

Spatial Matrix Based Clustering of Sparse Electric Power Networks

Md. Ashfaque Rahman, *Member, IEEE* and Ganesh Kumar Venayagamoorthy, *Senior Member, IEEE*

Abstract—Distributed computation is an effective policy to increase the speed of the sparse networked systems. In a sparse network, clustering methods like k -means does not work directly as it cannot explore the connectivity of the system. To solve the problem, two modification methods are proposed in the existing graph and a new graph named *Spatial Matrix* is introduced in this paper. The proposed modification is a fast process and the computation time can be considered negligible compared to the rest of the process. Thus it preserves the ultimate objective of the distribution. It works as a pre-conditioning that can be used with a wide range of clustering and mathematical tools. With distributed state estimation of IEEE 14, 68, and 118-bus systems with automatic clustering, the effectiveness of the spatial matrix is demonstrated.

Index Terms—Distributed computation, network clustering, sparse network, spatial matrix, WLS estimator.

I. INTRODUCTION

One of the major classes of practical systems is the sparse network [1]. It appears in a large scale in social, computer, biological, transportation, citation, and power systems networks [2]. In a sparse network, each node is connected with a few other nodes, and none of them is connected with a large number of nodes. However, the networks are usually vast in nature, and most of their computations are time-consuming [3].

A major way to make the process fast is to distribute the whole system in clusters. Each cluster executes a local process with its own nodes ignoring the effects of the others. Once the processes are done, they exchange and update the results. Based on the nature of the process, they may repeat the local processes and exchange the results. Repeating these steps, they try to converge to the optimal result provided by the non-distributed/centralized method.

However, it is not an easy task to build the clusters that keeps the connectivity of the network. Moreover, the accuracy of the distributed method needs to meet a minimum threshold. Though there are a good number of works on building the clusters in different fields, most of them are time-consuming that hampers the ultimate objective of clustering [4]. The related works on graph clustering can be classified in two

The authors are with Real-Time Power and Intelligent Systems Laboratory, Holcombe Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, United States. Dr. Rahman is also with the Department of Electrical Engineering and Computer Science, Texas A&M University - Kingsville, Kingsville, TX (email: mdashfaque.rahman@tamuk.edu). Dr. Venayagamoorthy is also with the School of Engineering, University of Kwazulu-Natal, Durban 4041, South Africa (email: gkumar@ieee.org)

This work was supported in part by US National Science Foundation (NSF) under grant #1312260, and #1408141, and the Duke Energy Distinguished Professor Endowment Fund. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the financial supporters.

major groups, general applications, and specific applications. A few of them are mentioned below.

a) General Applications: A complete survey on graph clustering can be found in [5]. A general method is proposed in [6] based on the *minimum cut trees* on the graph. A discrete uncoupling process called Markov cluster process is proposed for finite spaces in [7]. Two algorithms named k -medoids and the Girvan-Newman method are compared in [8]. To extract the common information shared in a cluster, a method named weighted minimum common supergraph is proposed in [9] that can separate the information from noise and distortions. The idea of *cliques* is generalized in [10] as *distance- k cliques*. The weighted kernel k -means and several graph clustering objectives are combined to perform a semi-supervised clustering of data in [11].

b) Specific Applications: An algorithm is proposed in [12] to identify the dense clusters of co-cited references. Another method is developed in [13] to determine the class of the software systems. A method called Spatial Partition Clustering (SPC) is compared with three other methods for path query processing in [14]. In [15], the authors present intelligent clustering techniques for efficient prediction of sugar production. A software named IND based clustering is used for dense imaging network in [16]. Quantum clustering based multi-valued quantum fuzzification decision tree is used to detect the fault in analog circuit in [17]. The feature selection is considered as a clustering process with data decomposition technique in [18] and a novel feature selection method is proposed based on the non-negation matrix factorization.

In this paper, a fast and simple modification is proposed to effectively cluster the nodes based on their physical positions. With the aid of k -means clustering, the proposed method is demonstrated to perform faster for power system state estimation in Section 4. k -means cannot be applied directly on a network as it cannot incorporate the effects of the edges/connectivity [19]. The proposed method is applicable for both weighted and non-weighted graph. The effectiveness of the proposed method is tested for the state estimation problem of IEEE 14, 68 and 118-bus test system.

State estimation of power systems is a centralized process based on Weighted Least Squares (WLS) method [20], [21]. WLS is the most efficient estimator for any process with Gaussian noise. It includes matrix inversion or Cholesky decomposition in each iteration that makes it time-consuming. The computational complexity of matrix inversion is in the order of n^3 [22]. So, the size of the system plays an important role in WLS estimation.

To reduce the size of the problem, distributed estimation is being analyzed for a long time. Different works focus on different aspects of the distribution. Some focus on the speed

[23], [24], some on the local observability [25], some on islanding [26], some on inter-area oscillation [27], some on voltage stability monitoring [28] and some on robustness [29]. However, none of them have focused on automatic clustering of the buses where the above mentioned constraints are easily met. An operator may deal with thousands of buses and he may need to distribute those regardless of the observability and robustness.

The main contributions of the paper can be summarized as follows,

- Two different methods are proposed to form a modified graph named spatial matrix based on the physical locations of the nodes and the connection matrix. With the help of the k -means, a complete clustering algorithm is formed. Both methods convert the connectivities to a pair of new nodes. Each new node holds the attribute of the two nodes representing the connection.
 - Method 1: It introduces the new nodes on the vertical and horizontal locations of the existing nodes. It creates some diversities in the placements of the nodes corresponding to the connectivity.
 - Method 2: It places the new nodes on the connecting edge of two existing nodes. The new nodes are placed close to the existing nodes. This is helpful for preserving the observability of the clusters.
- The effects of the clustering algorithm are demonstrated on two test power systems under different conditions. It exhibits an effective way to parallelize the state estimation process.

It is not mandatory to use physical locations of the nodes. In fact, it is recommended that the relevant distances are used for each application domain. For example, for electric power grids, the electrical distances can also be considered. However, except the power grid, the analysis for other application domains is beyond the scope of this paper.

The rest of the paper is organized as follows. Section II introduces the system model for centralized and distributed WLS estimation. Two different methods are proposed to form the spatial matrix and their working principles are analyzed in Section III. The methods are applied on IEEE 14, 68, and 118-bus systems and the formed clusters are studied in Section IV. The simulation results are discussed in Section V. The paper is concluded with future plans in Section VI.

II. SYSTEM MODEL

To monitor the current status of the power system, measurements are collected from different parts of the system. They are taken in the forms of power flows, power injections, voltage magnitudes, and current magnitudes etc. which contain errors of different levels. Let, \mathbf{z} denotes an $m \times 1$ measurement vector including error. The relation between \mathbf{z} , the state vector \mathbf{x} , and the measurement error \mathbf{e} can be written as,

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{e} \quad (1)$$

Where, $h(\cdot)$ denotes the nonlinear function of the measurements.

In state estimation, voltage magnitudes and angles are

considered as the state variables. They form the set with minimum cardinality which can describe the whole system [30]. The angle of the reference bus is considered as the reference angle and all other angles are calculated with respect to that. If there are N buses, the state vector \mathbf{x} can be represented as,

$$\mathbf{x} = [\theta_2 \ \theta_3 \dots \theta_N \ V_1 \ V_2 \dots V_N]^T \quad (2)$$

Here, θ and V , with proper subscripts, represent voltage angles and magnitudes respectively. If the number of buses in the system is N , there will be $2N - 1$ state variables. In the process of estimation, the number of measurements exceeds the number of states to form an overdetermined system.

The accuracy is measured by the L_2 -norm of the residues which are calculated as the difference between the original measurements and the estimated measurements. Minimizing the norm (also referred to as the fitness) is the objective of the optimization problem,

$$\min_{\hat{\mathbf{x}}} \|\mathbf{z} - h(\hat{\mathbf{x}})\| \quad (3)$$

where, $\hat{\mathbf{x}}$ is the estimated state vector.

A. Weighted Least Squares Estimation

Like other nonlinear problems, WLS estimator linearizes the system over a small range. Then it applies linear operations to get an updated value. The system is linearized again based on this updated value and uses the linear estimation. This process is repeated unless the estimated value converges. In these methods, \mathbf{x} is started with a close value to the solution. In the beginning, when there is no previous value, all voltage magnitudes start as 1 and all voltage angles as 0 which is known as flat start [31],

$$\mathbf{x} = [0 \ 0 \dots 0 \ 1 \ 1 \dots 1]^T$$

After collecting m measurements and constructing the Jacobian matrix $\mathbf{H}(\mathbf{x})$ at flat start, in WLS estimation, the following steps are repeated until the state vector converges to a solution,

- step 1: $\Delta \mathbf{x} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} (\mathbf{z} - h(\mathbf{x}))$
- step 2: $\mathbf{x}_{n+1} = \mathbf{x}_n + \Delta \mathbf{x}$
- step 3: update $h(\mathbf{x})$ with $\mathbf{x} = \mathbf{x}_{n+1}$
- step 4: update $\mathbf{H}(\mathbf{x})$ with $\mathbf{x} = \mathbf{x}_{n+1}$

Here, the matrix, \mathbf{W} denotes the relative weights of the measurements which are usually taken as the inverse of the corresponding error variances.

