



PARALLEL STATE ESTIMATION WITH CONFIDENCE LIMIT ANALYSIS

JOANNA K. HARTLEY* and ANDRZEJ BARGIELA*

*Real Time Telemetry Systems, Department of Computing, The Nottingham
Trent University, Burton Street, Nottingham NG1 4BU, U.K.*

(Received 22 January 1996; In final form 1 November 1996)

This paper describes the parallelisation of a state estimator with confidence limit analysis. State estimation involves the optimal fitting of an overdetermined set of measurements to the corresponding values calculated from the mathematical model of the system. The inaccuracies associated with measurements lead to discrepancies within the state estimate. Consequently for the state estimation algorithm to be of practical use it needs to quantify the effect of these discrepancies in the form of state confidence limits [2]. However, the quasi-quadratic numerical complexity of the state estimation algorithms suggests a need for parallel implementation of the probabilistic state estimation, so that the real-time performance may be maintained also for large-scale systems. The algorithm is based on the idea of 'tearing' the original system into subsystems and then coordination of the resulting subsystem solutions. The algorithm has been tested in the context of water distribution systems state estimation.

Keywords: Distributed systems; state estimation; confidence limit analysis; transputers; parallel algorithms

Classification Categories: C 1.2, F 1.2, g 1.6, g 2.2

1. INTRODUCTION

The basic role of state estimation is the generation of a coherent and reliable state vector based on the measurements and the mathematical model of the system. State estimation involves minimising the difference between the actual measurements and the corresponding values that are calculated using the current estimate of the state vector. The way in which this optimisation problem is posed, regarding the definition of the cost function, characterises

* Tel: +44 (0) 115 9418418 ext. 6172; Fax: +44 (0) 115 9486518; Email jkh@doc.ntu.ac.uk

different state estimators [4, 7, 8, 9, 12]. In general, state estimation procedures are approximately quadratically dependent on the size of the physical network. Consequently, for a large scale system, it is computationally advantageous to split the system and to perform state estimation of the resulting subsystems.

Recent advances in computing hardware also point towards the development of distributed, parallel programs as a cost-effective scheme. Multiprocessor systems can be classified into two categories according to the manner in which these processing elements communicate. The first category, called tightly coupled systems, consists of processors that use a common shared memory. They are used for problems that have many synchronisation points and thus require very fast access to shared data. The second category, more appropriate for optimisation problems which include coarse-grained tasks with few synchronisation points, consists of distributed processors which have their own local memories and which communicate by means of message passing. The advantage of such systems is that they can be flexibly modified to suit the problem structure since there are no conflicts in memory access. However such flexibility is achieved at the expense of limited efficiency of access to shared data.

In the case of state estimation of water distribution systems, the physical structure of the problem can be utilised when decomposing it into a number of smaller subproblems making it particularly suitable for use with distributed processors. Each of these subproblems are solved individually and the solutions are recombined to achieve the solution of the overall problem. To take into account the interactions between the subproblems, solution usually proceeds iteratively with information exchange between subproblems and a coordinating master problem. The manner in which coordination is achieved characterises the different decomposition methods [8, 10, 13].

2. STATE ESTIMATION

State estimation of a water distribution system involves solving an overdetermined set of equations describing mass-balances in network nodes and pressure-flow relationships:

$$\mathbf{g}(\mathbf{x}) = \mathbf{z} + \boldsymbol{\omega} \quad (1)$$

where \mathbf{x} is a vector of N state variables, called the state vector; \mathbf{z} is the measurement vector which consists of M real measurement values and pseudomeasurements (estimated values); \mathbf{g} is a nonlinear function relating the state

variables to the measurements (real and pseudo); and ω is a vector of measurement errors.

The network equation (1) can be described in greater detail in the context of water distribution systems:

When z_i represents a pressure measurement in node j , the i th component of the network function is as follows:

$$g_i(\mathbf{x}) = x_j = z_i + \omega_i \quad (2)$$

Similarly, when z_i represents an inflow measurement in node j , the i th component of the network function is as follows:

$$g_i(\mathbf{x}) = x_j = z_i + \omega_i \quad (3)$$

When z_i represents a flow measurement between nodes j and k , the i th component of the network function is as follows:

$$g_i(\mathbf{x}) = Q_{jk}(\mathbf{x}) = z_i + \omega_i \quad (4)$$

where Q_{jk} denotes the hydraulic pressure-flow relationship for the link connecting node j and node k .

