A DISTRIBUTED STATE ESTIMATOR FOR ELECTRIC POWER SYSTEMS

S.-Y. Lin
Department of Control Engineering
National Chiao Tung University
Hsinchu, TAIWAN, R.O.C.

Abstract - In this paper, we develop a theoretically robust and computationally efficient distributed state estimator, which is to solve the WLS problem by using distributed computation, for the power system. This distributed state estimator aims at its utilization in the decentralized control, and it is executing in a data communication network which is assumed to be topologically the same and physically in parallel with the power network. Along with this state estimator, we can obtain several attractive satellite functions which include (1) reduction of the time-skew problem, (2) being free from the power network topological error, (3) easy identification of the unobservable states and (4) bad data detection and identification. We have analyzed the computation complexity of this distributed state estimator. Moreover, we have also simulated this state estimator on several cases of the IEEE 30-bus system. The numerical accuracy of the simulation results are satisfactory, and the estimated computation time including the communication delay demonstrates the excellent computational performance of the distributed state estimator.

INTRODUCTION

There has been a long history in the development of power system state estimators since 1970 when Schweppe and his colleague first posed the problem [1]. Typical state estimators for centralized control were summarized in [3] by Bose and Clements. Evolving with the hierarchical control strategies of power systems, several state estimators for hierarchical control were developed (e.g., [4],[5]), and they were discussed in [6] by Van Cutsem and Ribbens-Pavella. Nowadays, the decentralized control technologies has made continual progress [8]-[10]. Although the decentralized control techniques have not yet applied to power system, however, the trend is unavoidable because the decentralized control has obvious advantages over the centralized and hierarchical control for the large-scale interconnected systems [10]. This draws enough attraction to develop the distributed state estimator.

This paper presents a theoretically robust and computationally efficient distributed state estimator under the assumption of the existence of a high speed data communication network. The proposed *distributed state estimator* is to solve the weighted least square (WLS) state estimation problem by using *distributed computation.* We begin by combining the recursive quadratic programming with the dual method (RQPD) to solve the WLS problem [11]. This RQPD method is robust because of its global convergence property. It is also computationally efficient due to its parallel computation nature. Furthermore, because of its complete decomposition property, it can be executed in a data communication network by distributed computation.

The above mentioned data communication network is assumed to be topologically the same and physically in parallel with the power network. The assumption is justified because (a) the cost of computer and memory continues to decline, and (b) the availability of data communication network and the technology of optical fiber transmission keep evolving. These two factors, pointed out in [13] by Gaushell and Darlington, will accelerate the trend toward the distributed processing of power system. Moreover, a trial of integrating the communication network and the power system is ongoing [14].

Along with the developed distributed state estimator, we can obtain several attractive satellite functions which include the reduction of the time-skew problem, the topological error free power network model, the easy identification of the unobservable states and the bad data detection and identification. Details of these functions are described in this paper.

We have analyzed the computation complexity of the developed distributed state estimator. The result shows that because of its inherent property of parallel computation and the nature of sparse-matrix technique, this state estimator achieves tremendous computational time saving. We have also simulated this state estimator on several cases of the IEEE 30-bus system. The simulation results are satisfactory, and the estimated time which includes the computation time and the communication delay outplays the conventional WLS state estimator by the Newton method [2].

The concept of distributed processing had been applied for state estimation by Zaborszky et. al., in [15] and Brice and Kavin in [16]. However, Our distributed state estimator differs from the ultra fast state estimator in [15] because their estimation scheme is mainly carried out in the control center. Although the estimator based on relaxation method in [16] can be considered as a distributed state estimator, it is not clear how their distributed computation is arranged.

WLS STATE ESTIMATION PROBLEM

Mathematically, the static-state estimation problem of an n-bus power system is described below as a WLS problem

$$\min_x \eta^T R^{-1} \eta$$

subject to:

$$z = h(x) + \eta \tag{1}$$

where

x: $2n$ dimensional vector of state variables—voltage magnitude v and phase angle θ,

z: m dimensional vector of measurements,

h: m nonlinear measurement functions,

η: m dimensional random vector of measurement errors,

R: $m \times m$ diagonal covariance matrix of measurement errors η.

THE RQPD METHOD

In order to solve (1) by the distributed computation, we need a method with property of decomposition up to bus level. From here on, such decomposition property will be addressed as complete decomposition. For the consideration of complete decomposition, the dual method is a good candidate but it requires that the problem to be solved should be separable and has a positive definite Hessian matrix [17]. Clearly, (1) is highly nonlinear and nonseparable. But the quadratic approximate subproblem of (1), at any k-th iteration of the recursive quadratic programming method, is separable as shown in (2) [18].

91 SM 463-0 PWRS A paper recommended and approved by the IEEE Power System Engineering Committee of the IEEE Power Engineering Society for presentation at the IEEE/PES 1991 Summer Meeting, San Diego, California July 28 - August 1, 1991. Manuscript submitted January 22, 1991; made available for printing June 5, 1991.

$$\min_{dx^k} \sum_{i=1}^{n} \eta_i^T r_i^{-1} \eta_i + \gamma (dx_i^k)^T dx_i^k$$

subject to:

$$z_i - h_i^k - H_{ii}^k dx_i^k - \sum_{l \in J_i} H_{il}^k dx_l^k - \eta_i = 0$$

$$i = 1, 2, \cdots, n \tag{2}$$

where

z_i: the measurements made at bus i such as the bus injection power or voltage magnitude at bus i, or the line flows from bus i,

h_i^k: $= h_i(x^k)$, a subset of $h(x^k)$ corresponding to z_i,

x_i^k: the values of the states x_i of bus i at the k-th iteration of the recursive quadratic programming method,

H_{ii}^k: $\partial h_i^k / \partial x_i$,

H_{il}^k: $\partial h_i^k / \partial x_l$,

J_i: the set of buses directly connected to bus i in the real-time power network,

η_i: subvector of η corresponding to z_i,

r_i: diagonal submatrix of R corresponding to η_i,

γ: weighting constant, a positive real number,

dx^k: $= (dx_1^k, dx_2^k, \cdots, dx_n^k)$, the variables of (2).

This separable quadratic approximate subproblem (2) has a positive definite Hessian matrix. Hence, it is ideally suited to the dual method to achieve the complete decomposition.