B. Distributed WLS Estimation

In case of distributed estimation, the system is divided in different mutually exclusive or non-exclusive clusters. Each cluster executes their local estimation. After completing estimation, they exchange and update their results.

The local estimation is similar to the centralized estimation. The only difference is that it includes only one cluster instead of the whole system. For each cluster, a bus is taken as

a reference and all measurements are normalized based on that. As a result, the estimation results are also based on those cluster references. After exchanging the results among each other, the results of the intersections of the clusters are updated. To do so, all of them are converted back to their corresponding references. As the buses on the intersections have different values for different clusters, their means can be taken for further processing.

However, it is not mandatory that the local processes have to run till complete convergence. The clusters may exchange and update their results after completing a fixed number of iterations. After updating, they can run the local processes again and exchange and update accordingly.

III. SPATIAL MATRIX BASED CLUSTERING

The spatial matrix is a 2-D graph with the base nodes added with some virtual nodes. The base nodes are the given nodes with known physical locations. The virtual nodes are created based on the base nodes to adapt the connectivity for the clustering. The complete process is summarized in Fig. 1.

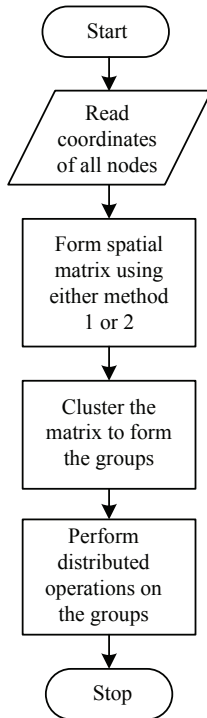


Fig. 1: Flowchart for forming spatial matrix and executing distributed operations.

A. Formation of Spatial Matrix

Spatial matrix is based on the spatial locations of the nodes and the connectivity of the nodes. The connectivity matrix plays an essential role in the formation of spatial matrix. If a system has N_n number of nodes, then the basic or first order connectivity matrix, B_1 is defined as an $N_n \times N_n$ matrix where,

$$B_1(i, j) = \begin{cases} 1 & \text{if node } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise} \end{cases}$$

B_1 represents the direct connections/edges of the network. B_1 can be used to find the second, third and higher order connectivity matrices. It is done in the following way by taking the power over B_1 , and converting the non-zero elements to 1,

$$B_2 = B_1 \times B_1 \quad (4)$$

$$B_3 = B_2 \times B_1 \quad (5)$$

$$B_4 = B_3 \times B_1 \quad (6)$$

$$B_n(i, j) = \begin{cases} 1 & \text{if } B_n(i, j) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

It can be noted that, all of B_1, B_2, B_3, B_4 are symmetric matrices. Only one of them can be used in forming the spatial matrix. Once the connectivity matrix is defined, the spatial matrix is started with a 2-D graph. The base nodes are placed on the graph according to their physical location. Then for each edge in B_1 (or higher orders, whatever is chosen) that is connected with two nodes located at (x_1, y_1) , and (x_2, y_2) , two virtual nodes are placed in one of the two methods,

- Method 1: at (x_1, y_2) , and at (x_2, y_1)
- Method 2: at $(x_1 + \alpha(x_2 - x_1), y_1 + \alpha(y_2 - y_1))$, and at $(x_2 + \alpha(x_1 - x_2), y_2 + \alpha(y_1 - y_2))$

Though α can take any value in between 0 and 1, the preferred choice is $0.05 \leq \alpha \leq 0.2$. Both methods take the neighbors close to each other. The details of the methods are discussed in Section III-C and Fig. 2.

B. Clustering of Spatial Matrix

Once the matrix is formed, it can be clustered with any standard clustering algorithm like the k -means clustering or hierarchical clustering. In this paper, k -means is used to make the clusters. With this method, the formed clusters contain base as well as virtual nodes. After clustering, all virtual nodes are converted back to their corresponding base nodes. As each virtual node is connected with two base nodes, each of them creates two nodes for the clusters. Then the clusters contain multiple placements of the base nodes in each cluster. At the end, the clusters are made precise by removing the redundant elements.

The complete process of spatial matrix based clustering using Method 1 is shown in Algorithm 1 where x_{base} is an $N_n \times 2$ matrix with the x , and y coordinates of the nodes, and N_g is the number of clusters.

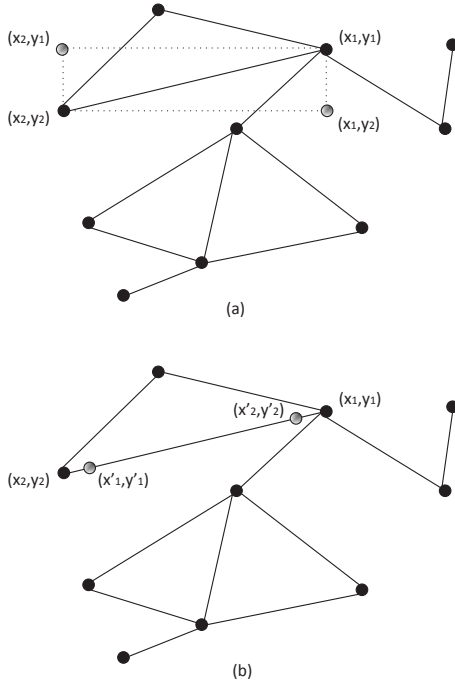


Fig. 2: Two methods to form the spatial matrix, (a) Method 1, (b) Method 2.

Algorithm 1 Pseudo-code for clustering with spatial matrix

```

1: Input  $x_{base}, N_g$ 
2: Define  $x_{vir} = []$ , and  $\mathcal{S} = [1 : N_n; 1 : N_n]^T$ 
3: for  $i = 1 : N_n$  do
4:   for  $j = 1 : N_n$  do
5:     if  $B(i, j) = 1$  and  $i \neq j$  then
6:        $x_{vir} = [x_{vir}; [x_{base}(i, 1) \ x_{base}(j, 2)]]$  // add
        locations of the new nodes at the end of  $x_{vir}$ 
7:        $\mathcal{S} = [\mathcal{S}; i \ j]$  // add the corresponding node
        numbers
8:     end if
9:   end for
10: end for
11:  $y = kmeans([x_{base}; x_{vir}], N_g)$  // create  $N_g$  clusters
12:  $groups = cell(1, N_g)$  // create empty cells
13: for  $i = 1 : N_g$  do
14:    $groups\{y(i)\} = [groups\{y(i)\} \ \mathcal{S}(i, :)]$ 
15: end for
16: for  $i = 1 : N_g$  do
17:    $groups\{i\} = unique(groups\{i\})$  // remove redundant
        elements
18: end for
19: Return  $groups$ 

```

For Method 2, Line 6 is replaced with,

$$x_{vir} = [x_{vir}; [x_{base}(i, 1) + \alpha * (x_{base}(j, 1) - x_{base}(i, 1)) \ x_{base}(i, 2) + \alpha * (x_{base}(j, 2) - x_{base}(i, 2))]]],$$

where, $0 \leq \alpha \leq 1$

C. Analysis of Spatial Matrix

The advantage of spatial matrix over the simple location based clustering is comprehensible from the formation of the matrix. For example, let us consider a part of a network shown in Fig. 2. The solid circles represent the base nodes, and the non-solid ones show the virtual nodes. The solid lines show the direct connection between the base nodes. Part (a) shows the placement of the virtual nodes using method 1, and part (b) shows that for method 2.

From the figure, it can be seen that the nodes at (x_1, y_1) , and (x_2, y_2) are physically far from each other though they are first order neighbors. In case of direct application of k -means clustering on the graph, it is very less probable that they will be clustered together. Upon creation of the virtual nodes at (x_1, y_2) , and at (x_2, y_1) according to method 1, the neighbors effectively come closer and the probability of having them together increases significantly.

The virtual nodes can be taken even closer using method 2. It creates the virtual nodes on the connecting lines of the base nodes. The value of α can control the distance between the base and virtual nodes. $\alpha = 0$ or $\alpha = 1.0$ means no distance, while $\alpha = 0.5$ means the maximum distance. At the maximum distance, the virtual nodes stay in the mid of the base nodes.

