}$$

Maximum mutual-sensitivity of DN $i$ to DN $j$ in $k$ iterations is defined as follow:

$$\left|\frac{\Delta S_{di}^{\mathrm{eq}}}{\Delta S_{dj}^{\mathrm{eq}}}\right| = \max_{k}\left(\left|\frac{\Delta S_{di,k}^{\mathrm{eq}}}{\Delta S_{dj,k}^{\mathrm{eq}}}\right|\right) \tag{22}$$

Through mutual-sensitivity analysis, two key DNs can be obtained whose indices are recorded as $v$ and $w$:

$$v, w = \underset{i,j}{\mathrm{argmax}}\left(\left|\frac{\Delta S_{di}^{\mathrm{eq}}}{\Delta S_{dj}^{\mathrm{eq}}}\right|\right) \tag{23}$$

In the sync-SE algorithm, the TN waits for the updates from all DNs and there is no problem with data missing. However, when the asynchronous method is introduced, different impacts of DNs on the whole problem should be taken into consideration. Data updates from key DNs should be used as much as possible. The key DN obtained in self-sensitivity analysis has the greatest impact on the whole calculation process. Thus, data updates from this key DN should be brought to the forefront in the extended-TNSE subproblem. The key DNs obtained in the mutual-sensitivity analysis have the second-largest impact on the whole problem and they should also be carefully considered. Through the sensitivity analysis method for key DNs selection, the performance of the algorithm proposed will be consistent under various parameter settings.

### 3.3. Implementation of Par&async-SE algorithm

In the par&async-SE algorithm proposed, parameter $\tau$ is expanded to be a vector denoted as $\boldsymbol{\tau} = [\tau_1\ \tau_2\ \ldots\ \tau_n]^T$. The parameter $\tau_i$ of key DN $i$ is given as:

$$\tau_u = 1 \tag{24}$$

$$\tau_v = \min(\overline{\tau}, 2), \tau_w = \min(\overline{\tau}, 3) \tag{25}$$

where $\overline{\tau}$ is the parameter of a non-key DN.

Parameters $\tau_u$, $\tau_v$ and $\tau_w$ can be obtained through the sensitivity analysis method performed on the sync-SE algorithm. The specific steps of the par&async-SE algorithm are shown in Algorithm 1–1 and 1–2. $n$ is the number of DNs.

**Algorithm 1**. (*−1 par&async-SE (processed by DN i)*)

```
initialize R_D and z_D.
V_ri and S_gi^eq are set to corresponding measurement values.
let k = 0;
Do in parallel
    Thread T_Di, cal;
    Thread T_Di, send;
    Thread T_Di, recv;
End Do in parallel

Thread T_Di, cal:
    Repeat
        wait;
    Until recently V_ri and S_gi^eq is set by T_Di, recv or k=0;
    iteratively solve non-linear equation (7), update x_D;
    calculate the total load demand and set the value to S_di^eq;
    k = k +1;
    If ΔV_rk+1 = || V_rk+1 − V_rk ||_∞ < ε
        Then terminate thread T_Di, send, T_Di, cal and T_Di, recv;
        Output x_D.
    End If
End Thread

Thread T_Di, send:
    Repeat
        wait;
    Until recently S_di^eq is set by T_Di, cal;
    send S_di^eq to TN;
End Thread

Thread T_Di, recv:
    Repeat
        wait;
    Until receiving the value of V_ri and S_gi^eq from TN;
End Thread
```

**Algorithm 1.** (*−2 par&async-SE (processed by TN)*)

initialize $R_{ET}$, $z_{ET}$, parameter $S$, $\overline{\tau}$, $k_0$ and stop criteria $\varepsilon$;
let iteration count $k = 0$;
load results from sensitivity analysis to set $\boldsymbol{\tau} = [\tau_1\ \tau_2\ \ldots\ \tau_n]^T$;
**Do in parallel**
  **Thread** $\boldsymbol{T}_{\mathrm{T,recv}}$;
  **Thread** $\boldsymbol{T}_{\mathrm{T,cal}}$;
  **Thread** $\boldsymbol{T}_{\mathrm{T,send}}$;
**End Do in parallel**

**Thread** $\boldsymbol{T}_{\mathrm{T,recv}}$**:**
  **If** $k \le k_0$
    **Repeat**
      wait;
    **Until** receiving updates of $S_{di}^{\mathrm{eq}}$ from all DNs
  **Else**
    **Repeat**
      wait;
    **Until** receiving $S$ updates of $S_{di}^{\mathrm{eq}}$ from DNs and satisfy $\max_{\mathrm{DN}\ i \notin \Phi_k} \tau_i^* < \tau_i$;
  **End If**
  put the satisfactory DNs into the collection $\Phi_k$;
**End Thread**

**Thread** $\boldsymbol{T}_{\mathrm{T,cal}}$**:**
  **Repeat**
    wait;
  **Until** $\Phi_k$ is set by $\boldsymbol{T}_{\mathrm{T,recv}}$;
  **For** DN $i \in \Phi_k$ do
    $\tau_i^* = 1$;
    set $S_{di}^{\mathrm{eq}}$ to newly received value from DN $i$;
  **End For**
  **For** DN $i \notin \Phi_k$ do
    $\tau_i^* = \tau_i^* + 1$;
    set $S_{di}^{\mathrm{eq}}$ of DN $i$ using last received data;
  **End For**
  iteratively solve equation (6) to obtain $V_{ri}$ and update $x_{ET}$;
  **If** $\Delta V_{r,k+1} = \| V_{r,k+1} - V_{r,k} \|_\infty < \varepsilon$
    **Then** terminate thread $\boldsymbol{T}_{\mathrm{T,recv}}$, $\boldsymbol{T}_{\mathrm{T,send}}$ and $\boldsymbol{T}_{\mathrm{T,cal}}$;
    **Output** $x_{ET}$.
  **End If**
  calculate the load of boundary bus and set the value to $S_{gi}^{\mathrm{eq}}$;
  $k = k + 1$;
**End Thread**

**Thread** $\boldsymbol{T}_{\mathrm{T,send}}$**:**
  **Repeat**
    wait;
  **Until** recently $V_{ri}$ and $S_{gi}^{\mathrm{eq}}$ is set by $\boldsymbol{T}_{\mathrm{T,cal}}$;
  send $V_{ri}$ and $S_{gi}^{\mathrm{eq}}$ to all DNs;
**End Thread**

Parameter tuning of $S$ and $\tau$ in the conventional async-SE is rough and complicated. If parameters are not properly set, the performance of the algorithm will not be satisfying. Some subsystems with high sensitivity may exert a tremendous influence on the whole calculating process. This problem can be avoided through the par&async-SE algorithm proposed in this paper by utilizing data updates from those high sensitivity subsystems more frequently. Besides, with several improvements of the conventional async-SE, the par&async-SE can further increase convergence rate and reduce blocking time to decrease total time cost.

### 3.4. Convergence analysis

The convergence of the CTDSE problem in this paper is analyzed based on the fixed-point theorem [9]. From the $k$-th iteration to $(k+1)$-th iteration, a composition mapping defined as equation (10) can be expressed as:

$$[d_{k+1}] = \Theta_B[d_k] \tag{26}$$

where $d_k = \left[S_{g,k}^{\mathrm{eq}}; V_{r,k}\right]$.

The sync-SE converges with a linear rate when the spectral radius of the derivative of $\Theta_B$ is less than 1:

$$\left\|[d_{k+1}] - \left[d_k^*\right]\right\|_\infty < \left\|[d_k] - \left[d_k^*\right]\right\|_\infty \tag{27}$$

$$\lim_{k\to\infty} \frac{\left\|[d_{k+1}] - \left[d_k^*\right]\right\|_\infty}{\left\|[d_k] - \left[d_k^*\right]\right\|_\infty} = \left\|\nabla\Theta_B\right\|_\infty < 1 \tag{28}$$

$$\lim_{k\to\infty} \left\|[d_k] - \left[d_k^*\right]\right\|_\infty = 0 \tag{29}$$

where $d_k^*$ is the true value of $d_k$.