Therefore, a new iterative method which combines the recursive quadratic programming with the dual method (RQPD) to solve (1) can be described below:

$$x^{k+1} = x^k + \bar{\beta}^k (dx^k)^* \tag{3}$$

where $(dx^k)^*$ is the optimal solution of (2) which will be solved by the dual method, and the stepsize $\bar{\beta}^k$ is determined by

$$\bar{\beta}^k = \arg\{\min_{\beta} [z - h(x^k + \beta(dx^k)^*)]^T R^{-1} [z - h(x^k + \beta(dx^k)^*)]\}.$$

In the following, we will illustrate how the dual method solve (2) to achieve the complete decomposition.

The dual problem of (2) is described below:

$$\max_{\lambda} \Phi(\lambda) \tag{4}$$

where

$$\Phi(\lambda) = \min_{dx^k, \eta} \sum_{i=1}^{n} \{\eta_i^T r_i^{-1} \eta_i + \gamma (dx_i^k)^T dx_i^k + \lambda_i^T \mathcal{H}_i(dx_i^k, dx_l^k)\} \tag{5}$$

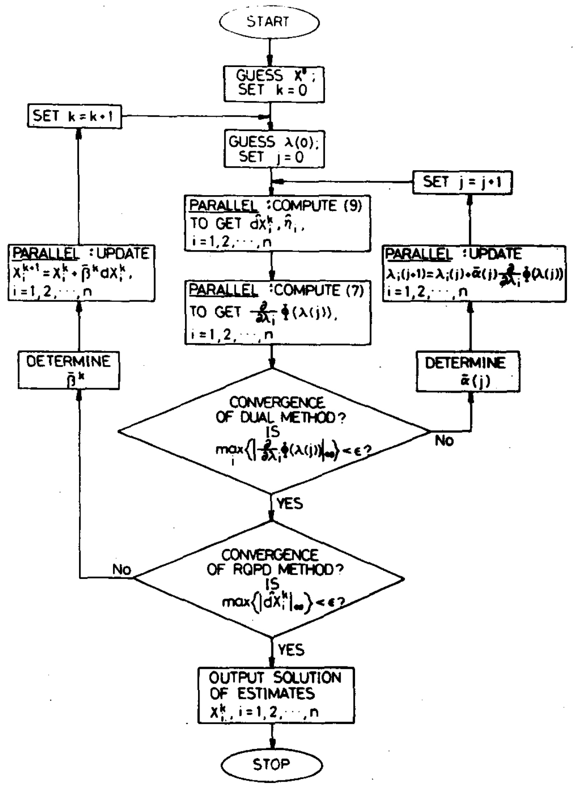
is the dual function; λ is the vector of Lagrange multipliers with appropriate dimension, and

$$\mathcal{H}_i(dx_i^k, dx_l^k) = z_i - h_i^k - H_{ii}^k dx_i^k - \sum_{l \in J_i} H_{il}^k dx_l^k - \eta_i.$$

The dual method for solving (2) is to use the gradient ascent method to solve (4) [17]. Its iterative procedure is simply:

$$\lambda(j+1) = \lambda(j) + \bar{\alpha}(j) \frac{\partial}{\partial \lambda} \Phi(\lambda(j)) \tag{6}$$

where the stepsize $\bar{\alpha}(j)$ of iteration j is determined by $\bar{\alpha}(j) = \arg\{\max_{\alpha \geq 0} \Phi(\lambda(j) + \alpha \frac{\partial}{\partial \lambda} \Phi(\lambda(j)))\}$; and the gradient $\frac{\partial}{\partial \lambda} \Phi(\lambda(j))$ are evaluated by [17]



START
GUESS X^0; SET $k=0$
SET $k=k+1$
GUESS $\lambda(0)$; SET $j=0$
SET $j=j+1$
PARALLEL : COMPUTE (9) TO GET $\hat{dX}_i^k, \hat{\eta}_i$, $i=1,2,\cdots,n$
PARALLEL : UPDATE $X_i^{k+1}=X_i^k+\bar{\beta}^k dX_i^k$, $i=1,2,\cdots,n$
PARALLEL : UPDATE $\lambda_i(j+1)=\lambda_i(j)+\bar{\alpha}(j)\frac{\partial}{\partial\lambda_i}\Phi(\lambda(j))$ $i=1,2,\cdots,n$
PARALLEL : COMPUTE (7) TO GET $\frac{\partial}{\partial\lambda_i}\Phi(\lambda(j))$, $i=1,2,\cdots,n$
DETERMINE $\bar{\beta}^k$
DETERMINE $\bar{\alpha}(j)$
CONVERGENCE OF DUAL METHOD? IS $\max_i\{|\frac{\partial}{\partial\lambda_i}\Phi(\lambda(j))|_\infty\}<\epsilon$?
No
YES
CONVERGENCE OF RQPD METHOD? IS $\max\{|\hat{dX}_i^k|_\infty\}<\epsilon$?
No
YES
OUTPUT SOLUTION OF ESTIMATES X_i^k, $i=1,2,\cdots,n$
STOP

Figure 1: flow chart of the RQPD method

$$\frac{\partial}{\partial \lambda_i} \Phi(\lambda(j)) = z_i - h_i^k - H_{ii}^k \hat{dx}_i^k - \sum_{l \in J_i} H_{il}^k \hat{dx}_l^k - \hat{\eta}_i$$

$$\text{for each } i = 1, 2, \cdots, n \tag{7}$$

The $\hat{dx}_i^k, \hat{\eta}_i, i = 1, \cdots, n$ in (7) are the minimum solution of the righthand side of (5) with $\lambda = \lambda(j)$. They have to be determined before (7) is carried out. To do so, parallel computation can be used because (5) can be decomposed into n independent minimization subproblems by suitably rearranging the terms of dx^k and λ in (5). Consequently, each independent subproblem i is shown in (8).

$$\min_{dx_i^k, \eta_i} \{\eta_i^T r_i^{-1} \eta_i + \gamma (dx_i^k)^T dx_i^k + \lambda_i^T(j)[z_i - h_i^k - H_{ii}^k dx_i^k - \eta_i]$$

$$- \sum_{l \in J_i} \lambda_l^T(j) [H_{li}^k dx_i^k]\}$$

$$\text{for each } i = 1, 2, \cdots, n \tag{8}$$

An even better result is that (8) can be solved analytically as in (9) because (8) is quadratic and has positive definite Hessian matrix.