Finally, when z_i represents a load (consumption) in node k , the i th component of the network function is as follows:

$$g_i(\mathbf{x}) = \sum_{j \in \Omega_k} Q_{jk}(\mathbf{x}) + x_l = z_i + \omega_i \quad (5)$$

where Ω_k is the set of nodes neighbouring node k , and x_l is the inflow variable for node k (if it exists). These equations are the mass-balance equations.

The state estimation procedure involves minimising the measurement errors ω by assigning appropriate values to the elements of the state estimate \mathbf{x} . The least absolute values method (6) is considered to be the most suitable minimisation method in the case of the state estimation of water distribution systems.

$$\text{minimise } |\mathbf{g}(\mathbf{x}) - \mathbf{z}| \quad (6)$$

This is because the set of measurements being considered may include gross errors due to the unreliable retrieval of data — the least absolute values procedure will implicitly reject such bad data.

The system of nonlinear equations can be solved using the Newton-Raphson method which obtains the solution to (1), by iteratively solving a system of linearised equations. The function $\mathbf{g}(\mathbf{x})$ is expanded by an initial estimate of

the state vector \mathbf{x}_0 using a Taylor series of first order, leading to the iterative equation

$$\mathbf{g}(\mathbf{x}_{(k+1)}) = \mathbf{g}(\mathbf{x}_{(k)}) + \left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_{(k)}} \cdot \Delta \mathbf{x}_{(k)} \quad (7)$$

where the subscript (k) implies that the attached variable belongs to iteration number k .

By denoting

$$\mathbf{g}(\mathbf{x}_{(k+1)}) - \mathbf{g}(\mathbf{x}_{(k)}) = \Delta \mathbf{z} \quad (8)$$

$$\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} = \Delta \mathbf{x} \quad (9)$$

$$\left. \frac{\partial \mathbf{g}}{\partial \mathbf{x}} \right|_{\mathbf{x}=\mathbf{x}_{(k)}} = \mathbf{J} \quad (10)$$

the equation to be solved in each iteration may be rewritten as

$$\Delta \mathbf{z} = \mathbf{J} \cdot \Delta \mathbf{x} \quad (11)$$

The state vector may then be corrected with the use of $\Delta \mathbf{x}$:

$$\mathbf{x}_{(k+1)} = \mathbf{x}_{(k)} + \Delta \mathbf{x} \quad (12)$$

3. STATE ESTIMATION WITH CONFIDENCE LIMIT ANALYSIS

Such a traditional, deterministic state estimator can however provide misleading information. The estimate for the state of a water distribution system is dependent on measurement data, but this data may be inaccurate. In particular, the values of the pseudomeasurements, which represent the estimates of consumption, can be very inaccurate (for example, between 50% and 100% in error). To counteract the effect of the pseudomeasurement errors, it is necessary to make accurate measurements of flows or pressures in the distribution networks, bearing in mind however that the deployment of additional meters is costly.

In order to enable the balancing of the concerns of estimation accuracy with metering costs, a state estimator has been developed which calculates the states in terms of their upper and lower bounds [3, 5]. A sensitivity matrix, representing the relationship between the change to the measurement value and the change to the state vector, is evaluated and used in calculating the maximum possible disturbance to each of the state variables in turn. The amalgamation of all "worst case" scenarios, devised for these state variables, represents confidence limits for the estimates.

4. THE DISTRIBUTED PROBABILISTIC STATE ESTIMATION ALGORITHM

In any procedure of state estimation, the difficulty of satisfying the conflicting requirements on speed and accuracy increases with the system's size [9, 12]. The general consensus is that in order to overcome such problems the system must be split into subsystems and the subsystem state estimations should be executed concurrently. Although the parallel state estimator is more complex and converges less rapidly than the "centralised" one, the advantage of using distributed computing resources for its execution gives a potential for modular augmentation of computing power which far outweighs the disadvantages.

Kron [11] developed a decomposition-coordination technique in the context of linear systems. A generalisation of this technique for nonlinear systems has been proposed in [1, 8]. It has been shown that the only requirement is that corrections to the subsolutions must be restricted in magnitude so that the local mathematical models of the subsystems are not invalidated when iterating towards the solution.