The convergence of asynchronous iterative method for fixed-point problems has been previously proved in [36]. Compared to the conventional async-SE algorithm, the par&async-SE algorithm proposed has several improvements. Improvement I can increase the convergence rate. Improvement II can not only relax the requirements of the initial value but also keep the convergence in a large range. It gives full play to the advantages of asynchronous to reduce the synchronization and communication overhead. Improvement III only deals with parallel

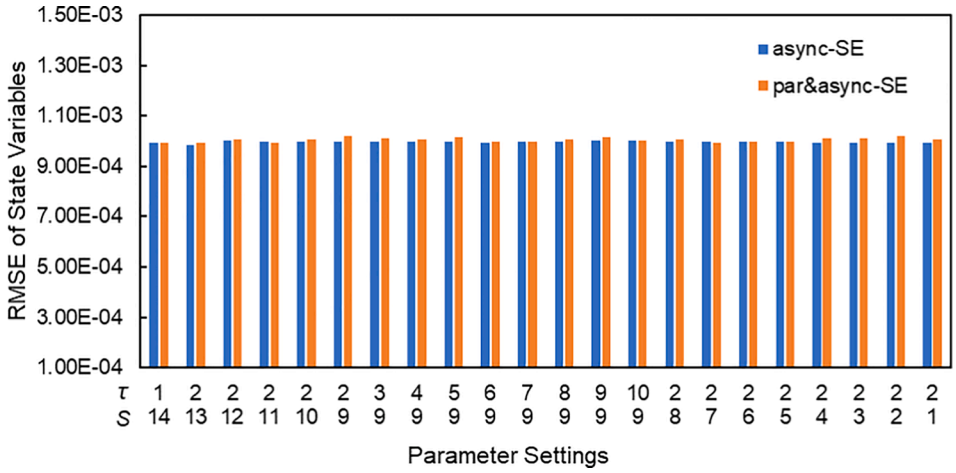


**Fig. 5.** Comparison of the RMSE results among the sync-SE, async-SE, and par&async-SE under situation #1 on TN30DN14.

issues among different threads, which has little impact on convergence. Consequently, the convergence of the par&async-SE algorithm can be guaranteed.

## 4. Discussions and extensions

### 4.1. Discussions about unbalanced DNs

In real-world operation, single-phase balanced modeling is often used for TN, while three-phase unbalanced modeling is often used for DN, as mentioned before. The three-phase imbalance in DN should be carefully considered, and the main challenge to extend the par&async-SE algorithm proposed for a three-phase formulation is in the boundary buses. The measurements of voltage magnitude $V_{ri}$ of bus $r_i$ and two kinds of equivalent measurement named $S_{gi}^{\text{eq}}$ and $S_{di}^{\text{eq}}$ need to be correspondingly adapted to three-phase. Three situations are discussed as follows.

First, consider that the TN and DN are both modeled as three-phase. The algorithm proposed can readily be generalized to a three-phase model and the calculation results are the most accurate. However, this treatment incurs additional computational expense and does not fully exploit the balanced TN.

Second, consider dealing with unbalanced DN by obtaining a single-phase positive sequence equivalent network of three-phase DN via the method in [38]. After the conversion, the TN and the DNs can be connected using the single-phase model for further analysis.

Third, consider that the TN is modeled as single-phase, but the DN is modeled as a three-phase. In specific, for a DN, it is assumed that the three-phase voltage at its root bus $r_i$ is symmetric at each iterative step [4,6,16,39], i.e., the $V_{ri,k}$ and $S_{gi,k}^{\text{eq}}$ received from TN in Algorithm 1-1 can be expressed as (30) and (31), respectively.

$$V_{ri,k}^{(a)} = V_{ri,k}^{(b)} = V_{ri,k}^{(c)} = V_{ri,k} \tag{30}$$

$$S_{gi,k}^{\text{eq}(a)} = S_{gi,k}^{\text{eq}(b)} = S_{gi,k}^{\text{eq}(c)} = \frac{S_{gi,k}^{\text{eq}}}{3} \tag{31}$$

where the superscripts $a$, $b$, and $c$ represent three different phases. For

**Table 1**
Performance Comparison of the Three Algorithms on Different Cases under situation #1.

| Case | Algorithm | Average iteration count | Average time cost (s) | Minimum time cost (s) | Standard deviation of time cost (s) |
|---|---|---|---|---|---|
| TN30DN4 | sync-SE | 8 | 1.73 | 1.73 | – |
| | async-SE | 19.9 | 1.73 | 1.40 | 0.173 |
| | par&async-SE | 8.3 | 1.30 | 1.16 | 0.125 |
| TN30DN14 | sync-SE | 8 | 1.91 | 1.91 | – |
| | async-SE | 20.8 | 2.33 | 1.55 | 0.590 |
| | par&async-SE | 8.0 | 1.73 | 1.41 | 0.192 |
| TN30DN20 | sync-SE | 9 | 2.19 | 2.19 | – |
| | async-SE | 24.3 | 2.75 | 1.88 | 0.734 |
| | par&async-SE | 10.0 | 1.89 | 1.51 | 0.164 |

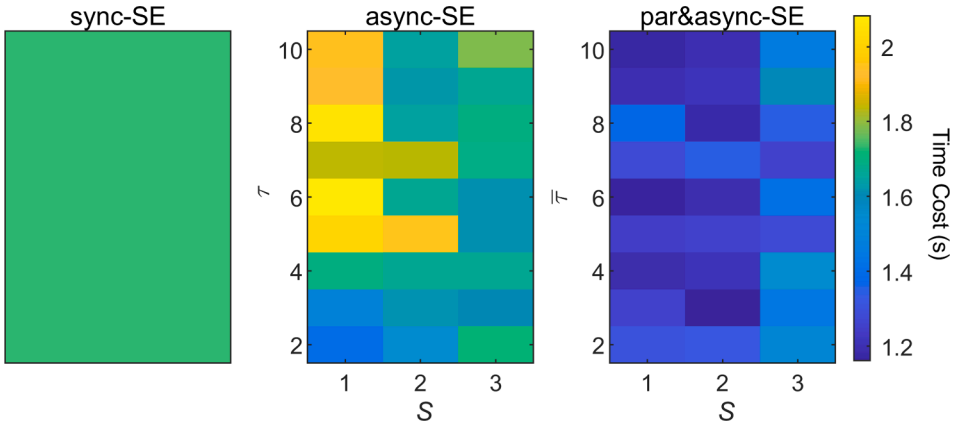


**Fig. 6.** Comparison of the time costs among the sync-SE, async-SE, and par&async-SE under situation #1 on TN30DN4.

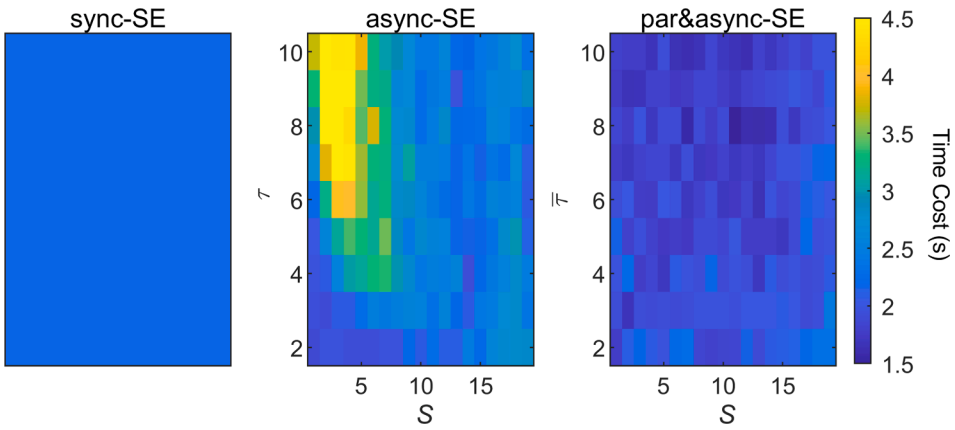


**Fig. 7.** Comparison of the time costs among the sync-SE, async-SE, and par&async-SE under situation #1 on TN30DN20.

the TN, the $S_{di,k}^{eq}$ used by TNSE in $k$-th iteration also needs to be modified, forming as equation (32).

$$S_{di,k}^{eq} = S_{di,k}^{eq(a)} + S_{di,k}^{eq(b)} + S_{di,k}^{eq(c)} \tag{32}$$

where $S_{di,k}^{eq(a)}$, $S_{di,k}^{eq(b)}$ and $S_{di,k}^{eq(c)}$ are the results of three-phase unbalanced DNSE.

### 4.2. Discussions about unobservable DNs

Performing SEs in the TN and DN are different. In a TN, generally, a clear system model and redundant measurements are guaranteed to support observability. However, in the DN, the modeling of the network is not always available. And the DNs are featured with a huge number of nodes but very limited measurements, which makes it unobservable [34,40]. The use of pseudo measurements based on interpolated observations or forecasts from historical data has been a dominant theme for DSSE [41]. With various methods of generating pseudo measurements, such as artificial neural network [42], bayesian network [41], gradient boosting tree models [43], convergent and accuracy solutions can be obtained.

Since this paper mainly focuses on the solving of CTDSE via the interaction of TN and DNs, it's assumed that all TN and DNs are observable after preprocessing. Besides, the CTDSE model reformulated in this paper is abstract. The algorithm proposed is general and different methods can be used to solve the extended-TNSE and DNSE subproblems forming equations (8) and (9), respectively, as which is mentioned in [11]. The iteration process of the algorithm has no relationship with how the SE subproblem is solved, but only provides an overall framework for state estimation for coupled transmission-distribution networks. No matter how the DN solves DNSE problem, it only needs to provide $S_{di,k}^{eq(a)}$, $S_{di,k}^{eq(b)}$ and $S_{di,k}^{eq(c)}$ for TNSE, as shown in equation (9) and (32), and the observability issue of DNSE can be dealt with mature solutions.