$$\hat{dx}_i^k = \frac{1}{2\gamma} [H_{ii}^k \lambda_i(j) + \sum_{l \in J_i} H_{li}^k \lambda_l(j)]$$

$$\hat{\eta}_i = \frac{1}{2} r_i \lambda_i(j)$$

$$\text{for each } i = 1, 2, \cdots, n \tag{9}$$

Thus, we may summarize the dual method as follows:

Starting from an initial λ, calculate (9) for each i in parallel to get $\hat{dx}_i^k$ and $\hat{\eta}_i$. Then calculate (7) for each i in parallel to get $\frac{\partial}{\partial\lambda_i}\Phi(\lambda)$. Finally update λ by (6). The above procedures will repeat until the convergence criteria $\max_i\{|\frac{\partial}{\partial\lambda_i}\Phi(\lambda(j))|_\infty\} < \epsilon$ occurs, where the notation $|(\cdot)|_\infty$ denotes the maximum of the absolute value of the components in the vector $(\cdot)$.

A flow chart for the overall computation procedures of the RQPD method is shown in Figure 1.

Remark 1: the states of the slack bus will remain fixed at its reference values throughout the iterative process. This implies that the dx term corresponding to the slack bus is always kept zero.

Global Convergence Property

Global convergence of the RQPD method had been shown in [11]. It means that starting from any initial point, the method will always converge. The proof is quite simple. It basically follows Han's work [18] that if the solution of (2) at any iteration k exists, the method will converge to a Kuhn-Tucker point of (1). Solution of (2) at any iteration k of course exists regardless of the values of x^k because η behaves like slack variables, and the dual method is a globally convergent method for quadratic problems.

Parallel Computation/Complete Decomposition Property

Clearly, the computations associated with any individual bus in (7) and (9), which are the major computation steps in the RQPD method, can be carried out independently. Such parallel computation property is achieved because of the complete decomposition property of the dual method.

THE DISTRIBUTED STATE ESTIMATOR

Clearly, the parallel computation for (7) and (9) can be carried out by n distributed processors. One processor corresponds to one bus. Furthermore, if the stepsizes $\bar{\alpha}$ and $\bar{\beta}$ in the updating procedures (3) and (6) are set as experienced constants, these n processors can update the values of x_i, $\lambda_i, i = 1, \cdots, n$ directly and in parallel. Therefore, we may organize n distributed processors and build up a data communication network environment to carry out the distributed and parallel computation procedures of the RQPD method to solve the WLS problem.

Environment for Distributed State Estimation

A data communication network with the following structure and functions is needed for distributed state estimation.

1. Nodes in the data communication network correspond one-to-one and onto the buses of the power network topologically and physically.

2. For each transmission line (i,j) in the power network, regardless of its circuit breaker $C(i,j)$ being open or closed, there exists a bi-directional high speed communication link in parallel with transmission line (i,j) physically.

3. Each node i in the data communication network contains a processor with local memories of a suitable size. The processor is capable of receiving the measurement data z_i and the detected status (open or close) of the circuit breakers $C(i,l)$, $l \in \mathcal{J}_i$. $\mathcal{J}_i$ denotes the set of nodes directly connected to node i in the data communication network. Clearly, J_i in the real-time power network is a subset of $\mathcal{J}_i$. The processor can transmit and receive data from the adjacent node processors in the network. In addition, the parameters of the transmission lines $(i,l), l \in \mathcal{J}_i$ will be stored in node i.

4. The data communication network has computer protocol designed to indicate the types of data transmitted from individual node.

Remark 2: Example of a data communication network for a 6-bus power network is shown in Figure 2. The black squares denote the node processors; the dashed lines denote the bi-directional communication links; the black or white circles denote the circuit breakers being closed or open. Based on the status of the circuit breakers at each node, the real-time power network topology can be described by this data communication network.

Computing Procedures of Each Processor

Thus, all the node processors in the data communication network will work as a team to execute the RQPD method for distributed state estimation. However, the accompanied issues of

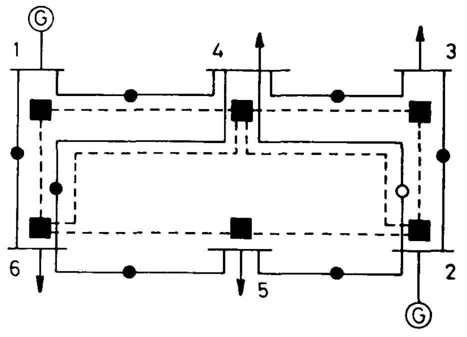


Figure 2: 6-bus power system & data communication network

synchronization and determination of the convergence have to be taken care of. The synchronization in distributed computation means that before the start of next algorithmic iteration, every node processor should have completed its computations in current iteration. In the environment of data communication network, the asynchronism induced by various communication delay in the data exchange and various computation time in node processors may destroy the synchronization of the algorithmic iteration. To cope with such problem of asynchronism, we employ a local synchronization scheme to maintain the synchronization of the RQPD method. Moreover, we have developed a global termination scheme to determine the convergence (or completion) of the distributed computation. Note that details of the above two schemes will be described in next subsection.

First of all, we divide the types of measurements, z_i, at each node i into $z_{ip}, z_{iq}, z_{iv}, z_{ilp}, z_{ilq}, l \in \mathcal{J}_i$. z_{ip} and z_{iq} denote the real-power and reactive-power injections at bus i respectively; z_{iv} denotes the voltage magnitude at bus i; z_{ilp} and z_{ilq} denote the real-power and reactive-power line flows in line (i,l) respectively. Furthermore, a null code will be used to replace the data of the non-existing measurements. Then, following is a list of the complete computing procedures of each node processor, say processor i, for its contribution to the team of the network of processors in executing the RQPD method.

Step 1: Guess the states x_i of bus i; set the weighting constant γ and the diagonal covariance submatrix r_i.

Step 2: For each $l \in \mathcal{J}_i$, if $C(i,l)$ is closed, send x_i to l; otherwise, replace the values of x_i by zero, then send to l.