The overdetermined set of M equations (1) are firstly partitioned into s subsets representing the subnetworks. The effect caused by the removal of the interconnecting branches is compensated for by adding appropriate flows to the corresponding boundary nodes (Figure 1). Having augmented these compensating flows, the measurement vector is updated to take these alterations into consideration. The resulting vector is:

$$z^P = z + f^P \quad (13)$$

where

$$f^P = C_{\alpha\beta} f_{\beta}(x) \quad (14)$$

with f_{β} representing the flow through the cut-pipe β and

$C_{\alpha\beta}$ denoting the extended node/cut-pipe incidence matrix defined as

$$C_{\alpha\beta} = \begin{cases} -1, & \text{if the node } \alpha \text{ is a sending node of the cut-pipe } \beta \\ -1, & \text{if the node } \alpha \text{ is a receiving node of the cut-pipe } \beta \\ 0, & \text{if the node } \alpha \text{ is not incident to cut-pipe } \beta \\ 0, & \text{if the node } \alpha \text{ does not exist (ie. } n < \alpha \leq m) \end{cases} \quad (15)$$

$$C_{\alpha\beta} = (C_{\beta\alpha})^T.$$

The removal of cutlines decouples the subnetworks. So, the Jacobian matrix for the partitioned network, J^P , assumes block diagonal form and becomes amenable to the distributed computation of the partitioned state vector x^P in the subnetworks. With reference to equation (11), the desired relation between

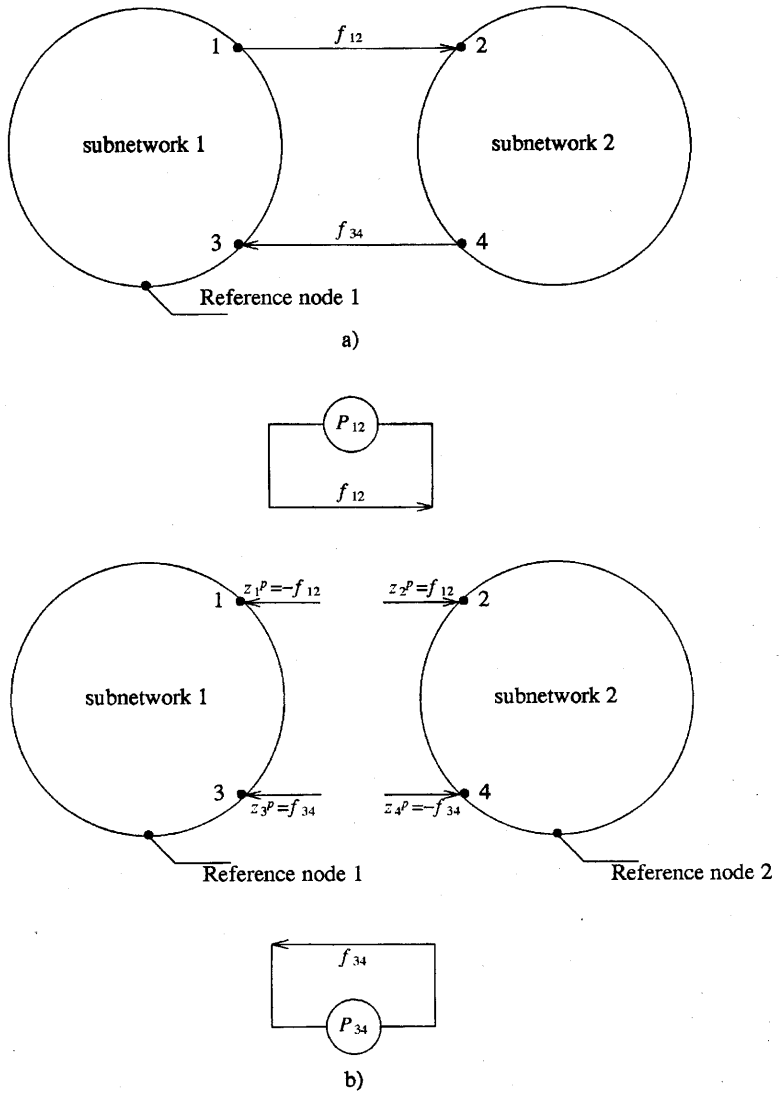


FIGURE 1 Partitioning the network. a) Original system; b) Equivalent partitioned system.