IV. SIMULATION RESULTS

To test the effectiveness of the clustering by spatial matrix, IEEE 14, 68, and 118-bus test systems are used. The 14-bus system has 20 unique transmission lines with five generators. For 68-bus system, the number of unique connections is 83, and it has 16 generators as shown in Fig. 3. 118-bus system has a total of 186 lines. The details of the systems can be found in [32] (14-bus), [33] (68-bus) and in [32] (118-bus). The physical distances for both systems are calculated based on the impedances of the network. The measurements are collected at the rate of 30 samples/second that is the typical rate of the phasor measurement units. Then an artificial Gaussian noise of around 0.25-5% of the original measurements is added with them.

A. Clusters Formed by Spatial Matrix

As mentioned earlier, the connectivity matrix has significant impact on the formation of the clusters. The spatial matrices formed by method 1 for different orders of connectivity for IEEE 14-bus system are shown in Fig. 4. The blue stars show the base nodes, and the red ones show the virtual nodes. It can be seen that with the increase in order, the number of virtual nodes increases. As a result, the connectivity gets stronger, and the sparsity gets reduced.

Based on the order of connectivity, the clusters formed by the k -means clustering are shown in Table I. The number of clusters is set to five. For method 2, α is set to 0.1. The table shows that with the increase in the order, the number of nodes in a cluster increases. There will be an order at which, the whole system will be included in one cluster. As the execution time directly depends on the size of the cluster with maximum nodes, a lower order of B is a preferred choice.

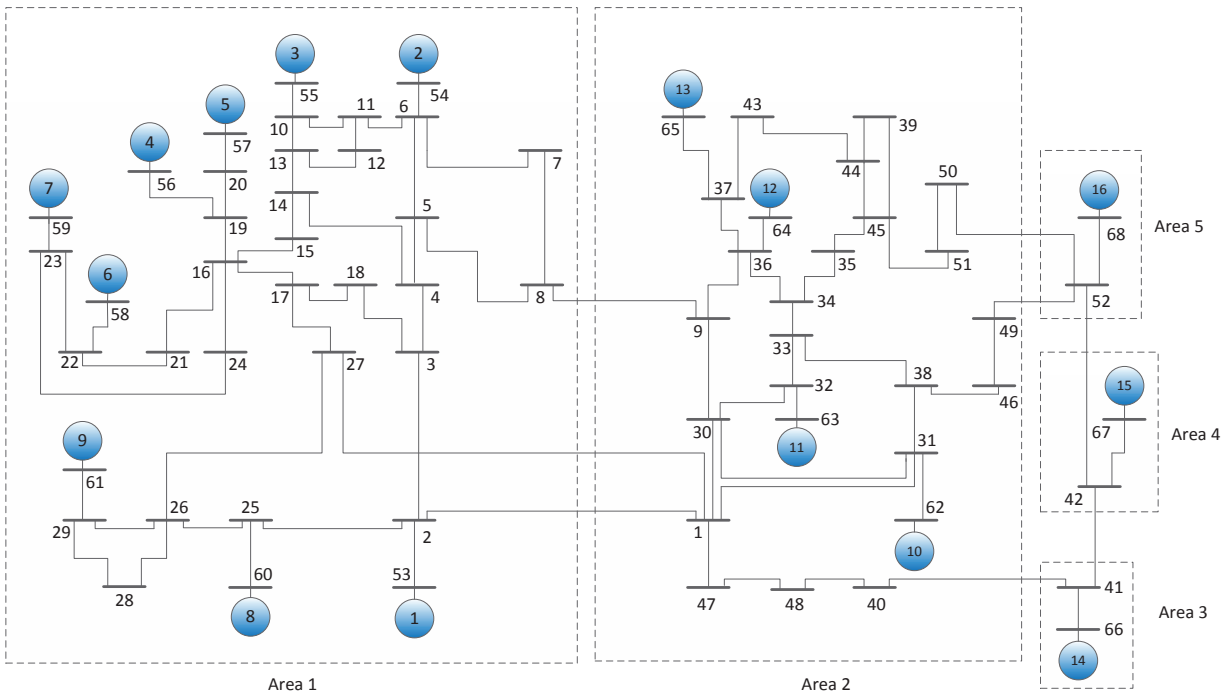


Fig. 3: IEEE 16-machine 68-bus New-York-New England test system.

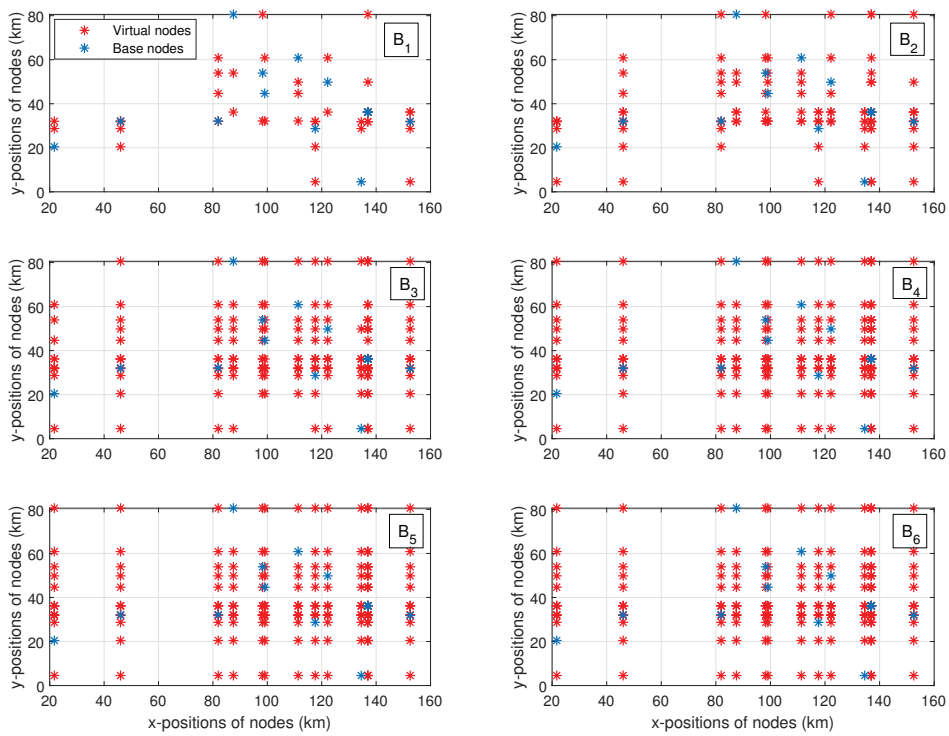


Fig. 4: The placement of the base and virtual nodes of IEEE 14-bus system for different orders of connectivity. The locations are clustered with *k*-means.

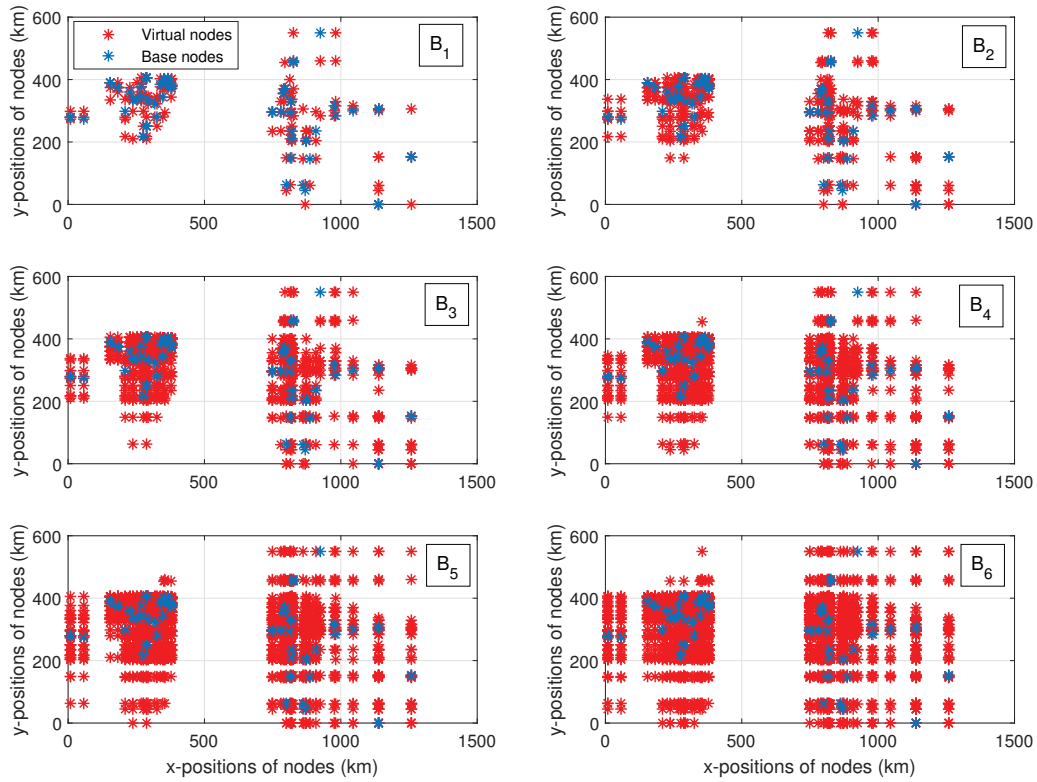


Fig. 5: The placement of the base and virtual nodes of IEEE 68-bus system for different orders of connectivity. The locations are clustered with k -means.