Step 3: (Local synchronization) check if all $x_l, l \in \mathcal{J}_i$ arrived. If yes, go to Step 4; otherwise, repeat Step 3.

Step 4: Compute $h_i, H_{ii}, H_{il}, H_{li}, l \in \mathcal{J}_i$.

Step 5: Guess Lagrange multipliers λ_i : $\{\lambda_{ip}, \lambda_{iq}, \lambda_{iv}, \lambda_{ilp}, \lambda_{ilq}, l \in \mathcal{J}_i\}$. If any of the corresponding measurements is null, the associated Lagrange multiplier is set to be zero.

Step 6: For each $l \in \mathcal{J}_i$, if $C(i,l)$ is closed, send $\{\lambda_{ip}, \lambda_{iq}, \lambda_{ilp}, \lambda_{ilq}\}$ to node l; otherwise, replace the values of $\lambda_{ip}, \lambda_{iq}, \lambda_{ilp}$ and λ_{ilq} by zero, then send to l. (Note that λ_{iv} is not sent to any node $l \in \mathcal{J}_i$ because λ_{iv} corresponds to z_{iv} which has nothing to do with x_l.)

Step 7: (Local synchronization) check if all $\{\lambda_{lp}, \lambda_{lq}, \lambda_{lip}, \lambda_{liq}\}$, $l \in \mathcal{J}_i$, arrived. If yes, go to Step 8; otherwise, repeat Step 7.

Step 8: Compute $\hat{dx}_i$ and $\hat{\eta}_i$ according to (9), in which J_i is replaced by $\mathcal{J}_i$.

Step 9: For each $l \in \mathcal{J}_i$, if $C(i,l)$ is closed, send $\hat{dx}_i$ to node l; otherwise, replace the values of $\hat{dx}_i$ by zero then send to l.

Step 10: (Local synchronization) check if all $\hat{dx}_l$, $l \in \mathcal{J}_i$, arrived. If yes, go to Step 11; otherwise, repeat Step 10.

Step 11: Compute $\frac{\partial}{\partial\lambda_i}\Phi(\lambda)$ according to (7), in which J_i is replaced by $\mathcal{J}_i$.

Step 12: Update λ_i by $\lambda_i + \bar{\alpha}\frac{\partial}{\partial\lambda_i}\Phi(\lambda)$, where $\bar{\alpha}$ is a preselected accelerating constant.

Step 13: Return to Step 6 unless the following situations in the global termination scheme are met. (a) If both the current quadratic subproblem and the state estimation problem are complete, stop. (b) If the current quadratic subproblem is complete but the state estimation problem is not, update x_i by $x_i + \bar{\beta} \hat{dx}_i$ and return to Step 2, where $\bar{\beta}$ is a preselected accelerating constant. If (a) holds, the estimated states x_i of bus i are stored in node i.

Remark 3: From the steps associated with local synchronization scheme (Step 3, 7 and 10), it can be seen that the algorithm may hang up altogether if any communication link (i, l), which is responsible for the data exchange of nodes i and l, is down. To cope with such potential faults, we may assign in advance an alternative communication path to each pair of adjacent nodes besides the direct connected communication link. However, more sophisticated computer protocols are needed to detect the faulty communication link and get the alternative communication path to work. This is beyond the scope of this paper. Therefore, we will restrict ourselves in a fault-free communication network at present and leave the issues concerning communication faults for future research.

Remark 4: the iteration index is not needed here, and the dx term corresponding to the slack bus will be kept zero.

Remark 5: because the presence of Step 2, 6 and 9, and the replacement of J_i in (7) and (9) by $\mathcal{J}_i$, this distributed state estimator is working under the real-time power network model.

Remark 6: The transmitted data in between adjacent nodes may have errors due to the communication noise. Such transmission errors can be handled by error correcting codes [19], however, it is beyond the scope of this paper.

Local Synchronization and Global Termination

Local Synchronization Scheme: the idea of this scheme is if a processor knows which data to expect in the algorithmic iteration, it can start its predetermined computation once all the expected data are received [20]. This scheme especially suits the distributed computation of the RQPD method as we have described in Step 3, 7 and 10. By this local synchronization scheme, the accompanied communication delay in one data transmission and reception steps (e.g., Step 2 and 3) is $\max_{k\in L}\{T_k(C)\}$. $T_k(C)$ denote the communication delay of the data exchange in one communication link, and L denote the set of all communication links in the data communication network. Although $\max_{k\in L}\{T_k(C)\}$ is the maximum one of the communication delay among all data exchange, it is still only the communication delay of one data exchange. Such small communication delay is one of the advantage of the local synchronization scheme. In the following, we will use a simple example to illustrate the timing of the local synchronization scheme based distributed computation. Figure 3 shows a 5-processor data communication network, and each processor is denoted by a black square marked below by a processor number. The communication delay of the data exchange in between the adjacent nodes are marked on the communication links, where T denotes a unit time. The elapsed time of the computation performed in each node processor in one iteration are marked on top of the node processor. For the sake of explanation, the communication delay and computation time in this example are arbitrarily assigned. ***However, the qualitative conclusions about synchronization derived from this example is true for all combinations of computation time and communication delay.*** Now, to imitate the RQPD method based distributed computation, we make the following assumptions in the considered example. (1) One iteration means that every node processor should perform its computation once and just once. (2) In every iteration, each node processor only needs data from adjacent nodes and itself to perform its computation. (3) In every iteration, each node starts computing after all the expected data are received. (4) The computation process starts from time 0, at which each node transmits the initial computed data to adjacent nodes. Figure 4 shows four time frames of the procession of the local synchronization scheme based distributed computation of the considered example. The arrow marked by the processor number on top of the time scale indicates the instance that the processor receives all the expected data. Correspondingly, the instance that the processor finishes computation after receiving all the expected data is indicated by the arrow right below the time scale. From Figure 4, we see that the synchronization is already maintained