## 5. Case analysis

This section validates the effectiveness of the par&async-SE algorithm proposed considering communication delays. The programming language used is MathScript and the version of MATLAB is R2019b. The

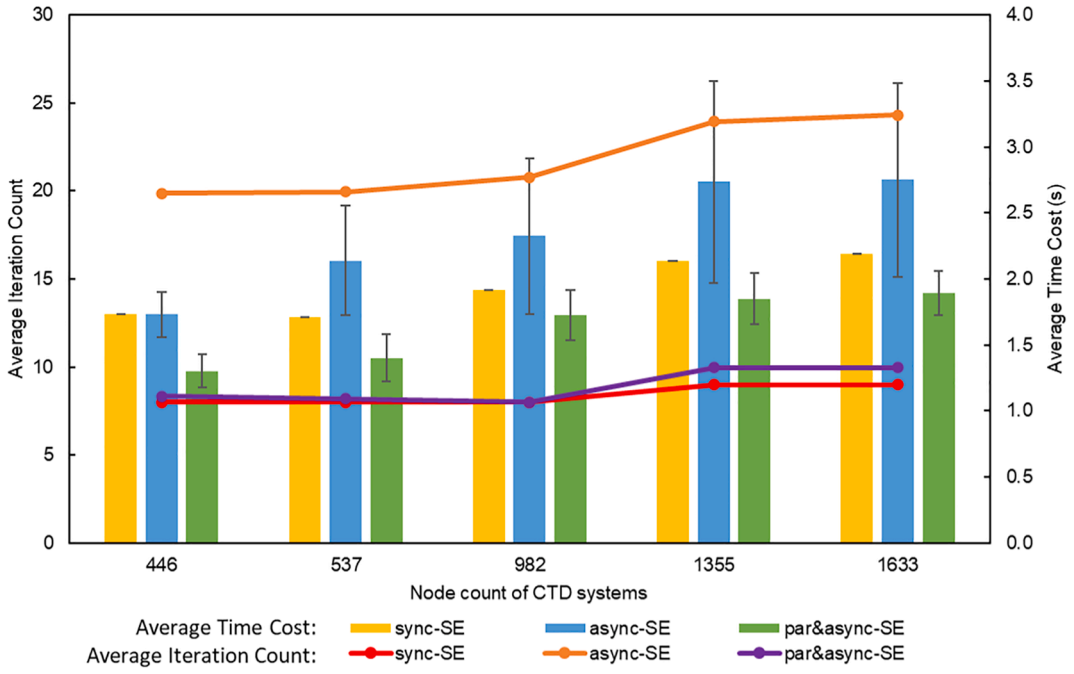


**Fig. 8.** Average iteration count and time cost among the sync-SE, async-SE, and par&async-SE under situation #1 on TN30DNis.

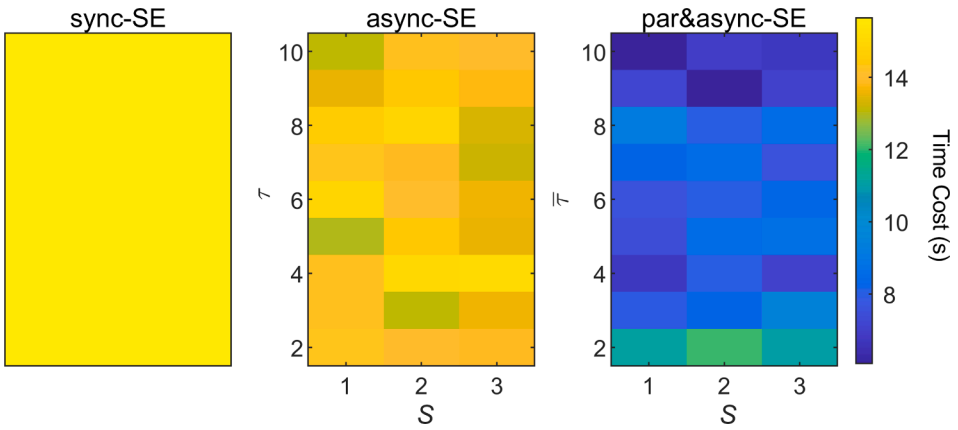


**Fig. 9.** Comparison of the time costs among the sync-SE, async-SE, and par&async-SE under situation #2 on TN30DN4.

operating system used is Windows 10 of 64 bits, with a 3.20 GHz master frequency Intel Core i7-8700 CPU and 15.8G available memory. The communication process is simulated locally, i.e. the calculation processes of TN and DNs run in MATLAB on the same computer without delays in the real network environment. The communication delay is simulated by adding a random delay to the process.

**Table 2**
Performance comparison of the three algorithms on different cases under situation #2.

| Case | Algorithm | Average iteration count | Average time cost (s) | Average percent of time cost reduced compared to sync-SE (%) |
|---|---|---|---|---|
| TN30DN4 | sync-SE | 8 | 15.65 | 0 |
| | async-SE | 36.6 | 14.01 | 10.47 |
| | par&async-SE | 8.4 | 8.17 | 47.79 |
| TN30DN6 | sync-SE | 8 | 16.07 | 0 |
| | async-SE | 36.5 | 14.18 | 11.74 |
| | par&async-SE | 8.1 | 7.65 | 52.39 |
| TN30DN14 | sync-SE | 8 | 16.65 | 0 |
| | async-SE | 32.5 | 14.76 | 11.37 |
| | par&async-SE | 8.0 | 11.25 | 32.42 |
| TN30DN18 | sync-SE | 9 | 18.41 | 0 |
| | async-SE | 35.0 | 17.55 | 4.71 |
| | par&async-SE | 9.3 | 12.22 | 33.63 |
| TN30DN20 | sync-SE | 9 | 17.99 | 0 |
| | async-SE | 35.5 | 17.52 | 2.58 |
| | par&async-SE | 10.4 | 10.53 | 41.49 |

The test cases are constructed by connecting a case30 TN with several DN cases from MATPOWER 7.1 [44,45], denoted as TN30DN*i*. For instance, TN30DN*i* means a TN case30 connected with DN cases 1 to *i* in Table A1 in appendix. The impedance of transmission branches connected between TN and DNs is set as $0.002 + j0.01$ in per-unit value.

The measurement configurations are as follows. Measurements of bus power injections, voltage magnitudes, and real/reactive power flow of lines are equipped in TN since TN is usually observable. In DN, measurements of bus voltages, branch currents, powers and switch status at a few feeder locations are gathered [46]. Some pseudo measurements and virtual measurements also need to be added to improve the coverage level of measurements. Consequently, measurements of bus power injections and voltage magnitude of buses can be used. On the transmission line between TN and DN, measurements of real/reactive power of lines are equipped. The tolerance of each SE subproblem is set as $10^{-6}$ according to reference [12], and that of boundary state variables is set as $10^{-4}$. A different selection of these values of tolerance may affect the performance of algorithm, and the core principle for choosing values is to strike a balance between computational efficiency and accuracy.

According to references [47], state estimation is an optimization problem. The convergence condition of equal boundary state quantities can only make the distributed state estimation results consistent with the state estimation results of the whole network within a certain accuracy range, instead of completely consistent. However, the error of the boundary state variable is very small. It has the same order as the convergence condition of the boundary state variable 0.001, which is acceptable in engineering applications. Therefore, the value of tolerance in the numerical experiments in this paper is also set according to this