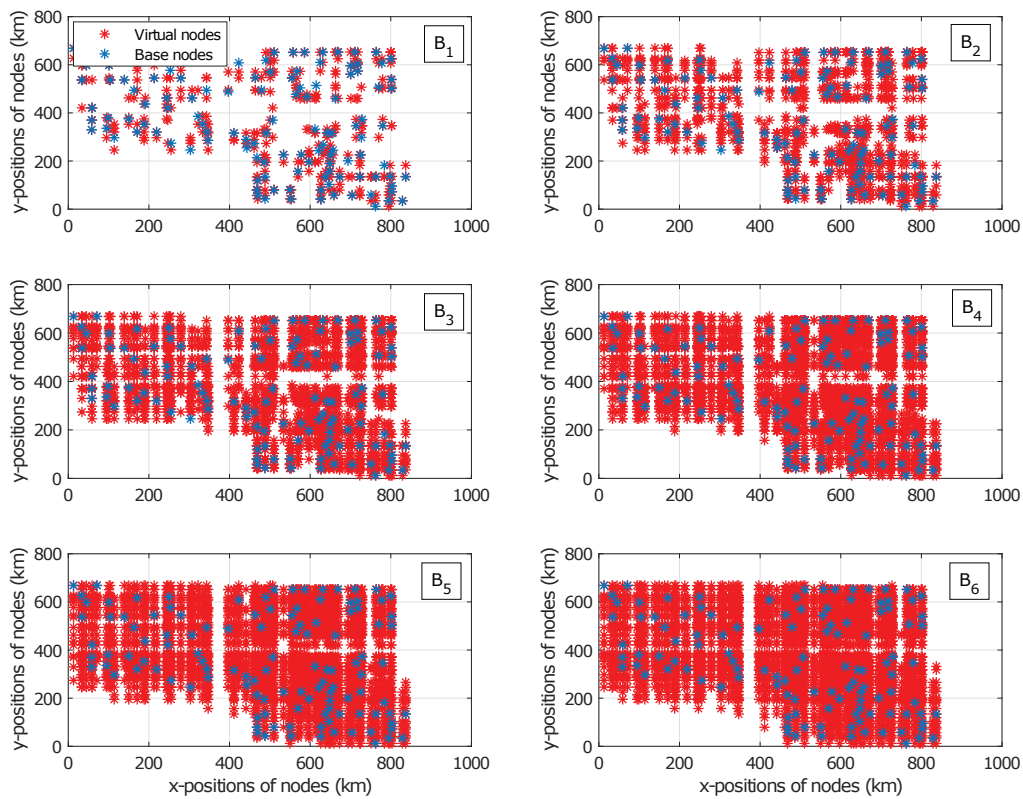


Fig. 6: The placement of the base and virtual nodes of IEEE 118-bus system for different orders of connectivity. The locations are clustered with k -means.

TABLE I-a: 14-bus system: Clusters formed by k -means clustering for different orders of connectivity with $N_g = 5$ using method 1

| Order of connectivity | Clusters with bus numbers |
|-----------------------|---|
| B_1 | {1 2 4 5 6},{9 10 11 12 13 14},{5 6 9 10 11 12 13},{1 2 3 4 5 6 7 8 9 13 14},{2 3 4 5 7 9} |
| B_2 | {1 2 3 4 5 6},{1 2 3 4 5 6 7 8 9 14},{1 2 4 5 6 7 9 10 11 12 13 14},{1 2 4 5 6 7 9 11 12 13}, {2 4 5 6 7 9 10 11 12 13 14} |
| B_3 | {1 2 3 4 5 6 7 8 9 12 13 14},{1 2 3 4 5 6 7 8 9 10 11 12 13 14}, {1 2 3 4 5 6 7 8 9 11 12 13 14},{2 3 4 6 7 8 9 10 11 12 13 14}, {1 2 3 4 5 6 7 8 9 10 11 12 13 14} |

TABLE I-b: 14-bus system: Clusters formed by k -means clustering for different orders of connectivity with $N_g = 5$ using method 2

| Order of connectivity | Clusters with bus numbers |
|-----------------------|--|
| B_1 | {5 6 9 10 11 12 13 14},{ 2 3 4 5 7 8 9 10 14},{ 1 2 3 4 5 9 13 14 },{ 1 2 4 5 6},{ 2 3 4 } |
| B_2 | { 4 5 6 7 9 10 11 12 13 14},{ 1 2 3 4 5 6 7 9 11 12 13},{ 1 2 3 4 5 6 7 8 9 10 11 13 14}, { 1 2 3 4 5 6 7 9 10 12 13 14 }, { 1 2 4 5 6 10 11 12 13 14 } |
| B_3 | { 1 2 3 4 5 6 7 9 10 11 12 13 14 },{ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 }, { 2 3 4 5 6 7 8 9 10 11 12 13 14 },{ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 }, { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 } |

TABLE II-a: 14-bus system: Clusters formed by k -means clustering for first order connectivity, B_1 with different N_g using method 1

| Number of clusters | Clusters with bus numbers |
|--------------------|---|
| $N_g = 2$ | { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 },{ 1 2 4 5 6 } |
| $N_g = 3$ | { 1 2 3 4 5 7 8 9 14 },{5 6 9 10 11 12 13 14},{1 2 4 5 6} |
| $N_g = 4$ | { 5 6 9 10 11 12 13 },{1 2 3 4 5 7 8 9 14 },{9 10 11 12 13 14 },{1 2 4 5 6} |
| $N_g = 5$ | {5 6 9 10 11 12 13},{1 2 4 5 6},{1 2 3 4},{9 10 11 12 13 14 },{2 3 4 5 7 8 9 14 } |

TABLE II-b: 14-bus system: Clusters formed by k -means clustering for first order connectivity, B_1 with different N_g using method 2

| Number of clusters | Clusters with bus numbers |
|--------------------|--|
| $N_g = 2$ | { 1 2 3 4 5 6 7 8 9 10 12 13 14 },{ 1 2 4 5 6 9 10 11 12 13 } |
| $N_g = 3$ | { 1 2 3 4 5 7 8 9 10 13 14 },{1 2 4 5 6 },{ 5 6 9 10 11 12 13 14 } |
| $N_g = 4$ | { 2 3 4 5 7 8 9 10 14 },{ 1 2 4 5 6 },{ 9 10 11 },{ 1 2 5 6 10 11 12 13 14 } |
| $N_g = 5$ | {5 6 9 10 11 12 13 14},{ 2 3 4 5 7 8 9 10 14 },{ 1 2 3 4 5 9 13 14 },{ 1 2 4 5 6 },{ 2 3 4 } |

TABLE III-a: 68-bus system: Clusters formed by k -means clustering with different orders of connectivity with $N_g = 5$ using method 1

| Order of connectivity | Clusters with bus numbers |
|-----------------------|---|
| B_1 | {1 21 22 23 24 25 26 27 28 29 58 59 61}, {1 2 8 9 27 30 31 32 33 34 35 36 37 38 39 40 41 43 44 45 46 47 48 49 52 62 63 64 65}, {40 41 42 49 52 66 67},{35 37 39 43 44 45 50 51 52 68}, {1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 53 54 55 56 57 60 } |
| B_2 | {40 41 42 45 46 48 49 50 51 52 66 67 68},{16 17 21 22 23 24 25 26 27 28 29 58 59 61}, {2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 27 36 54 55 56 57 58 59}, {1 2 3 4 5 7 8 9 14 16 17 18 21 24 25 26 27 28 29 30 31 47 53 60 61}, {1 2 3 5 7 8 9 17 25 26 27 30 31 32 33 34 35 36 37 38 39}, {40 41 42 43 44 45 46 47 48 49 50 51 52 53 62 63 64 65 66 68} |