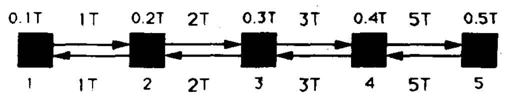


Figure 3: an example of data communication network

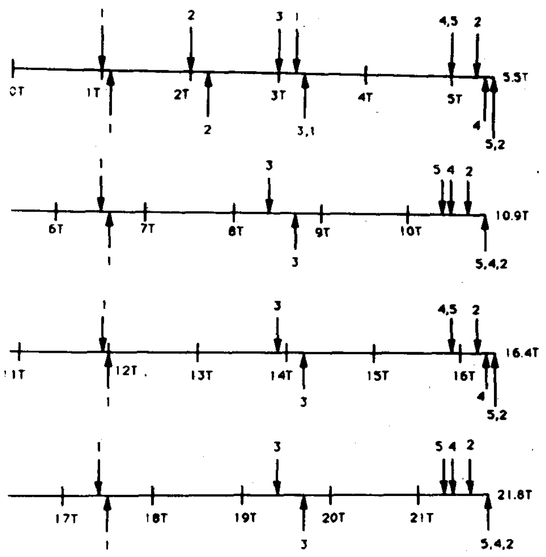


Figure 4: procession of the distributed computation

in and after the second time frame (i.e., the second iteration). Moreover, the communication delay in one synchronized iteration is 5T which is the maximum delay of a communication link in this example.

Global Termination Scheme: a two-phase global termination scheme is employed here. The first phase is used to determine the convergence (or completion) of the quadratic subproblem. It checks whether $\max_i\{|\frac{\partial}{\partial\lambda_i}\Phi(\lambda)|_\infty\} < \epsilon$. If yes, the second phase start to check whether $\max_i\{|\hat{dx}_i|_\infty\} < \epsilon$ to determine the completion of the distributed state estimation. In order to carry out this two-phase global termination scheme in the data communication network, we need to define a spanning tree of the data communication network and a special node designated as the root of the tree. A spanning tree of the network in Figure 2 is shown in Figure 5, where node 1 is the root node. A node is said to be a follower to node i if it is the head of an arrow starting from i, and we let $F(i)$ denote the set of the followers to node i. Clearly, $F(i) = \emptyset$ if i is a leaf node. A node is said to be a predecessor of node i if it is the tail of an arrow ending at i. For example, $F(4) = \{2, 3\}$ and node 4 is the predecessor of nodes 2 and 3. We define the status of a node i, S_i, as

$$S_i = \begin{cases} 1, & \text{if } |\frac{\partial}{\partial\lambda_i}\Phi(\lambda)|_\infty < \epsilon \text{ and } S_l = 1, \forall\ l \in F(i); \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

We also define the maximum observed deviation $|\hat{dy}_i|_\infty$ at node i as

$$|\hat{dy}_i|_\infty = \max[|\hat{dx}_i|_\infty, \max_l |\hat{dy}_l|_\infty, l \in F(i)] \quad (12)$$

Initially we assume that the status of all nodes are 0, which may change as computations proceed. The timing for any node processor i to transmit its S_i and $|\hat{dy}_i|_\infty$ to its predecessor is when $S_i = 1$ or S_i just changed from 1 to 0. Any node processor i will

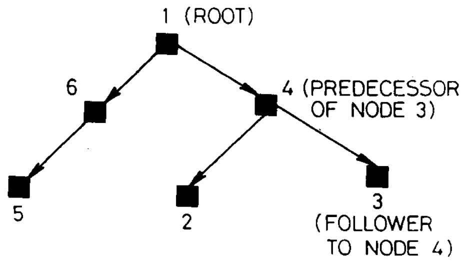


Figure 5: the spanning tree of the network in Figure 2

update S_i and $|\hat{dy}_i|_\infty$ according to (11) and (12) once S_l and $|\hat{dy}_l|_\infty$ from all $l \in F(i)$ are received. Let t denote the number of branches of the longest tree path from the root to a leaf in the spanning tree. If the status of the root S_r changes from 0 to 1 and remains 1 for consecutive t iterations of the dual method, then the current quadratic subproblem has completed t iterations before. When this occurs, the following decisions and the associated operations will be made at the root node.

1) If $|\hat{dy}_r|_\infty < \epsilon$, broadcasts a signal of completion of the state estimation problem to all nodes along the spanning tree.

2) If $|\hat{dy}_r|_\infty \geq \epsilon$, broadcasts a signal of updating x to all nodes, and each node i will return to Step 2 after it updates x_i.

Modification of the Global Termination Scheme: in order to determine the completion of the quadratic subproblem, updating the status S_i at each node i and broadcasting the result of the decision from the root node requires much communication overhead. Usually, the dual method improves fast for the first few tens of iterations but slows as it gets close to the solution. Therefore, we may set a maximum number of iterations, denoted by $\bar{It}$, that the dual method will iterate in solving the quadratic subproblem. Then, the first phase of the global termination scheme is modified by the directly updating x_i of each node i at the end of every $\bar{It}$ iterations of the dual method. Then the second phase of the global termination scheme will be modified as follows:

After every $\bar{It}$ iterations of the dual method, the following procedures will be performed. Starting from the leaf nodes, each node i will update $|\hat{dy}_i|_\infty$ according to (12) once it received all $|\hat{dy}_l|_\infty, l \in F(i)$. Thereafter, it will transmit $|\hat{dy}_i|_\infty$ to its predecessor. If the root node find $|\hat{dy}_r|_\infty < \epsilon$ after update, it will broadcast a signal of completion of estimation to all nodes.

Such modifications make the termination scheme simple, and also reduce the computation time.

ANALYSIS OF COMPUTATION COMPLEXITY

Following notations are used in this analysis:

IR: total number of the updates of x,

ID: total number of the updates of λ,

$|h_i|$: the number of measurements z_i,

$|\mathcal{J}_i|$: the number of adjacent node processors of node i,

t: the number of branches of the longest tree path in the spanning tree,

$T(\times)$: computation time for a multiplication,

$T(+)$: computation time for an addition or a subtraction,

$T(h_i)$: computation time for the evaluation of $h_i(x)$,

$T(H_{ij})$: computation time for the evaluation of $\partial h_i(x)/\partial x_j$,

$T(C)$: communication delay of one data word transmission,

$O(C)$: communication delay of global termination scheme.

Remark 7: We use $T(h_i)$ and $T(H_{ij})$ because h_i and H_{ij} contain trigonometric functions, the computation of which can not be decomposed into the number of multiplications and additions.