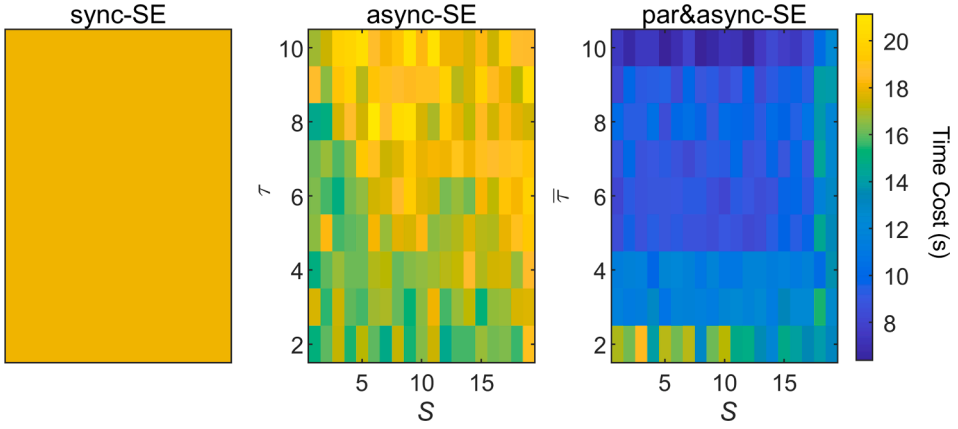


**Fig. 10.** Comparison of the time costs among the sync-SE, async-SE, and par&async-SE under situation #2 on TN30DN20.

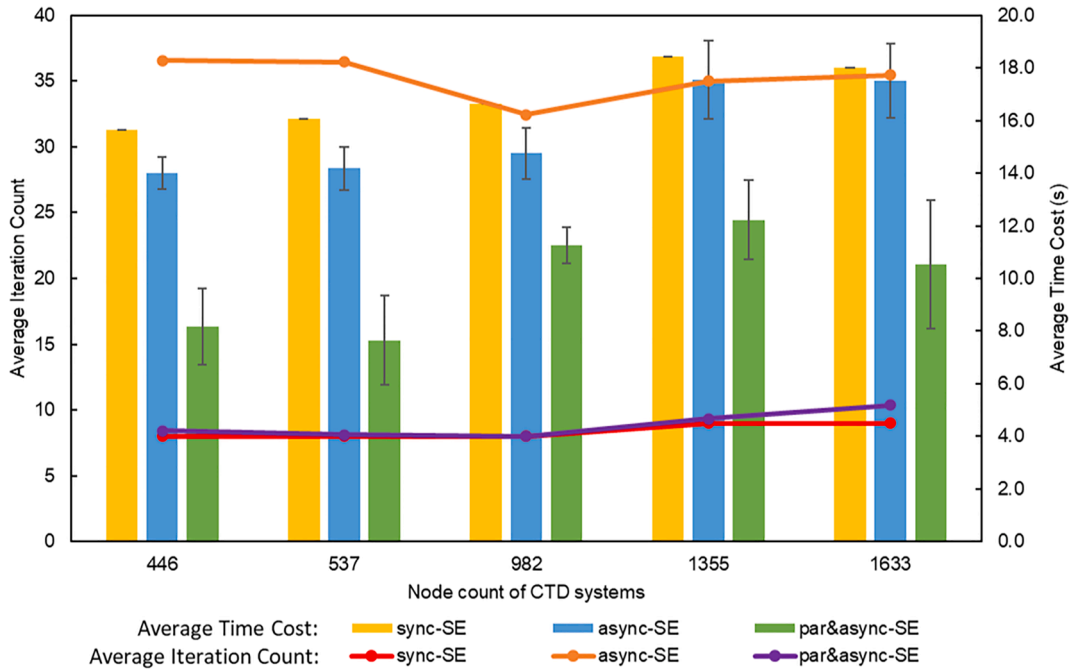


**Fig. 11.** Average iteration count and time cost among the sync-SE, async-SE, and par&async-SE under situation #2 on TN30DNis.

**Table 3**
The results of different levels of standard deviation $\sigma_1$ under situation #3 on TN30DN14.

| $\sigma_1$ | SE algorithm used in TN | SE algorithm used in DN | RMSE of state Variables ($\times 10^{-4}$) | Time cost (s) | Average iteration count |
|---|---|---|---|---|---|
| 0.010 | WLS | WLS | 4.75 | 0.979 | 5.61 |
| | WLAV | WLS | 1.09 | 2.095 | 3.29 |
| | MEAV | MEAV | 0.29 | 1.785 | 3.00 |
| 0.015 | WLS | WLS | 6.66 | 1.062 | 6.15 |
| | WLAV | WLS | 1.81 | 2.071 | 3.12 |
| | MEAV | MEAV | 0.39 | 1.854 | 3.00 |
| 0.020 | WLS | WLS | 6.75 | 1.064 | 6.12 |
| | WLAV | WLS | 2.20 | 2.100 | 3.19 |
| | MEAV | MEAV | 1.04 | 2.089 | 3.14 |
| 0.025 | WLS | WLS | 8.51 | 0.652 | 3.59 |
| | WLAV | WLS | 2.98 | 1.958 | 3.00 |
| | MEAV | MEAV | 1.55 | 2.082 | 3.11 |

precision. Compared with the convergence condition of $10^{-6}$ of each sub-problem, the convergence condition of the boundary state variable is appropriately relaxed to $10^{-4}$, which can meet the needs of engineering practice. It is reasonable to block the estimation error to the external network and not to pursue distributed state estimation that is completely consistent with the state estimation results of the entire network.

Elements of $R^{-1}$ are set as equation (33).

$$R_{ij}^{-1} = \begin{cases} \frac{1}{\sigma_{ii}^2} & i = j \\ 0 & i \neq j \end{cases} \tag{33}$$

Voltage magnitude and angle of buses are taken as state variables, whose root mean square error (RMSE) is used to evaluate the accuracy. Besides, the total time cost is used to evaluate effectiveness, which is the sum of the running time of the reception, calculation and transmission threads of TN.

In situations #1 and #2, the measured data are all valid data without gross error. Measurements are generated by adding to the power flow values a Gaussian random component with zero mean and standard deviation chosen referring to references [48,49] as, $\sigma = 0.01$ for power injection measurements, $\sigma = 0.01$ for voltage magnitude measurements, $\sigma = 0.02$ for line power flow measurements, and $\sigma = 0.005$ for measurements of boundary buses. This setting is for an ideal measurement situation and the influence of communication delays under different conditions on the algorithm is tested emphatically.

### 5.1. *Algorithm performance when considering stochastic communication delays (Situation #1)*

This subsection tests the accuracy and efficiency of different algorithms when stochastic communication delays are considered. Firstly, sensitivity analysis is performed when conducting sync-SE algorithm, and results of TN30DN14 under this situation are shown in Table A2 in the appendix. According to equation (20)(23)(24)(25), parameters $\tau$ are set as $\tau_7 = 1$, $\tau_1 = \min(\bar{\tau}, 2)$ and $\tau_6 = \min(\bar{\tau}, 3)$.

The RMSE results under different parameter settings of the sync-SE algorithm (when $S$, $\tau$ or $\bar{\tau}$ are set as 14,1, a special case of async-SE), conventional async-SE algorithm, and par&async-SE algorithm are shown in Fig. 5. The accuracies of the three algorithms are all around $10^{-3}$, with no significant difference. This meets the practical needs of the engineering and validates the accuracy of the par&async-SE algorithm proposed.

Time costs of sync-SE, async-SE, and par&async-SE of TN30DN4 and TN30DN20 with different parameter settings are shown in Table 1, Fig. 6 and Fig. 7. In Table 1, the average iteration count of par&async-SE algorithm proposed is less than that of async-SE. This result indicates that the par&async-SE algorithm has a higher convergence rate than the conventional asynchronous algorithm. It should also be noticed that the average iteration count of par&async-SE algorithm is almost the same as the conventional synchronous algorithm, which indicates that the convergence performance of par&async-SE algorithm proposed is similar to sync-SE algorithm. The average time cost, minimum time cost and standard deviation of time cost of par&async-SE algorithm proposed are all less than those of sync-SE and async-SE. This result indicates a better performance and higher calculation efficiency of the algorithm proposed, which has robustness against different parameter settings and cases. Integrated the two indices named iteration count and time cost, the par&async-SE algorithm proposed can combine the advantages of both sync-SE and async-SE algorithm and has higher convergence rate

**Table 4**
The results of conforming bad data away from boundary with $\sigma_1 = 0.01$ using different algorithms under situation #3 on TN30DN14.

| $\sigma_1$ | Measurement Position | Measurement Type | True Value (p.u.) | Bad Data (p.u.) | WLS-WLS Estimated Value (p.u.) | WLAV-WLAV Estimated Value (p.u.) | MEAV-WLS Estimated Value (p.u.) | MEAV-MEAV Estimated Value (p.u.) |
|---|---|---|---|---|---|---|---|---|
| 0.01 | TN | $P_4$ | **−0.0761** | −5.0900 | −1.1345 | −2.4604 | −0.0764 | −0.0764 |
| | | $Q_4$ | **−0.0159** | 0.0159 | −0.4738 | 0.0159 | −0.0067 | −0.0150 |
| | | $PF_{4\text{-}6}$ | **0.4416** | 9.4416 | 1.2121 | −0.9985 | 0.4415 | 0.4414 |
| | | $QF_{4\text{-}6}$ | **0.1006** | −3.1006 | −0.5588 | 0.0697 | 0.1059 | 0.1009 |
| | | $P_{10}$ | **−0.0580** | 6.3000 | 4.0378 | 2.4078 | −0.0562 | −0.0575 |
| | | $Q_{10}$ | **−0.0200** | 2.0000 | 1.5751 | 0.8972 | −0.0142 | −0.0190 |
| | | $V_{13}$ | **1.0000** | 1.5000 | 0.9674 | 0.9908 | 0.9995 | 0.9999 |
| | DN1 | $P_7$ | **−0.0032** | 5.0032 | 4.9590 | 4.0751 | 4.9616 | −0.0073 |
| | | $P_{12}$ | **−0.0013** | 0.0013 | −0.0430 | 0.0013 | −0.0394 | 0.0013 |
| | | $V_{10}$ | **0.9318** | 1.9318 | 1.0124 | 0.8853 | 0.9576 | 0.9390 |
| | DN3 | $Q_{17}$ | **−0.0012** | −2.0012 | −1.9732 | −0.7511 | −1.9567 | −0.0009 |
| | DN6 | $V_3$ | **0.9565** | 2.9617 | 0.9157 | 1.0137 | 0.9562 | 0.9564 |
| | DN9 | $Q_4$ | **−0.0022** | 3.0022 | 2.9005 | 1.2698 | 2.9000 | −0.0029 |

and lower time cost.