TABLE III-b: 68-bus system: Clusters formed by k -means clustering with different orders of connectivity with $N_g = 5$ using method 2

| Order of connectivity | Clusters with bus numbers |
|-----------------------|--|
| B_1 | { 1 2 9 27 30 31 32 33 38 40 41 46 47 48 49 52 62}, { 2 3 16 17 21 22 23 24 25 26 27 28 29 53 58 59 60 61 }, { 8 9 30 32 33 34 35 36 37 38 39 43 44 45 50 51 63 64 65},{ 40 41 42 49 50 51 52 66 67 68}, { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 24 27 54 55 56 57} |
| B_2 | { 1 2 3 4 15 16 17 18 19 21 22 23 24 25 26 27 28 29 53 58 59 60 61}, {40 41 42 45 46 48 49 50 51 52 66 67 68 }, { 1 5 7 8 9 30 31 32 33 34 35 36 37 38 39 43 44 45 46 50 51 52 63 64 65}, { 1 2 3 8 9 17 25 26 27 30 31 32 33 34 35 36 38 40 41 42 45 46 47 48 49 50 51 52 53 62 63 66 68 }, {1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 24 25 26 27 30 31 36 47 53 54 55 56 57 } |

TABLE IV-a: 68-bus system: Clusters formed by k -means clustering for B_1 with different N_g using method 1

| Number of clusters | Clusters with bus numbers |
|--------------------|--|
| $N_g = 2$ | { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 53 54 55 56 57 58 59 60 61}, {1 2 8 9 27 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 62 63 64 65 66 67 68} |
| $N_g = 3$ | {1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 53 54 55 56 57 60}, {1 2 8 9 27 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 62 63 64 65 66 67 68}, {1 21 22 23 24 25 26 27 28 29 58 59 61} |
| $N_g = 4$ | {1 2 3 4 8 9 15 16 17 18 19 21 22 24 25 26 27 28 29 53 60}, {1 2 8 9 27 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 62 63 64 65 66 67 68}, {3 4 5 6 7 8 10 11 12 13 14 15 16 19 20 23 24 54 55 56 57}, {21 22 23 24 26 28 29 58 59 61} |
| $N_g = 5$ | {1 21 22 23 24 25 26 27 28 29 58 59 61}, {1 2 8 9 27 30 31 32 33 34 35 36 37 38 39 40 41 43 44 45 46 47 48 49 52 62 63 64 65}, {40 41 42 49 52 66 67},{35 37 39 43 44 45 50 51 52 68}, {1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 53 54 55 56 57 60 } |

TABLE IV-b: 68-bus system: Clusters formed by k -means clustering for B_1 with different N_g using method 2

| Number of clusters | Clusters with bus numbers |
|--------------------|--|
| $N_g = 2$ | { 1 2 8 9 27 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 62 63 64 65 66 67 68}, { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 53 54 55 56 57 58 59 60 61} |
| $N_g = 3$ | {1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 53 54 55 56 57 58 59 60 61 }, { 40 41 42 49 50 51 52 66 67 68 }, { 1 2 8 9 27 30 31 32 33 34 35 36 37 38 39 40 41 43 44 45 46 47 48 49 50 51 52 62 63 64 65} |
| $N_g = 4$ | { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 53 54 55 56 57 58 59 60 61}, { 40 41 42 49 50 51 52 66 67 68}, { 1 2 9 27 30 31 32 33 38 40 41 46 47 48 49 52 62 }, { 8 9 30 32 33 34 35 36 37 38 39 43 44 45 50 51 63 64 65} |
| $N_g = 5$ | { 1 2 9 27 30 31 32 33 38 40 41 46 47 48 49 52 62}, { 2 3 16 17 21 22 23 24 25 26 27 28 29 53 58 59 60 61 }, { 8 9 30 32 33 34 35 36 37 38 39 43 44 45 50 51 63 64 65},{ 40 41 42 49 50 51 52 66 67 68}, { 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 24 27 54 55 56 57} |

The effects of the number of groups on the formation of clusters are shown in Table II. The clusters are made for the first order of connectivity. It is clear from the table that with the increase in the number of clusters, the sizes of the clusters decrease and that makes the whole process faster.

A similar analysis is found for IEEE 68 and 118-bus test system from Figs. 5, 6, and Tables III, IV.

B. Distributed Estimation of 14-bus System

The distributed estimation is applied on the steady state operation for the 14-bus system. As the system does not change over time, only one sample is estimated. The steady state results are acquired using MATPOWER [32]. The actual, and estimated values by the centralized and distributed estimators with different methods for B_1 connectivity and $N_g = 5$ are shown in Table V.

From the table, it is seen that both methods perform well in comparison to the centralized estimator. However, it is not possible to yield a more accurate result than the centralized WLS estimator, as it is the most accurate one with Gaussian noise.

C. Distributed Estimation of 68-bus System

For 68-bus system, the simulation is executed for two different cases with 150 samples of measurements over five seconds, one with some random dynamic changes, another with a large disturbance in the system. The measurements are taken from Real-Time Digital Simulator. The total number of iteration is set to ten. The estimated results are exchanged every five iterations.

a) Case I: In this case, the change in the system is taken with pseudo-random binary signals (PRBS) applied at the excitation systems of the generators. This exhibits the normal load change behavior of the system. Five clusters are made with both methods using the first and second order connectivity as shown in Table III. The norm of the residues for 150 samples of data are shown in Fig. 7.

b) Case II: In case II, a three phase to ground fault is simulated at bus 8 of the 68-bus system. The fault occurs at time $t = 1s$. Similar to case I, five clusters are made with both methods and 150 samples are taken over five seconds with a higher noise level. The norms of the residues are shown in Fig. 8.

For both cases, it can be seen that method 2 with the first order connectivity has an overall low norm compared to other methods and connectivity. As a result, it performs better than others in terms of accuracy.

The mean absolute error (MAE) of the magnitude and angles of the two cases are shown in Table VI. The MAE for state i is defined as follows,

$$MAE(i) = \frac{1}{N_s} \sum_{t=1}^{N_s} |x_i(t) - \hat{x}_i(t)| \quad (7)$$

where, N_s represents the number of samples.

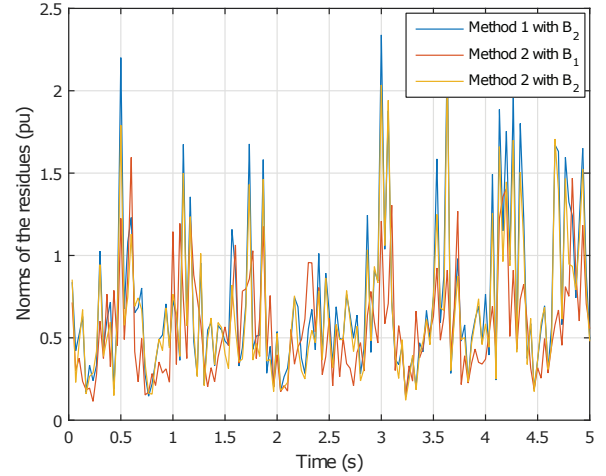


Fig. 7: The norms of the residues for the distributed state estimation of the 68-bus system with PRBS signals at the excitation of the generators.

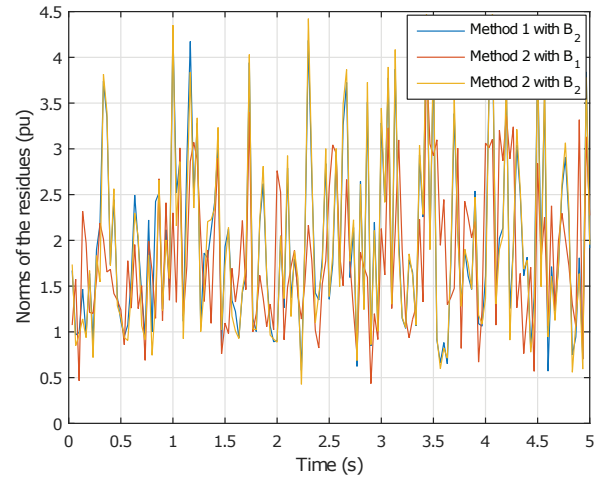


Fig. 8: The norms of the residues for the distributed state estimation of the 68-bus system with a three phase line to ground fault at bus 8.

V. ANALYSIS OF THE SIMULATION RESULTS

In this section, two important characteristics of an estimator, accuracy and time, are analyzed for the simulation results found in Section IV.