The computation time including communication delay of the distributed state estimator consists of two parts: (a) the recursive quadratic programming method in Steps 1–4 and Step 13 and (b) the dual method in Steps 5–12. Step 1 and 5 are initial guesses, thus no computation time is consumed.

Based on the defined notations, the computation time or the communication delay of the data exchange corresponding to Step 2–12 that consumed in a node, say node i, can be explicitly expressed in the following. Step 2 and 3 — $2T(C)$; Step4 — $T(h_i) + T(H_{ii}) + \sum_{l \in \mathcal{J}_i}[T(H_{il}) + T(H_{li})]$; Step 6 and 7 — $4T(C)$; Step8 — $[5|h_i| + 2\sum_{l \in \mathcal{J}_i}|h_l|]T(\times) + [2\sum_{l \in \mathcal{J}_i}|h_l|]T(+)$; Step 9 and 10 — 2T(C); Step 11 — $[2|h_i| + 2|\mathcal{J}_i||h_i|]T(\times) + [5|h_i| + 2|\mathcal{J}_i||h_i|]T(+)$; Step 12 — $|h_i|T(+)$. In the modified global termination scheme of Step 13, updating x_i will take $2T(\times) + 2T(+)$ computation time at node i, however, the communication delay $O(C) \leq 2tT(C) + t\max_i |\mathcal{J}_i|T(+)$. The upper bound of $O(C)$ is obtained based on the following facts (a) the computation time of "+" and "compare" are about the same, and (b) the number of followers to any node i is less than $\max_i |\mathcal{J}_i|$.

In each step of Step 2–12 and the procedure of updating x_i in Step 13, the computations or the data exchange for all processors are carried out in parallel. Therefore, based on the local synchronization scheme, the computation time (or the communication delay) of the whole system in each of the above step is only the maximum one of those consumed in all the individual nodes. For example, the computation time of the whole system in Step 4 is $\max_i\{T(h_i) + T(H_{ii}) + \sum_{l \in \mathcal{J}_i}[T(H_{il}) + T(H_{li})]\}$. Thus, we may express the total computation time of the distributed state estimator below:

$$IR \cdot [T(R) + 2T(C) + O(C)] + ID \cdot [T(D) + 6T(C)] \qquad (13)$$

where

$$T(R) = \max_i\{T(h_i) + T(H_{ii}) + \sum_{l \in \mathcal{J}_i}[T(H_{il}) + T(H_{li})]\}$$
$$+2[T(\times) + T(+)] \qquad (14)$$
$$T(D) = \max_i\{[5|h_i| + 2\sum_{l \in \mathcal{J}_i}|h_l|]T(\times) + [2\sum_{l \in \mathcal{J}_i}|h_l|]T(+)\}$$
$$+\max_i\{[2|h_i| + 2|\mathcal{J}_i||h_i|]T(\times) + [5|h_i| + 2|\mathcal{J}_i||h_i|]T(+)\}$$
$$+\max_i |h_i|T(+) \qquad (15)$$

It is worth noting that the distributed state estimator has the nature of sparse-matrix technique. This property can be observed from the computation procedures of each node processor, in which each individual processor only needs data from adjacent nodes and itself to carry out its computation. This indicates that only the operations involving nonzero terms are performed. Such nature of sparse-matrix technique is one of the causes that makes the computation complexity shown in (13)–(15) so simple.

Despite of the computation complexity obtained above, let us compare the convergence rate of the RQPD method based distributed state estimator with the popular Newton method [2] based centralized state estimator. Newton method has quadratic convergence rate, however, at the expense of solving a large set of linear equations in each iteration. In the RQPD method, the recursive quadratic programming has superlinear convergence rate, and each quadratic subproblem is solved by the dual method which has linear convergence rate accompanied with the extremely simple calculations for each iteration as shown in (6), (7) and (9). From the above comparison, we see that the distributed state estimator will consume a lot more iterations than the centralized state estimator. However, the inherent properties of parallel computation and sparse-matrix technique of the distributed state estimator will achieve tremendous computation time saving as described by the simple computation complexity in (13)–(15). A strong support for this claim is manifested by simulations demonstrated in the section next to the following.

THE SATELLITE FUNCTIONS

Along with the developed distributed state estimator, we can obtain several attractive satellite functions. These functions are described or developed in the following.

(1) Reduction of the Time-Skew Problem

The time-skew problem is greatly reduced because of the decentralized data acquisition.

(2) Topological Error Free Power Network Model

It is true as we have explained in Remark 5 in previous section.

(3) Easy Identification of the Unobservable States

The states x_i of bus i can only be observed from its measurements z_i or from the possible related measurements $z_l : \{z_{lp}, z_{lq}, z_{lip}, z_{liq}\}, l \in \mathcal{J}_i$. z_{lv} is excluded because it only has to do with v_l, the voltage magnitude of bus l. Based on this, the following simple procedures will be executed at each node i to identify the unobservable states.

Step id1: For each node $l \in \mathcal{J}_i$, send $\{z_{ip}, z_{iq}, z_{ilp}, z_{ilq}\}$ to node l.

Step id2: If z_i and z_l, for each $l \in \mathcal{J}_i$, are all null, the states of this bus are unobservable.

Advantage of the above identification before estimating states is that we may add pseudo measurements for the unobservable states if necessary.

(4) Bad Data Detection and Identification

Although the decentralized data acquisition will have less communication noise in telemetry, the consideration of the bad data detection and identification is still necessary.

Bad Data Detection Scheme– We employ the overall quadratic cost function $J(\hat{x})$ test [7] as our bad data detection scheme, where $J(\hat{x}) = [z - h(\hat{x})]^T R^{-1}[z - h(\hat{x})]$ and $\hat{x}$ denotes the solution of the estimated states obtained by the distributed state estimator.

By defining the node quadratic cost function $J(\hat{x}_i) = [z_i - h(\hat{x}_i)]^T r_i^{-1}[z_i - h(\hat{x}_i)]$, we have $J(\hat{x}) = \sum_{i=1}^n J(\hat{x}_i)$. Based on the spanning tree (Figure 5), we define $\tilde{J}(\hat{x}_i) = J(\hat{x}_i) + \sum_{l \in F(i)} \tilde{J}(\hat{x}_l)$. According to the above definition, we obtain $\tilde{J}(\hat{x}_r) = J(\hat{x})$. Then the bad data detection scheme can be carried out along with the modified global termination scheme of the distributed state estimator as described below.