In Fig. 6, the time costs of conventional async-SE range from 1.6 s to 2.0 s, which varies greatly. When using conventional async-SE along with improper parameter settings, the efficiency may be worse than sync-SE. As to the sync-SE, the time cost (almost 1.8 s) is larger than those of par&async-SE. By contrast, the time costs of par&async-SE (1.2 s to 1.6 s) are less sensitive to parameter settings, displaying a more stable performance. Fig. 7 presents the results for TN30DN20, which also demonstrates this point.

The average iteration count and time cost under different parameter settings of the three algorithms are shown in Fig. 8. The iteration count of conventional async-SE is more than that of sync-SE and par&async-SE, which may cause low efficiency. Besides, the average time cost of par&async-SE is less than that of sync-SE and async-SE, indicating par&async-SE algorithm has a significant acceleration effect. The standard derivation of the par&async-SE is also less than that of async-SE. It can be concluded the par&async-SE has a more stable performance under different test cases.

### 5.2. Algorithm performance under bad communication conditions (Situation #2)

In actual condition, communication channels may have a low throughput, sometimes even overload or failure is encountered [50,51]. Thus, the communication between two entities may be very slow or even blocked. This subsection tests the accuracy and efficiency of different algorithms under these bad communication conditions. Several communication delays are set longer in Table A1 for simulation.

Time costs of sync-SE, async-SE, and par&async-SE with different parameter settings under bad communication conditions are shown in Table 2, Fig. 9 and Fig. 10. In Table 2, the average iteration count and average time cost of par&async-SE algorithm proposed are less than those of sync-SE and async-SE. The average percent of time cost reduced compared to sync-SE of the algorithm proposed is around 30–50%. This result indicates that the algorithm is effective even under bad communication conditions. In the subfigure of par&async-SE in Fig. 9, the time costs range from 6 s to 10 s under case TN30DN4. The time costs of the async-SE (12 s to 14 s) have a big difference among each other. The time cost of the sync-SE algorithm is approximately 16 s. Under different parameters, par&async-SE has a more stable performance and less time cost than sync-SE and async-SE. Fig. 10 presents the results for TN30DN20, which also demonstrates this point.

The average iteration count and time cost of sync-SE, async-SE, and par&async-SE on various TN30DN*s* under situation #2 are shown in Fig. 11. The iteration count of async-SE is more than that of sync-SE and par&async-SE, which is similar to the results in situation #1. Nonetheless, the average time cost of conventional async-SE is a little less than that of sync-SE. When it comes to the par&async-SE algorithm proposed, with subequal iteration count to sync-SE, the average time cost of it is dramatically less than that of sync-SE and async-SE. This indicates that the par&async-SE algorithm proposed has a significant acceleration effect in calculation efficiency.

The standard derivation of time cost of par&async-SE is a bit more than that of conventional async-SE in TN30DN4, TN30DN6, and TN30DN20. It can be seen from Figs. 9 and 10 that the time cost of par&async-SE under these cases is less than that of async-SE on the whole. Slow computing efficiency only appears under a few parameter settings, under which the time costs are still less than that of sync-SE.

### 5.3. Algorithm performance under gross error and bad data away from boundary (Situation #3)

In engineering practice, the presence of gross error and bad data may affect the results of state estimation. Under situation #3, measurements are generated by adding to the power flow values a Gaussian random component with zero mean and standard deviation chosen as $\sigma = \sigma_1$ for power injection measurements, voltage magnitude measurements, and line power flow measurements, and $\sigma = 0.005$ for measurements of boundary buses. This section tests the algorithm performance when different values of $\sigma_1$ are set, along with some bad data added to the measurement set. The algorithms used to solve extended-TNSE and DNSE subproblems are chosen separately, including WLS [34,35], WLAV [52,53] and maximum-exponential-absolute-value (MEAV) algorithms [54,55].

The SE results of different levels of standard deviation $\sigma_1$ under situation #3 on TN30DN14 are shown in Table 3. As $\sigma_1$ becomes larger, the algorithm proposed gets less accurate and the RMSE of state variables becomes larger, as expected, since the error in measurements disturbs the results of SE. More detailed, when using WLS algorithm without bad data detection and robustness both in TN and DN, the RMSE gets from $4.75(\sigma_1 = 0.010)$ to $8.51(\sigma_1 = 0.025)$. When WLAV algorithm is chosen for extended-TNSE subproblem, the RMSE gets from $1.09(\sigma_1 = 0.010)$ to $2.98(\sigma_1 = 0.025)$, which is less than that of WLS-WLS algorithm. Furthermore, when MEAV is chosen for both extended-TNSE and DNSE subproblems, the RMSE gets from $0.29(\sigma_1 = 0.010)$ to $1.55(\sigma_1 = 0.025)$, showing the advantage in accuracy compared with WLAV-WLS algorithm. The time cost of MEAV-MEAV algorithm is exactly the same as WLS-WLS and WLAV-WLS algorithm, ensuring the calculation efficiency of the algorithm proposed. The results also indicate that when measurements have a gross error in TN and DNs, an algorithm with robustness should be applied in TN and DNs to achieve high efficiency and accuracy.

A set of multiple conforming bad data imposed on the measurements for TN30DN14 is shown in Table 4. The performance of the different

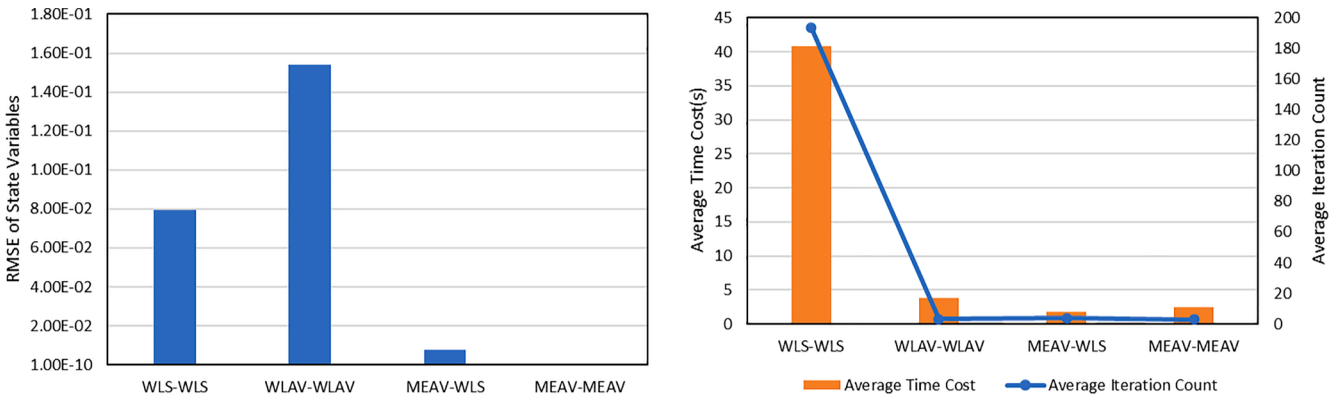


**Fig. 12.** Comparison of the RMSE, average time cost and iteration count of different algorithms with bad data under situation #3 on TN30DN14.