A. Accuracy

The accuracy of the estimators on the 68-bus system for different number of clusters are shown in Figs. 9, 10, and 11. The 150 samples of case II is chosen for the comparison. When the number of cluster is one, it refers to the centralized estimation. From the figures, it can be seen that the norms of the residues increase with the increase in the number of

TABLE V: The actual and estimated values of the angles and magnitudes of 14-bus systems using different clustering methods ($N_g = 5, B_1$).

| Bus No. | θ_{act} | Centralized | Method 1 | Method 2 | V_{act} | Centralized | Method 1 | Method 2 |
|---------|----------------|-------------|-----------|-----------|-----------|-------------|----------|----------|
| 1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |
| 2 | -0.086970 | -0.085244 | -0.087449 | -0.085253 | 0.985849 | 0.986143 | 0.985646 | 0.986139 |
| 3 | -0.222093 | -0.221732 | -0.224030 | -0.221859 | 0.952830 | 0.953042 | 0.952413 | 0.952992 |
| 4 | -0.179996 | -0.178975 | -0.180541 | -0.179085 | 0.960377 | 0.960562 | 0.960090 | 0.960510 |
| 5 | -0.153135 | -0.152093 | -0.153238 | -0.151886 | 0.962264 | 0.962398 | 0.962130 | 0.962429 |
| 6 | -0.248203 | -0.248478 | -0.248655 | -0.248482 | 1.009434 | 1.009169 | 1.009128 | 1.009023 |
| 7 | -0.233176 | -0.233158 | -0.233917 | -0.232717 | 1.001887 | 1.001764 | 1.000970 | 1.001901 |
| 8 | -0.233176 | -0.233158 | -0.233793 | -0.232717 | 1.028302 | 1.027898 | 1.027679 | 1.028046 |
| 9 | -0.260735 | -0.260665 | -0.261679 | -0.259635 | 0.996226 | 0.996055 | 0.995548 | 0.996176 |
| 10 | -0.263492 | -0.263411 | -0.264073 | -0.263309 | 0.991509 | 0.991409 | 0.991129 | 0.991320 |
| 11 | -0.258152 | -0.258296 | -0.258819 | -0.258468 | 0.997170 | 0.996885 | 0.996722 | 0.996620 |
| 12 | -0.263126 | -0.263174 | -0.263736 | -0.263234 | 0.995283 | 0.994975 | 0.994895 | 0.994802 |
| 13 | -0.264522 | -0.264477 | -0.265063 | -0.263383 | 0.990566 | 0.990312 | 0.990190 | 0.990425 |
| 14 | -0.279846 | -0.279897 | -0.280559 | -0.278840 | 0.977358 | 0.977160 | 0.976864 | 0.977304 |

clusters. It means that the accuracy reduces with the number of clusters.

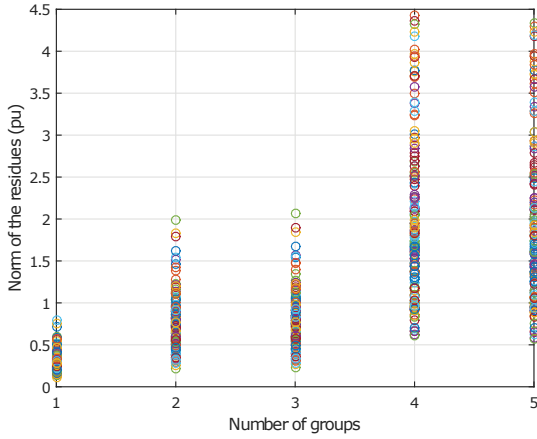


Fig. 9: The accuracy of the centralized and distributed estimation for 150 samples with clusters formed by method 1 with second order connectivity.

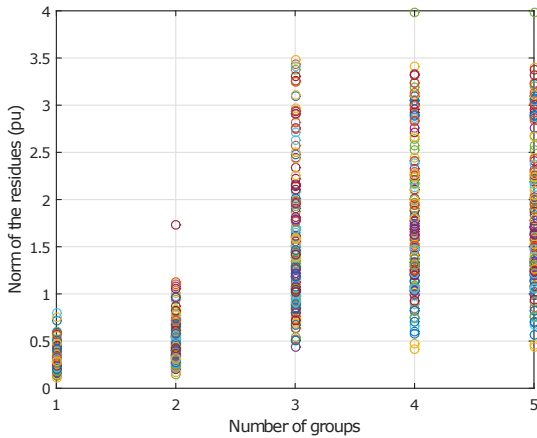


Fig. 10: The accuracy of the centralized and distributed estimation for 150 samples with different number of clusters formed by method 2 with first order connectivity.

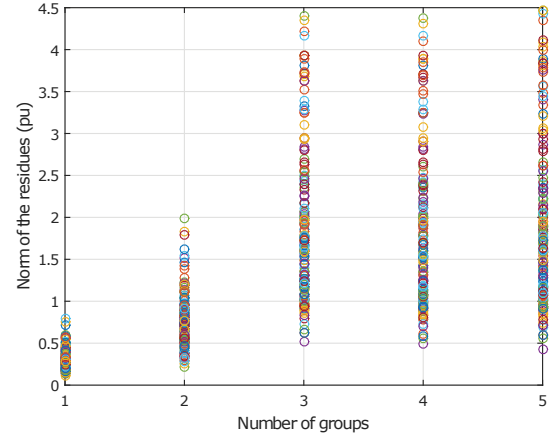


Fig. 11: The accuracy of the centralized and distributed estimation for 150 samples with different number of clusters formed by method 2 with second order connectivity.

The reason behind this behavior is the effects of the reduced number of measurements in the clusters. The centralized estimator takes all measurements at a time and gets a better Probability Density Function (PDF) for the Gaussian noise. On the other hand, in the distributed estimation, each cluster works on a part of the measurement set, and the PDF becomes less similar to the Gaussian nature. Though the exchange process improves the results to some extent, it cannot recover the accuracy completely.

It should be kept in mind that, though the norms of the residues of the distributed systems are several times that of the centralized estimator, the difference between the actual value and the estimated values do not differ much as shown in Table VI. It is due to the fact that, the norms change a lot for a small change in the estimated values. As a result, the accuracy of the distributed estimation can be considered as acceptable.

B. Required Time

The required time by different number of clusters is shown in Figs. 12 to 15. The first three figures are for IEEE 68-bus system, where the last one is for 118-bus system. The simulation is run on an Intel Core 2 Duo 2.2GHz processor

TABLE VI: The mean absolute errors of the estimated value and the actual value of the magnitudes and angles of 68 buses using method 1 ($N_g = 5, B_1$).