After every $\bar{It}$ iterations of the dual method, the following procedures will be performed. Starting from the leaf nodes, any node i will update its value of $\tilde{J}(\hat{x}_i)$ once it receives all the values of $\tilde{J}(\hat{x}_l), l \in F(i)$. Moreover, each node i will send the value of $\tilde{J}(\hat{x}_i)$ to its predecessor once $\tilde{J}(\hat{x}_i)$ is updated. Suppose the global termination criteria of the distributed state estimator, $|\hat{dy}_r|_\infty < \epsilon$, is satisfied, the root node will compare $\tilde{J}(\hat{x}_r)$ with a preselected threshold value ξ_Q to test the hypothesis of the existence of the bad data. If $\tilde{J}(\hat{x}_r) > \xi_Q$, the root node will broadcast a command of starting bad data identification to all nodes along the spanning tree; otherwise, the estimation is complete.

Bad Data Identification Scheme— The weighted residual test will be employed here as our bad data identification scheme.

When receiving the command of starting bad data identification scheme, each node processor will compute the weighted residual $r_j^W = [z_j - h_j(\hat{x})]/\sigma_{jj}$ of the associated measurement z_j, where σ_{jj} is the square root of the j-th diagonal element of R^{-1}. Then each node processor will compare the magnitude of the computed weighted residual with a preselected threshold value ξ_W. If $|r_j^W| > \xi_W$, this measurement will be treated as a suspected bad data. Consequently, each node processor will send the index and the magnitude of r_j^W of the suspected bad measurement to the root node along the reverse direction of the spanning tree. The root node will then determine the largest one among all the received $|r_j^W|$'s, and broadcast the corresponding index to all nodes. Meanwhile a re-estimation will start after the elimination of the bad data.

It is commonly known that the weighted residual test is less effective because the largest $|r_j^W|$ is not necessarily corresponding to the bad measurement. However, a more sophisticated two-step residual test, which is as effective as but consuming less computation time than the normalized residual test, is currently under development [12].

SIMULATIONS

In this section, we will demonstrate the computational performance of the distributed state estimator and its numerical accuracy by simulations.

Numerical Accuracy

We pick up three of our simulated cases on the IEEE 30-bus system for discussions. In these cases, we let the measurements be the real and reactive power line flows; the initial guess are from the flat start; the parameters and constants are set below: $\epsilon = 0.001$, $\bar{It} = 30$, $\bar{\alpha} = 0.01$, $\bar{\beta} = 1.50$, $R = I$ (the identity matrix), and $\gamma = 25$. Then in the first case, we input the line flows from a load flow solution of the 30-bus system as our measurements. The convergent result deviates from the load flow solution by $\pm 0.0007 p.u.$ in voltage magnitude and $\pm 0.004 rad$ in phase angle on the average. Such deviations are satisfactory and they will be smaller if ϵ is chosen smaller. In the second case, we arbitrarily set the states of a bus unobservable by discarding the data of line flows connected to bus 30 from previous set of measurements. The convergent result, excluding bus 30, of our distributed state estimator deviates from the load flow solution by $\pm 0.0008 p.u.$ in voltage magnitude and $\pm 0.006 rad$ in phase angle on the average. It is slightly less accurate than case 1 because less measurement information is supplied in this case. In the third case, we assume that the sign of the real power flows in two lines (lines 2-6 and 10-22) are mistakenly reversed. The convergent result deviates from that obtained by Newton method by $\pm 0.0003 p.u.$ in voltage magnitude and $\pm 0.002 rad$ in phase angle. From the results of the above three cases, the numerical accuracy of the distributed state estimator are quite satisfactory.

Computational Performance

The iterations of our distributed state estimator for the above three cases are $(IR = 27, ID = 810)$, $(IR = 24, ID = 720)$ and $(IR = 25, ID = 750)$ respectively. The values $ID = IR \times \bar{It}$ are large as we expect. Following is a comparison of the computational performance between the distributed state estimator and the Newton method based state estimator. We assume that the processing time of $T(\times), T(+), T(h_i)$ and $T(H_{ij})$ are the same in both methods, and compare their computation time first. The total computation time excluding the communication delay of the distributed state estimator includes two terms $IR \cdot T(R)$ and $ID \cdot T(D)$ according to (13). On the other hand, the computation procedures in each iteration, say iteration k, of the Newton's method is to set up the coefficient matrices $H(x^k)(= [\partial h(x)/\partial x]|_{x=x^k})$, $C(x^k)(= H(x^k)^T R^{-1} H(x^k))$ and the vector $z - h(x^k)$ first. Then it will solve the set of linear equations $C(x^k)\Delta x^k = z - h(x^k)$ to obtain Δx^k and update $x^{k+1} = x^k + \Delta x^k$. According to (14)-(15) and the simulation results, we find that (a) the computation time $IR \cdot T(R)$ is approximate to the elapsed time required to calculate the nonzero elements of $H(x^0)$ and $z - h(x^0)$, where x^0 is the initial guess of the states x, and (b) $ID \cdot T(D)$ is about one tenth of (or comparable to) the computation time consumed in calculating the full matrix $C(x^0)$ (or the nonzero elements of $C(x^0)$). Therefore, the total computation time excluding communication delay in the proposed state estimator is just enough for the Newton's method to get ready to solve the set of linear equations in its first iteration. Furthermore, the total communication delay of the distributed state estimator is $IR \cdot [2T(C) + O(C)] + 6 \cdot ID \cdot T(C)$ according to (13). Based on current optical fiber technologies, the data transmission speed ranges from 565 Mbits/sec [21] to 20 Gbit/sec [22]. For a data word of 32 bits, the total communication delay of our simulated cases is less than 0.3 ms, which is very small compared to the computation time. Therefore, our distributed state estimator definitely outplay the Newton method in the aspect of computational performance.