**Table 5**
The results of conforming bad data close to boundary using different algorithms under situation #4 on TN30DN14.

| Measurement Position | Measurement Type | True Value (p.u.) | Bad Data (p.u.) | WLS-WLS Estimated Value (p.u.) | WLAV-WLAV Estimated Value (p.u.) | MEAV-WLS Estimated Value (p.u.) | MEAV-MEAV Estimated Value (p.u.) |
|---|---|---|---|---|---|---|---|
| TN-DN1 | V | **0.9408** | 5.9594 | 1.0463 | 0.9408 | 0.9408 | 0.9408 |
| TN-DN5 | V | **0.9609** | 0.5800 | 0.9860 | 0.9608 | 0.9608 | 0.9609 |
| TN-DN8 | V | **0.9997** | 1.3052 | 1.0436 | 0.9997 | 0.9998 | 0.9997 |
| TN-DN11 | V | **0.9806** | 0.8763 | 0.9984 | 0.9806 | 0.9806 | 0.9806 |
| TN-DN3(44) | PT | **−0.0252** | −0.0548 | −0.0434 | −0.0253 | −0.0535 | −0.0253 |
| TN-DN3(44) | QT | **−0.0257** | −0.0058 | −0.0238 | −0.0256 | −0.0257 | −0.0256 |
| TN-DN5(46) | PT | **−0.0380** | −0.0877 | −0.0652 | −0.0877 | −0.0845 | −0.0381 |
| TN-DN5(46) | QT | **−0.0270** | −0.0070 | −0.0285 | −0.0268 | −0.0268 | −0.0269 |
| TN-DN8(49) | QT | **−0.0230** | −0.0131 | −0.0099 | −0.0226 | −0.0187 | −0.0225 |
| TN-DN12(53) | PT | **−0.0193** | 0.0192 | −0.0098 | −0.0193 | −0.0193 | −0.0193 |

algorithm sets is shown as below, with $\sigma_1$ set to 0.01. When bad data are imposed on the measurements in TN and DNs, WLS-WLS algorithm has been disrupted and the estimate values are far from true values. The performance of WLAV-WLAV algorithm is also not good, since the measurements named $P_4$, $Q_4$, $PF_{4\text{-}6}$, $QF_{4\text{-}6}$ are leverage measurements. The performance of the WLAV algorithm may dramatically deteriorate when there exist leverage points in measurements [53]. When MEAV robust SE algorithm is used for extended-TNSE subproblem and WLS algorithm is used for DNSE subproblem, the estimated values of TN are almost the same as true values, while in DNs, the estimate values are far from true values as well. Only when MEAV robust SE algorithm is used in TN and DNs, the estimate values are able to be perfectly consistent with the true values, illustrating the good performance and strong robustness of the MEAV-MEAV estimator.

The overall performance of different algorithms used to solve extended-TNSE and DNSE subproblems is shown in Fig. 12. When

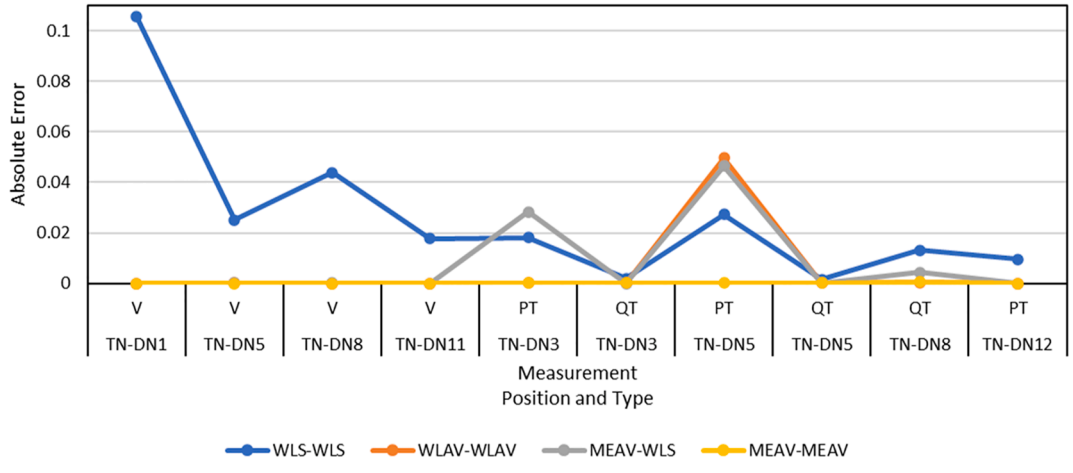


**Fig. 13.** Absolute error of estimated values using different algorithms with bad data under situation #4 on TN30DN14.

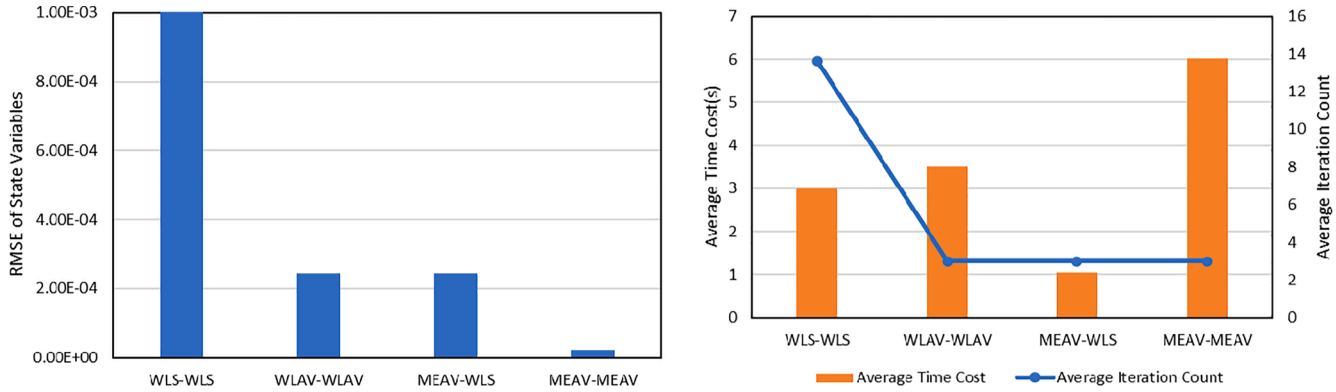


**Fig. 14.** Comparison of the RMSE, average time cost and iteration count of different algorithms with bad data under situation #4 on TN30DN14.

multiple bad data are imposed, the MEAV-MEAV algorithm performs best since it is more resistant to bad data. The combination of WLS-WLS and WLAV-WLAV perform not good and the RMSEs of state variables are large. This indicates that WLS algorithm without bad data detection is not capable for situation #3 with gross error and bad data away from boundary. The WLS algorithm also needs more back-and-forth iterations between TN and DN to converge. When it comes to WLAV algorithm, the occurrence of leverage points in measurements has an impact on the estimation results. The MEAV algorithm has good error detection and identification strategy and it can be applied to solve subproblems in the par&async-SE algorithm to achieve higher accuracy and ensure the resilience of the proposed method against the presence of gross errors.

### *5.4. Algorithm performance under gross error and bad data close to boundary (Situation #4)*

In actual conditions, the measurements at the TN and DNs coupling or close to it are usually high-precision devices, including PMUs with high accuracy. However, the occurrence of bad data of these measurements should also be considered, since it may affect the performance of the proposed method when an inappropriate algorithm is chosen to solve SE subproblems. This section tests the accuracy and efficiency of the algorithm proposed when bad data are imposed on the boundary measurements. Under situation #4, measurements are generated by adding to the power flow values a Gaussian random component with zero mean and standard deviation chosen as $\sigma = 0.01$ for power injection measurements, voltage magnitude measurements, and line power flow measurements, $\sigma = 0.02$ for power flow measurements of boundary buses, and $\sigma = 0.01$ for voltage magnitude measurements of boundary buses.

A set of multiple conforming bad data imposed on the measurements for TN30DN14 is shown in Table 5. The estimated value of different algorithm sets is also shown. A visualization form of Table 5 is shown in Fig. 13. WLS-WLS algorithm has also been disrupted and the estimate values are far from true values, leading to large absolute error. The performance of WLAV-WLAV and MEAV-WLS algorithms is also not good. WLAV-WLAV algorithm can be easily influenced by the leverage points of measurements, including voltage magnitude measurement and line power flow measurements of TN-DN5 boundary bus. When solving DNSE subproblems using WLS algorithm without robustness, only the robust algorithm used in extended-TNSE subproblems fails to identify all the bad data and leads to a large margin of error in the estimated values. When MEAV algorithm is used to solve extended-TNSE and DNSE subproblems, the estimate values are able to be perfectly consistent with the true values, resulting in low absolute error.

The overall performance of different algorithms is shown in Fig. 14. When multiple bad data are imposed on the boundary measurements, the MEAV-MEAV algorithm performs best. The result of WLS-WLS is the worse since the RMSE of state variables is the largest. This indicates that WLS algorithm without bad data detection is not capable of this situation. The WLS algorithm also needs more back-and-forth iterations between TN and DN to converge. When it comes to the time cost, the MEAV algorithm needs more time to finish the computation, which is still within tolerable limits of engineering practice.