| Bus No. | Case I | | Case II | |
|---------|----------|----------|----------|----------|
| | θ | $ V $ | θ | $ V $ |
| 1 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 2 | 0.000013 | 0.000978 | 0.000902 | 0.000510 |
| 3 | 0.000000 | 0.000931 | 0.001057 | 0.000567 |
| 4 | 0.000073 | 0.001166 | 0.000255 | 0.000935 |
| 5 | 0.000117 | 0.001629 | 0.000183 | 0.001320 |
| 6 | 0.000130 | 0.000699 | 0.000031 | 0.000214 |
| 7 | 0.000113 | 0.001690 | 0.000434 | 0.001458 |
| 8 | 0.000121 | 0.001703 | 0.000428 | 0.001440 |
| 9 | 0.000162 | 0.001778 | 0.000364 | 0.001175 |
| 10 | 0.000167 | 0.000845 | 0.000609 | 0.000277 |
| 11 | 0.000148 | 0.000952 | 0.000696 | 0.000106 |
| 12 | 0.000165 | 0.000855 | 0.000493 | 0.000335 |
| 13 | 0.000142 | 0.000167 | 0.000140 | 0.000188 |
| 14 | 0.000016 | 0.002005 | 0.000689 | 0.000958 |
| 15 | 0.000324 | 0.000356 | 0.001341 | 0.000053 |
| 16 | 0.000127 | 0.000591 | 0.000805 | 0.000342 |
| 17 | 0.000056 | 0.000214 | 0.000837 | 0.000073 |
| 18 | 0.000094 | 0.001558 | 0.001318 | 0.000833 |
| 19 | 0.000203 | 0.001039 | 0.001402 | 0.000108 |
| 20 | 0.000218 | 0.001334 | 0.001474 | 0.000214 |
| 21 | 0.000179 | 0.001090 | 0.000959 | 0.000126 |
| 22 | 0.000199 | 0.000001 | 0.002603 | 0.000586 |
| 23 | 0.000191 | 0.000217 | 0.002610 | 0.000709 |
| 24 | 0.000035 | 0.000242 | 0.000803 | 0.000282 |
| 25 | 0.000025 | 0.001105 | 0.001226 | 0.000352 |
| 26 | 0.000082 | 0.001116 | 0.000707 | 0.000020 |
| 27 | 0.000035 | 0.000059 | 0.000610 | 0.000048 |
| 28 | 0.000223 | 0.003942 | 0.001235 | 0.000079 |
| 29 | 0.000202 | 0.003613 | 0.001673 | 0.000139 |
| 30 | 0.000460 | 0.000705 | 0.000428 | 0.000768 |
| 31 | 0.000190 | 0.000274 | 0.000226 | 0.000490 |
| 32 | 0.000560 | 0.000223 | 0.001491 | 0.001041 |
| 33 | 0.000448 | 0.001153 | 0.002433 | 0.001111 |
| 34 | 0.000484 | 0.002012 | 0.005201 | 0.000985 |
| 35 | 0.000448 | 0.001987 | 0.005167 | 0.001024 |
| 36 | 0.000029 | 0.002220 | 0.005184 | 0.000498 |
| 37 | 0.000235 | 0.001615 | 0.003642 | 0.000721 |
| 38 | 0.000201 | 0.000009 | 0.000201 | 0.000590 |
| 39 | 0.000160 | 0.003472 | 0.005098 | 0.000705 |
| 40 | 0.000136 | 0.000211 | 0.000006 | 0.000393 |
| 41 | 0.000145 | 0.001068 | 0.000304 | 0.002817 |
| 42 | 0.000164 | 0.001536 | 0.001983 | 0.001621 |
| 43 | 0.000217 | 0.002522 | 0.004436 | 0.000665 |
| 44 | 0.000214 | 0.002553 | 0.004439 | 0.000675 |
| 45 | 0.000159 | 0.001705 | 0.010978 | 0.004562 |
| 46 | 0.000143 | 0.000046 | 0.002983 | 0.000230 |
| 47 | 0.000102 | 0.000147 | 0.000074 | 0.000936 |
| 48 | 0.000098 | 0.000119 | 0.000726 | 0.000616 |
| 49 | 0.000060 | 0.002955 | 0.002767 | 0.000598 |
| 50 | 0.000184 | 0.001857 | 0.004737 | 0.002755 |
| 51 | 0.000188 | 0.001859 | 0.008884 | 0.004378 |
| 52 | 0.000169 | 0.001832 | 0.002237 | 0.000994 |
| 53 | 0.000039 | 0.001298 | 0.000840 | 0.000855 |
| 54 | 0.000153 | 0.000707 | 0.000039 | 0.000208 |
| 55 | 0.000148 | 0.000840 | 0.000591 | 0.000278 |
| 56 | 0.000213 | 0.001018 | 0.001386 | 0.000115 |
| 57 | 0.000235 | 0.001312 | 0.001491 | 0.000224 |
| 58 | 0.000180 | 0.000024 | 0.002608 | 0.000577 |
| 59 | 0.000193 | 0.000208 | 0.002627 | 0.000712 |
| 60 | 0.000165 | 0.002738 | 0.000810 | 0.001261 |
| 61 | 0.000235 | 0.003623 | 0.001689 | 0.000154 |
| 62 | 0.000183 | 0.000226 | 0.000166 | 0.000511 |
| 63 | 0.000559 | 0.000235 | 0.001475 | 0.001056 |
| 64 | 0.000158 | 0.002649 | 0.005030 | 0.000686 |
| 65 | 0.000237 | 0.001621 | 0.003622 | 0.000736 |
| 66 | 0.000143 | 0.001045 | 0.000328 | 0.002819 |
| 67 | 0.000078 | 0.000430 | 0.001197 | 0.003703 |
| 68 | 0.000158 | 0.001815 | 0.002256 | 0.000973 |

with 4GB of memory. The effectiveness of the distributed estimation can be realized from the figures. There exist some outliers which are caused by the background processes of the computing machine. It is not unusual to have these outliers in a generic computer. If the outliers are excluded, it can be seen that for all methods, the execution time is significantly less compared to the centralized one.

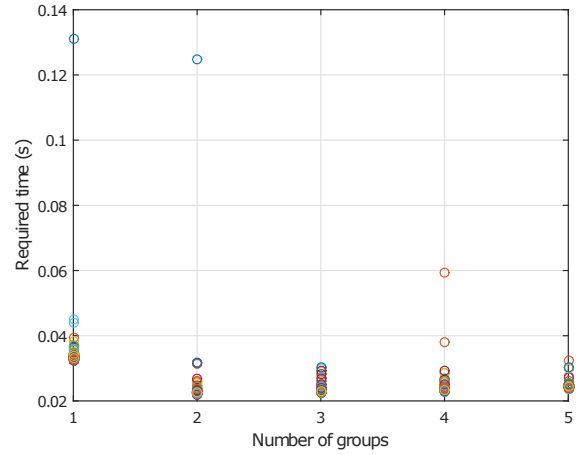


Fig. 12: The required time for 150 samples by different number of clusters formed by method 1 with second order connectivity.

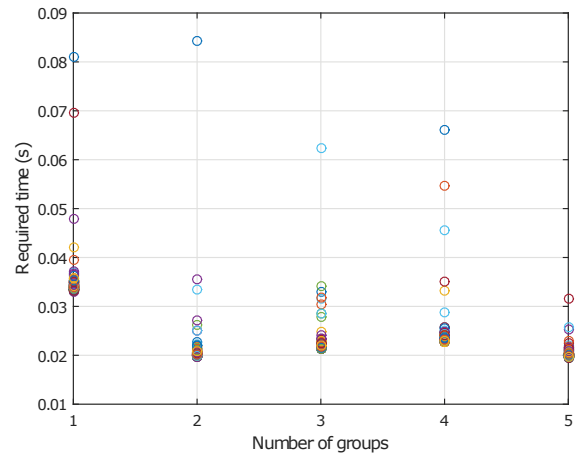


Fig. 13: The required time for 150 samples by different number of clusters formed by method 2 with first order connectivity.

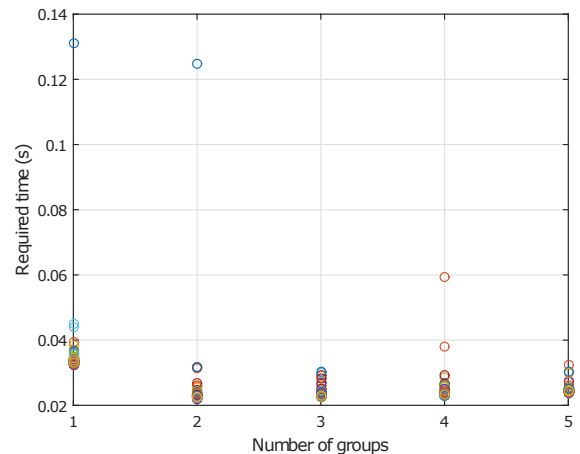


Fig. 14: The required time for 150 samples by different number of clusters formed by method 2 with second order connectivity.

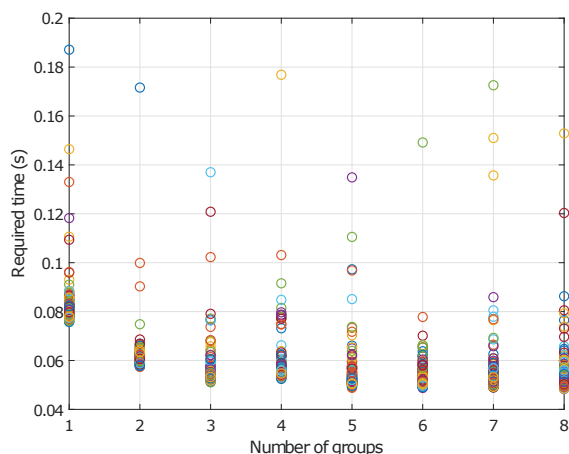


Fig. 15: The required time for 150 samples of IEEE 118-bus test system by different number of clusters formed with method 2 and first order connectivity.

However, it can also be observed that the execution time did not reduce with the increased number of clusters. This is due to the fact that the execution time does not depend on the number of clusters, rather on the size of the largest cluster. As the clusters run in parallel in the distributed estimation, the overall execution time is determined by the slowest cluster i.e. the largest cluster.

VI. CONCLUSION

In this paper, a new graph called sparse matrix is proposed and its advantages are demonstrated to automatically cluster the electric power network. Two different methods are mixed with different orders of connectivity to find the best candidate for the matrix. It is applied on the state estimation of IEEE 14, 68, and 118-bus systems under different conditions and the computations are made faster than the existing centralized method. In reality, the system may include thousands of buses and a distributed estimation can make it fast enough for real-time operations.

From analytic and simulation results, it is shown that the sparse matrix graph can be effective in making coherent clusters of any networked system. Even though it is applied on a specific application of power system transmission network in this paper, it can also be applied to a varied range of applications with sparse networked systems. In future, the graph can be grouped with other clustering algorithms like mean-shift clustering or agglomerative hierarchical clustering. Some deep learning techniques can also be applied to find an improved results.