CONCLUSION

We have developed a globally convergent distributed state estimator which has several attractive satellite functions. Though we can not deny that when the system size grows, the iteration number of the distributed state estimator will be increasing at least proportionally. However, the required computation time is still small because of its inherent property of parallel computation and the nature of sparse-matrix technique. As we have demonstrated by simulations that the developed distributed state estimator has excellent computational performance.

It seems that the distributed state estimator is currently impractical because the control technology of the power system is not yet reaching the status of decentralization, and the integration of the data communication and power network is not yet matured. However, it is promising as pointed out by Gaushell and Darlington that we will be seeing the distributed processing throughout the utility's facilities in this decade [13]. Considering the possibility of direct angle measurement in the future, the state estimation is still needed because bad data may be present. In that case, it is quite possible to develop a more simplified distributed state estimator than the one proposed.

Finally, we would like to point out that the distributed computation technique developed in this paper can be extended to solve the constrained nonlinear optimization problems of large-scale interconnected systems. Many power system management problems lie in this category. Therefore, we believe that our technique will not be limited to state estimation.

ACKNOWLEDGMENT

This work was supported by National Science Council under Grant NSC79-0404-E009-48.

REFERENCES

[1] F. C. Schweppe and J. Wildes,"Power system static-state estimation, part I: exact model," *IEEE Trans. Power App. Syst.*, vol. PAS-89, no. 1, pp. 120-125, Jan. 1970.

[2] F. C. Schweppe,"Power system static-state estimation, part III: implementation," *IEEE Trans. Power App. Syst.*, vol. PAS-89, no. 1, pp. 130-135, Jan. 1970.

[3] A. Bose and K.A. Clements," Real-time modeling of power networks,"*Proc. IEEE,* vol. 75, no. 12, pp. 1607-1622, Dec. 1987.

[4] H. Kobayashi, S. Narita and M. S. A. A. Hamman,"Model coordination method applied to power system control and estimation problems," *Proc. of the IFAC/IFIP 4th Int. Conf. on Digital Computer Appl. to Process Control,* pp. 114-128, 1974.

[5] H. Mukai," Parallel multiarea state estimation," *Electric Power Research Institute*, report of research project 1764-1, Jan. 1982.

[6] Th. Van Cutsem and M. Ribbens-Pavella,"Critical survey of hierarchical methods for state estimation of electric power systems,"*IEEE Trans. Power App. Syst.*, vol. PAS-102, no. 10, Oct. 1983.

[7] L. Mili, Th. Van Cutsem and M. Ribbens-Pavella,"Bad data identification methods in power system state estimation — a comparative study,"*IEEE Trans. Power App. Syst.* vol. PAS-104, no. 11,pp. 3037-3049 Nov. 1985.

[8] M. Ikeda, D. D. Siljak and K. Yasuda,"Optimality of decentralized control for large-scale systems,"*Automatica*, 19, pp. 309-317, 1983.

[9] Z. Shi and W. B. Gao, "Stabilization by decentralized dontrol for large-scale interconnected systems,"*Large-Scale Systems,* 10,pp. 147-155 ,1986.

[10] M. K. Sundareshan and R. M. Elbanna,"Large-scale systems with symmetrically interconnected subsystems: analysis and synthesis of decentralized controllers,"*Proc. 29th CDC,* Honolulu, Hawaii, pp. 1137-1142, Dec. 1990.

[11] S.-Y. Lin,"A method to solve power system static-state estimation problem by using a network of processors," *Proc. 29th CDC,* Hawaii, Honolulu, pp. 3067-3072, Dec. 5-7, 1990.

[12] S.-Y. Lin, " A distributed state estimator and bad data detection/identification of Power System, " *NSC Report under contract NSC79-0404-E009-48,* Dept. of Control Eng., National Chiao Tung Univ., TAIWAN, ROC, Feb. 1991.

[13] J. G. Dennis and T. D. Henry,"Supervisory control and data acquisition,"*Proc. IEEE,* vol. 75, no. 12, Dec. 1987.

[14] K. C. Holte,"Technology requirements for a competitive electric utility market in the 21st century," *IEEE Power Eng. Review,* pp. 18-22, July 1989.

[15] J. Zaborszky, K. W. Whang and K. V. Prasad,"Ultra fast state estimation for the large electric power system,"*IEEE Trans. Automat. Contr.*, vol. AC-25,no. 4 pp. 839-841, Aug. 1980.

[16] C. W. Brice and R. K. Cavin,"Multiprocessor static state estimation," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 2,pp. 302-308 Feb. 1982.

[17] D.G. Luenberger, *Linear and nonlinear programming,* second edition, MA: Addison-Wesley Publ. Co., 1984.

[18] S.P. Han," A globally convergent method for nonlinear programming," *Jounal of Optimization Theory and Applications,* vol. 22, no. 3, pp. 297-309, July 1977.

[19] A. S. Tanenbaum, *Computer networks,* N. J. Englewood Cliffs: Prentice Hall Inc. , 1981.

[20] D.P. Bertsekas and J.N. Tsitstklis, *Parallel and distributed computation,* United Kingdom, London: Prentice Hall International Limited, 1989.

[21] M. Sotom et. al.,"Multichannel FSK transmission experiment at 565 Mbits/s using tunable three-electrode DFB lasers," *Electronics Letters,* vol. 26, no. 13, pp. 869-870, 21st June 1990.

[22] M. Nakazawa et. al.,"20 Gbit/s solution transmission over 200 km using Erbium-doped fibre repeaters," *Electronics Letters,* vol. 26, no. 19, pp. 1592-1593, 13th Sep. 1990.

Shin-Yeu Lin was born in Taiwan, ROC, on June 26, 1953. He received the B.S. degree in electronic engineering from National Chiao Tung University, the M.S. degree in electrical engineering from University of Texas at El Paso and the D.Sc. degree in systems science and mathematics from Washington University in St. Louis, Missouri, in 1975, 1979 and 1983 respectively.

From 1984 to 1985, he was with Washington University working as a Research Associate then a Visiting Assistant Professor. From 1985 to 1986, he joined GTE Laboratories working in the Switching Department as a Senior MTS. Since 1987 till now, he is an Associate Professor of Control Engineering of National Chiao Tung University. His major research interests are Large-Scale Power Systems, Optimization Theory and Applications, Distributed and Parallel Computations.