## 6. Conclusion

This paper proposes a par&async-SE algorithm for CTD systems. Through extensive demonstration on test cases, the following observations can be obtained:

(i) The accuracies of sync-SE, conventional async-SE, and par&async-SE algorithms are similar to each other. The correctness of the par&async-SE algorithm can be verified.
(ii) The par&async-SE algorithm can deal with the circumstance when communication delays exist and achieve high efficiency under different parameter settings. With a TN broadcast transmitting strategy, a synchronous preprocessing technique, and a receive-compute-transmit parallel mechanism, the iteration count of the par&async-SE algorithm proposed has no significant difference compared to the sync-SE algorithm. Meanwhile, it can achieve a 30%~50% reduction in time cost compared to the sync-SE.
(iii) With key DN selection through sensitivity analysis method, the performance of the par&async-SE algorithm proposed is less sensitive to parameter settings than the conventional async-SE algorithm. This indicates that parameter tuning is needless for the algorithm proposed.
(iv) The par&async-SE algorithm has bad data robustness. Choosing appropriate methods to solve extended-TNSE and DNSE subproblems can eliminate the effect caused by the occurrence of bad data.

The limitation of the algorithm is that if the communication interrupts, the TN needs to deal with this situation and some fitting and fault-tolerant strategies should be put into use. As a result, further researches are needed on this issue.

*CRediT authorship contribution statement*

**Yingqi Tang:** Conceptualization, Methodology, Software, Investigation, Formal analysis, Data curation, Writing – original draft, Writing – review & editing. **Kunjie Tang:** Data curation, Writing – original draft. **Shufeng Dong:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China, under Grant 52077193.

## Appendix A. Information about DN cases

The test cases used in this paper are shown in Table A1. Here, all cases are from MATPOWER 7.1 [44,45] and detailed information of the cases can be found in the source code of MATPOWER [45]. The values of column named "Base DN Case from MATPOWER" are filenames of the cases in the source code. The values of column named "# of Bus Connected to TN" are the ids of buses in TN which the specific DN connects to. Communication delays obey normal distribution, in which μ for mean values and σ for standard deviation values of delays.

## Appendix B. Information about sensitivity analysis result

See Table A2.

**Table A1**
Information about DN Cases.

| # of DN | Base DN Case from MATPOWER | # of Bus Connected to TN | Communication Delays Distribution $N(\mu, \sigma^2)$ | | | |
|---|---|---|---|---|---|---|
| | | | Situation #1 | | Situation #2 | |
| | | | μ/ms | σ/ms | μ/ms | σ/ms |
| 1 | case136ma | 11 | 100 | 20 | 1000 | 200 |
| 2 | case118zh | 12 | 75 | 15 | 75 | 15 |
| 3 | case85 | 13 | 50 | 10 | 50 | 10 |
| 4 | case74ds | 14 | 25 | 5 | 25 | 5 |
| 5 | case69 | 15 | 100 | 20 | 100 | 20 |
| 6 | case22 | 20 | 75 | 15 | 75 | 15 |
| 7 | case28da | 21 | 50 | 10 | 500 | 100 |
| 8 | case33mg | 22 | 25 | 5 | 25 | 5 |
| 9 | case34sa | 23 | 100 | 20 | 100 | 20 |
| 10 | case38si | 24 | 75 | 15 | 75 | 15 |
| 11 | case51ga | 25 | 50 | 10 | 50 | 10 |
| 12 | case51he | 26 | 25 | 5 | 25 | 5 |
| 13 | case69 | 27 | 100 | 20 | 100 | 20 |
| 14 | case141 | 28 | 75 | 15 | 75 | 15 |
| 15 | case74ds | 29 | 50 | 10 | 500 | 100 |
| 16 | case85 | 30 | 25 | 5 | 25 | 5 |
| 17 | case94pi | 19 | 100 | 20 | 100 | 20 |
| 18 | case118zh | 18 | 75 | 15 | 75 | 15 |
| 19 | case136ma | 17 | 50 | 10 | 50 | 10 |
| 20 | case141 | 16 | 25 | 5 | 25 | 5 |

**Table A2**
The results of self and mutual-sensitivity analysis under situation #1 on TN30DN14.

| $i$ of $\left\|\frac{\Delta S_{di}^{eq}}{\Delta V_{ri}}\right\|$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Value | 4.08 | 16.58 | 15.15 | 6.15 | 14.30 | 10.05 | **55.92** | 21.04 | 10.40 | 6.46 | 13.78 | 7.15 | 30.90 | 36.85 |
| Value | | $j$ of $\left\|\frac{\Delta S_{di}^{eq}}{\Delta S_{dj}^{eq}}\right\|$ | | | | | | | | | | | | |
| $i$ of $\left\|\frac{\Delta S_{di}^{eq}}{\Delta S_{dj}^{eq}}\right\|$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| | 1 | 1.00 | 2.04 | 7.69 | 1.30 | 17.31 | **139.07** | 3.34 | 14.40 | 0.13 | 0.14 | 5.81 | 1.84 | 4.74 | 2.91 |
| | 2 | 0.57 | 1.00 | 4.37 | 1.06 | 9.84 | 79.03 | 2.01 | 8.18 | 0.76 | 0.96 | 3.30 | 1.05 | 3.79 | 1.53 |
| | 3 | 6.78 | 0.35 | 1.00 | 0.43 | 2.25 | 18.08 | 0.57 | 4.11 | 0.91 | 0.98 | 2.75 | 1.53 | 1.16 | 0.50 |
| | 4 | 1.31 | 1.57 | 6.53 | 1.00 | 14.70 | 118.09 | 2.57 | 12.23 | 0.00 | 0.00 | 4.93 | 1.57 | 3.71 | 2.23 |
| | 5 | 7.01 | 0.21 | 2.53 | 1.08 | 1.00 | 8.03 | 0.34 | 4.25 | 0.94 | 1.01 | 2.84 | 1.58 | 2.93 | 0.29 |
| | 6 | 2.65 | 0.17 | 0.39 | 0.04 | 0.38 | 1.00 | 0.34 | 1.61 | 0.35 | 0.38 | 1.07 | 0.60 | 0.14 | 0.26 |
| | 7 | 0.35 | 0.70 | 2.73 | 0.75 | 6.13 | 49.27 | 1.00 | 5.10 | 0.38 | 0.48 | 2.06 | 0.65 | 2.67 | 0.87 |
| | 8 | 1.65 | 0.16 | 0.53 | 0.10 | 1.20 | 9.66 | 0.30 | 1.00 | 0.22 | 0.24 | 0.67 | 0.37 | 0.37 | 0.23 |
| | 9 | 7.48 | 1.31 | 1.10 | 0.00 | 1.07 | 7.69 | 2.63 | 8.75 | 1.00 | 1.26 | 3.03 | 1.69 | 0.00 | 2.00 |
| | 10 | 6.94 | 1.04 | 1.02 | 0.00 | 0.99 | 6.09 | 2.09 | 6.94 | 1.23 | 1.00 | 2.81 | 1.57 | 0.00 | 1.58 |
| | 11 | 2.47 | 0.48 | 1.32 | 0.24 | 2.98 | 23.95 | 0.97 | 3.22 | 0.37 | 0.46 | 1.00 | 0.56 | 0.88 | 0.73 |
| | 12 | 4.43 | 1.93 | 4.17 | 0.73 | 9.39 | 75.45 | 3.89 | 12.92 | 1.48 | 1.86 | 4.02 | 1.00 | 2.67 | 2.95 |
| | 13 | 0.48 | 0.43 | 1.76 | 0.37 | 3.96 | 31.81 | 0.71 | 3.29 | 0.00 | 0.00 | 1.33 | 0.42 | 1.00 | 0.61 |
| | 14 | 0.43 | 1.20 | 3.31 | 1.28 | 7.45 | 59.81 | 1.71 | 6.19 | 0.50 | 0.63 | 2.50 | 0.79 | 4.57 | 1.00 |