REFERENCES

- [1] S. Zhai and T. Jiang, "A novel sense-through-foliage target recognition system based on sparse representation and improved particle swarm optimization-based support vector machine," *Measurement*, vol. 46, no. 10, pp. 3994–4004, 2013.
- [2] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear programming and network flows*. New York: J. Wiley, 1977.
- [3] V. Satuluri and S. Parthasarathy, "Scalable graph clustering using stochastic flows: Applications to community discovery," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009, pp. 737–746.
- [4] S. Shamshirband, A. Amini, N. B. Anuar, M. L. M. Kiah, Y. W. Teh, and S. Furnell, "D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks," *Measurement*, vol. 55, pp. 212–226, 2014.
- [5] S. E. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, 2007.
- [6] G. W. Flake, R. E. Tarjan, and K. Tsioutsoulouklis, "Graph clustering and minimum cut trees," *Internet Mathematics*, vol. 1, no. 4, pp. 385–408, 2004.
- [7] S. Van Dongen, "Graph clustering via a discrete uncoupling process," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 1, pp. 121–141, 2008.
- [8] M. J. Rattigan, M. Maier, and D. Jensen, "Graph clustering with network structure indices," in *Proceedings of the 24th International Conference on Machine Learning*. ACM, 2007, pp. 783–790.
- [9] H. Bunke, P. Foggia, C. Guidobaldi, and M. Vento, "Graph clustering using the weighted minimum common supergraph," in *International Workshop on Graph-Based Representations in Pattern Recognition*. Springer, 2003, pp. 235–246.
- [10] J. Edachery, A. Sen, and F. J. Brandenburg, "Graph clustering using distance-k cliques," in *International Symposium on Graph Drawing*. Springer, 1999, pp. 98–106.
- [11] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: A kernel approach," *Machine Learning*, vol. 74, no. 1, pp. 1–22, 2009.
- [12] H. A. Schildt and J. T. Mattsson, "A dense network sub-grouping algorithm for co-citation analysis and its implementation in the software tool Sitkis," *Scientometrics*, vol. 67, no. 1, pp. 143–163, 2006.
- [13] Y. Chircota, F. Jourdan, and G. Melançon, "Software components capture using graph clustering," in *11th IEEE International Workshop on Program Comprehension, 2003*. IEEE, 2003, pp. 217–226.
- [14] Y.-W. Huang, N. Jing, and E. A. Rundensteiner, "Effective graph clustering for path queries in digital map databases," in *Proceedings of the Fifth International Conference on Information and Knowledge Management*. ACM, 1996, pp. 215–222.
- [15] V. G. Kaburlasos, V. Spais, V. Petridis, L. Petrou, S. Kazarlis, N. Maslaris, and A. Kallinakos, "Intelligent clustering techniques for prediction of sugar production," *Mathematics and Computers in Simulation*, vol. 60, no. 3, pp. 159–168, 2002.
- [16] A. H. Ahmadabadian, R. Yazdan, A. Karami, M. Moradi, and F. Ghorbani, "Clustering and selecting vantage images in a low-cost system for 3D reconstruction of texture-less objects," *Measurement*, vol. 99, pp. 185–191, 2017.
- [17] Y. Cui, J. Shi, and Z. Wang, "Analog circuit fault diagnosis based on Quantum Clustering based Multi-valued Quantum Fuzzification Decision Tree (QC-MQFDT)," *Measurement*, vol. 93, pp. 421–434, 2016.
- [18] L. Liang, F. Liu, M. Li, K. He, and G. Xu, "Feature selection for machine fault diagnosis using clustering of non-negation matrix factorization," *Measurement*, vol. 94, pp. 295–305, 2016.
- [19] A. K. Tung, J. Hou, and J. Han, "Spatial clustering in the presence of obstacles," in *Data Engineering, 2001. Proceedings. 17th International Conference on*. IEEE, 2001, pp. 359–367.
- [20] F. C. Schweppe and J. Wildes, "Power system static-state estimation, Part I: Exact model," *IEEE Trans. Power Apparatus and Systems*, vol. 1, pp. 120–125, Jan 1970.
- [21] F. C. Schweppe and D. B. Rom, "Power system static-state estimation, Part II: Approximate model," *IEEE Trans. Power Apparatus and Systems*, vol. 1, pp. 125–130, Jan 1970.
- [22] W. Hackbusch, "A sparse matrix arithmetic based on \mathcal{H} -matrices. Part I: Introduction to \mathcal{H} -matrices," *Computing*, vol. 62, no. 2, pp. 89–108, Apr 1999.
- [23] D. M. Falcao, F. F. Wu, and L. Murphy, "Parallel and distributed state estimation," *IEEE Transactions on Power Systems*, vol. 10, no. 2, pp. 724–730, 1995.
- [24] M. A. Rahman and G. K. Venayagamoorthy, "Convergence of the fast state estimation for power systems," *SAIEE Research Journal*, vol. 108, pp. 117–127, 2017.
- [25] L. Xie, D. H. Choi, S. Kar, and H. Poor, "Fully distributed state estimation for wide-area monitoring systems," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1154–1169, Sept 2012.
- [26] W. Liu, L. Liu, D. A. Cartes, and G. K. Venayagamoorthy, "Binary particle swarm optimization based defensive islanding of large scale power systems," *International Journal of Computer Science Applications*, vol. 4, no. 3, pp. 69–83, 2007.
- [27] K. Tang and G. K. Venayagamoorthy, "Adaptive inter-area oscillation damping controller for multi-machine power systems," *Electric Power Systems Research*, vol. 134, pp. 105–113, 2016.

- [28] G. K. Venayagamoorthy and K. J. Makasa, "Online voltage stability monitoring with wind farms and electric vehicles in a smart grid," in *Proceedings of the 20th International Conference on Electrical Engineering*, 2014, pp. 1770–1774.
- [29] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Systems*, vol. 28, no. 2, pp. 1617–1626, May 2013.
- [30] A. Abur and A. G. Exposito, *Power System State Estimation: Theory and Implementation*. New York: Marcel Dekker Inc., 2004.
- [31] A. Monticelli, *State estimation in electric power systems: A generalized approach*. Massachusetts: Kluwer Academic Pub., 1999.
- [32] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.
- [33] B. Chaudhuri, R. Majumder, and B. C. Pal, "Wide-area measurement-based stabilizing control of power system considering signal transmission delay," *IEEE Trans. Power Systems*, vol. 19, no. 4, pp. 1971–1979, Nov 2004.



Md Ashfaqur Rahman is currently serving as a Visiting Assistant Professor at Texas A&M University - Kingsville. He received his Ph.D. from the department of Electrical and Computer Engineering of Clemson University in 2017. Before that he received his Master of Science from Texas Tech University in 2012, and Bachelor of Science from Bangladesh University of Engineering and Technology in 2009. He has served as a lecturer of electrical engineering for around two years. He has also served as a research assistant at Clemson University and Texas

Tech University. His research interest is in the broad area of power system state estimation with an emphasis on cyber security and distributability of different estimation methods. He published two journal articles and eight conference papers with a total of 210 citations. He has been a reviewer of IEEE and Elsevier journals, and conference papers. He is a member of IEEE.



Ganesh Kumar Venayagamoorthy (S'91-M'97-SM'02) is the Duke Energy Distinguished Professor of Power Engineering and Professor of Electrical and Computer Engineering and Automotive Engineering at Clemson University. Prior to that, he was a Professor of Electrical and Computer Engineering at the Missouri University of Science and Technology (Missouri S&T), Rolla, USA from 2002 to 2011. Dr. Venayagamoorthy is the Founder (2004) and Director of the Real-Time Power and Intelligent Systems Laboratory (<http://rtpis.org>). He holds an

Honorary Professor position in the School of Engineering at the University of Kwazulu-Natal, Durban, South Africa. Dr. Venayagamoorthy received his Ph.D. degree in electrical engineering from the University of Natal, Durban, South Africa. He holds an MBA degree in Entrepreneurship and Innovation from Clemson University, SC.

Dr. Venayagamoorthy's interests are in the research, development and innovation of smart grid technologies and operations, including intelligent sensing and monitoring, intelligent systems, integration of renewable energy sources, power system optimization, stability and control, and signal processing. He has published over 500 refereed technical articles. His publications are cited over 15,000 times with a h-index of 60. Dr. Venayagamoorthy has been involved in over 70 sponsored projects in excess of US \$10 million. Dr. Venayagamoorthy has given over 300 invited keynotes, plenaries, presentations, tutorials and lectures in over 40 countries to date.

Dr. Venayagamoorthy is involved in the leadership and organization of many conferences including the General Chair of the Annual Power System Conference (Clemson, SC, USA) since 2013, and Pioneer and Chair/co-Chair of the IEEE Symposium of Computational Intelligence Applications in Smart Grid (CIASG) since 2011. He is currently the Chair of the IEEE PES Working Group on Intelligent Control Systems, and the Founder and Chair of IEEE Computational Intelligence Society (CIS) Task Force on Smart Grid. Dr. Venayagamoorthy has served as Editor/Guest Editor of several IEEE Transactions and Elsevier Journals. Dr. Venayagamoorthy is a Senior Member of the IEEE, and a Fellow of the IET, UK, and the SAIEE.