## References


[1] Dy-Liacco TE. Modern control centers and computer networking. IEEE Comput Appl Power 1994;7(4):17–22.
[2] Cassel WR. Distribution management systems: functions and payback. IEEE Trans Power Syst 1993;8(3):796–801.
[3] Lin C, Wu W, Zhang B, Wang B, Zheng W, Li Z. Decentralized reactive power optimization method for transmission and distribution networks accommodating large-scale DG integration. IEEE Trans Sustain Energy 2017;8(1):363–73.
[4] Tang K, Dong S, Song Y. Successive-intersection-approximation-based power flow method for integrated transmission and distribution networks. IEEE Trans Power Syst 2020;35(6):4836–46.
[5] Tang K, Dong S, Liu Y, Wang L, Song Y. Asynchronous distributed global power flow method for transmission–distribution coordinated analysis considering communication conditions. Electr Power Syst Res 2020;182:106256.
[6] Sun H, Guo Q, Zhang B, Guo Y, Li Z, Wang J. Master–slave-splitting based distributed global power flow method for integrated transmission and distribution analysis. IEEE Trans Smart Grid 2015;6(3):1484–92.
[7] Gupta M, Abhyankar AR., Coordinated load flow solution for coupled transmission-distribution system incorporating load modelling. In: 2018 20th National Power Systems Conference (NPSC), December 2018, pp. 1–6. doi: 10/ghkvvm.
[8] Lin C, Wu W, Shahidehpour M. Decentralized AC optimal power flow for integrated transmission and distribution grids. IEEE Trans Smart Grid 2020;11(3):2531–40.
[9] Tang K, Dong S, Ma X, Fei Y, Song Y. Heterogeneous-decomposition-based coordinated optimisation for integrated transmission and distribution networks considering communication conditions. Transmiss Distribut IET Generat 2020;14 (13):2558–65.
[10] Li Z, Guo Q, Sun H, Wang J. Coordinated transmission and distribution AC optimal power flow. IEEE Trans Smart Grid 2018;9(2):1228–40.
[11] Tang K, Dong S, Ma X, Lv L, Song Y. Chance-constrained optimal power flow of integrated transmission and distribution networks with limited information interaction. IEEE Trans Smart Grid 2021;12(1):821–33.
[12] Sun HB, Zhang BM. Global state estimation for whole transmission and distribution networks. Electr Power Syst Res 2005;74(2):187–95.
[13] Nogueira EM, Portelinha RK, Lourenço EM, Tortelli OL, Pal BC. Novel approach to power system state estimation for transmission and distribution systems. Transm Distribut IET Generat 2019;13(10):1970–8.
[14] Abbasi AR, Seifi AR. A new coordinated approach to state estimation in integrated power systems. Int J Electr Power Energy Syst 2013;45(1):152–8.
[15] Yuan Z, Hesamzadeh MR. Hierarchical coordination of TSO-DSO economic dispatch considering large-scale integration of distributed energy resources. Appl Energy 2017;195:600–15. https://doi.org/10.1016/j.apenergy.2017.03.042.
[16] Li Z, Sun H, Guo Q, Wang J, Liu G. Generalized master–slave-splitting method and application to transmission-distribution coordinated energy management. IEEE Trans Power Syst 2019;34(6):5169–83.
[17] Wang B, He G, Dong S. An improved algorithm for state estimator based on maximum normal measurement rate. IEEE Trans Power Syst 2013;28(4):4879–80.
[18] He G, Dong S, Qi J, Wang Y. Robust state estimator based on maximum normal measurement rate. IEEE Trans Power Syst 2011;26(4):2058–65.
[19] Zhao H. A new state estimation model of utilizing PMU measurements; 2006, pp. 1–5. doi: 10.1109/ICPST.2006.321443.
[20] Gu Y, Liu T, Wang D, Guan X, Xu Z. Bad data detection method for smart grids based on distributed state estimation. In: 2013 IEEE International Conference on Communications (ICC); 2013. p. 4483–4487. doi: 10/ghk76w.
[21] Ângelos EWS, Asada EN. Improving state estimation with real-time external equivalents. IEEE Trans Power Syst 2016;31(2):1289–96.
[22] Horisberger HP, Richard JC, Rossier C. A fast decoupled static state-estimator for electric power systems. IEEE Trans Power Apparat Syst 1976;95(1):208–15.
[23] Guo J, Hug G, Tonguz OK. On the role of communications plane in distributed optimization of power systems. IEEE Trans Ind Inf 2018;14(7):2903–13.
[24] Xu J, Sun H, Dent CJ. ADMM-based distributed OPF problem meets stochastic communication delay. IEEE Trans Smart Grid 2019;10(5):5046–56.
[25] Kekatos V, Giannakis GB. Distributed robust power system state estimation. IEEE Trans Power Syst 2013;28(2):1617–26.
[26] Wang J, Li T. Distributed multi-area state estimation for power systems with switching communication graphs. IEEE Trans Smart Grid 2021;12(1):787–97.
[27] Anandini GN, Gupta I. Estimating the lost real-time measurements under communication failure for distribution system state estimation. In: 2016 National Power Systems Conference (NPSC), December 2016, pp. 1–5. doi: 10/ghkxsj.
[28] Lakshminarasimhan S, Girgis AA. Hierarchical state estimation applied to wide-area power systems. In: 2007 IEEE Power Engineering Society General Meeting; 2007. p. 1–6. doi: 10/d2mzhb.
[29] Rana M, Li L, Su SW. Distributed state estimation over unreliable communication networks with an application to smart grids. IEEE Trans Green Commun Network 2017;1(1):89–96.
[30] Cavraro G, Dall'Anese E, Bernstein A. Dynamic power network state estimation with asynchronous measurements. In: 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Nov. 2019, pp. 1–5. doi: 10/ghk77p.
[31] Xie L, Choi D, Kar S. Cooperative distributed state estimation: Local observability relaxed. In: 2011 IEEE Power and Energy Society General Meeting, Jul. 2011, pp. 1–11. doi: 10/ctqjj8.
[32] Ren Z, Chen Y, Huang S, Heleno M, Xia Y. A fully distributed coordination method for fast decoupled multi-region state estimation. IEEE Access 2019;7:132859–70.
[33] Korres GN. A distributed multiarea state estimation. IEEE Trans Power Syst 2011; 26(1):73–84.
[34] Abur A, Exposito AG. Power system state estimation: theory and implementation. CRC Press; 2004.
[35] Wood AJ, Wollenberg BF, Sheblé GB. Power generation, operation, and control. John Wiley & Sons; 2013.
[36] Baudet GM. Asynchronous iterative methods for multiprocessors. Journal of the ACM (JACM) 1978;25(2):226–44.
[37] J. Guo, G. Hug, O. Tonguz, Impact of communication delay on asynchronous distributed optimal power flow using ADMM, in: 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm), Oct. 2017, pp. 177–182. doi: 10/ghkxzw.
[38] Schmidt HP, Guaraldo JC, Jardini JA, Da Mota Lopes M. Interchangeable balanced and unbalanced network models for integrated analysis of transmission and distribution systems. IEEE Trans Power Syst 2015;30(5):2747–54.
[39] Tang K, Dong S, Zhu C, Song Y. Affine arithmetic-based coordinated interval power flow of integrated transmission and distribution networks. IEEE Trans Smart Grid 2020;11(5):4116–32.
[40] Monticelli A. State estimation in electric power systems: a generalized approach. Springer Science & Business Media; 2012.
[41] Mestav KR, Luengo-Rozas J, Tong L. Bayesian State estimation for unobservable distribution systems via deep learning. IEEE Trans Power Syst 2019;34(6): 4910–20.
[42] Manitsas E, Singh R, Pal BC, Strbac G. Distribution system state estimation using an artificial neural network approach for pseudo measurement modeling. IEEE Trans Power Syst 2012;27(4):1888–96.
[43] Cao Z, Wang Y, Chu C-C, Gadh R. Robust pseudo-measurement modeling for three-phase distribution systems state estimation. Electr Power Syst Res 2020;180: 106138.
[44] Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. MATPOWER: steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans Power Syst 2011;26(1):12–9.
[45] R.D. Zimmerman, C.E. Murillo-Sánchez, MATPOWER. Zenodo, 2020. doi: 10.5281/zenodo.4074135.
[46] Primadianto A, Lu C-N. A review on distribution system state estimation. IEEE Trans Power Syst 2017;32(5):3875–83.
[47] Zhang H, Yi W. Distributed state estimation method for power systems based on asynchronous iteration mode. Automat. Electric Power Syst. 2014;38(09):125–31.
[48] Tang Y, Dong S, Tang K, Zhu Y, Ma X. Transmission network broadcasting mechanism-based improved asynchronous state estimation method for transmission and distribution interconnected power grid. High Voltage Eng. 2021; 47(07):2377–86.
[49] Wang S, Liu G, Huang R, Qin S. State estimation method for active distribution networks under environment of hybrid measurements with multiple sampling periods. Automat. Electric Power Syst. 2016;40(19):30–6.
[50] Cho J, Ramgolam DI, Schaefer KM, Sandlin AN. The Rate and delay in overload: an investigation of communication overload and channel synchronicity on identification and job satisfaction. J. Appl. Commun. Res. 2011;39(1):38–54.
[51] Stephens KK, Mandhana DM, Kim JJ, Li X, Glowacki EM, Cruz I. Reconceptualizing communication overload and building a theoretical foundation. Commun. Theory 2017;27(3):269–89.
[52] Singh H, Alvarado FL. Weighted least absolute value state estimation using interior point methods. IEEE Trans Power Syst 1994;9(3):1478–84.
[53] Chen Y, Liu F, Mei S, Ma J. A robust WLAV state estimation using optimal transformations. IEEE Trans Power Syst 2015;30(4):2190–1.
[54] Chen Y, Liu F, He G, Mei S, Yanlan Fu. Maximum exponential absolute value approach for robust state estimation. In: 2012 IEEE International Conference on Power System Technology (POWERCON); 2012. p. 1–6.
[55] Chen Y, Ma J, Zhang P, Liu F, Mei S. Robust state estimator based on maximum exponential absolute value. IEEE Trans Smart Grid 2017;8(4):1537–44.