these newly defined variables can be expressed as:

$$z^p = g(x_0) + J^p \Delta x^p \tag{16}$$

Due to the non-singularity of the Jacobian matrix, equation (16) can be rearranged in the form:

$$\Delta x^p = [(J^p)^T W J^p]^{-1} (J^p)^T W (z^p - g(x_0)) \tag{17}$$

The change to the substate estimates, $\Delta \mathbf{x}^p$, can be calculated by substituting for \mathbf{z}^p (see equations (13) and (14)) and linearising the cut-line flows around the initial state estimate \mathbf{x}_0 :

$$\Delta \mathbf{x}^p = [(J^p)^T W J^p]^{-1} (J^p)^T W (\mathbf{z} + C_{\alpha\beta} \mathbf{f}_\beta(\mathbf{x}_0) - \mathbf{g}(\mathbf{x}_0)) + [(J^p)^T W J^p]^{-1} \times (J^p)^T W \cdot C_{\alpha\beta} \Delta \mathbf{f}_\beta \quad (18)$$

Equation (18) can be simplified with the utilisation of $\mathbf{z}_0^p = \mathbf{z} + C_{\alpha\beta} \mathbf{f}_\beta(\mathbf{x}_0)$. This vector \mathbf{z}_0^p represents the measurement vector updated with respect to the initial state estimate, to compensate for the cutting of the interconnecting branches.

$$\Delta \mathbf{x}^p = [(J^p)^T W J^p]^{-1} (J^p)^T W (\mathbf{z}_0^p - \mathbf{g}(\mathbf{x}_0)) + [(J^p)^T W J^p]^{-1} (J^p)^T W \cdot C_{\alpha\beta} \Delta \mathbf{f}_\beta \quad (19)$$

Due to the alterations to the substate estimates, $\Delta \mathbf{x}^p$, being dependent on the alterations made to the cut-line flows $\Delta \mathbf{f}_\beta$, it is necessary to augment a number of equations relating the subnetwork pressures and the cut-line flows. These equations are found using the pressure balance equations for the removed pipes:

$$\mathbf{h}(\mathbf{f}_\beta) + \mathbf{P}_\beta = \mathbf{0} \quad (20)$$

$\mathbf{h}(\mathbf{f}_\beta)$ relates the flow and the pressure drop across the pipe β .

$\mathbf{P}_\beta = C_{\beta\alpha} \mathbf{x}^p + M_{\beta\gamma} \mathbf{x}_\gamma$ denotes the pressure drop in terms of the pressure differences in the end-nodes of the pipe β and the difference of reference pressures \mathbf{x}_γ in the relevant subnetwork γ .

The cut-pipe/subnetwork incidence matrix, $M_{\beta\gamma}$, is described as

$$M_{\beta\gamma} = \begin{cases} -1, & \text{if the flow in cut-pipe } \beta \text{ is directed into subnetwork } \gamma \\ -1, & \text{if the flow in cut-pipe } \beta \text{ is directed away from} \\ & \text{subnetwork } \gamma \\ 0, & \text{if the cut-pipe } \beta \text{ is not incident to subnetwork } \gamma \end{cases} \quad (21)$$

$$M_{\beta\gamma} = (M_{\gamma\beta})^T.$$

Linearising equation (20) about \mathbf{x}_0 , the pressure balance equations for the removed networks become:

$$\mathbf{h}(\mathbf{f}_\beta(\mathbf{x}_0)) + \frac{\partial \mathbf{h}}{\partial \mathbf{f}_\beta} \Delta \mathbf{f}_\beta + C_{\beta\alpha} \mathbf{x}_0 + C_{\beta\alpha} \Delta \mathbf{x}^p + M_{\beta\gamma} \mathbf{x}_\gamma + M_{\beta\gamma} \Delta \mathbf{x}_\gamma = \mathbf{0} \quad (22)$$

The pressure balance equations (20) hold for \mathbf{x}_0 . So equation (18) can be written as:

$$\mathbf{H} \cdot \Delta \mathbf{f}_\beta + C_{\beta\alpha} \Delta \mathbf{x}^p + M_{\beta\gamma} \Delta \mathbf{x}_\gamma = \mathbf{0} \quad (23)$$

where $\mathbf{H} = (\partial \mathbf{h} / \partial \mathbf{f}_\beta)|_{\mathbf{f}_\beta(\mathbf{x}_0)}$

This equation illustrates that the alterations made to the subnetworks' reference pressures, Δx_γ , are related to the flow difference, Δf_β , in cut-pipes. So the amount of transfer between subnetworks is duly affected. This means that the mass balance equations for the subnetworks must be considered:

$$z_\gamma + M_{\gamma\beta} \cdot f_\beta = 0 \quad (24)$$

$z_\gamma = \sum_{i \in I_\gamma} z_i$, and I_γ is the set of nodes in the subnetwork γ .

The subnetwork equations (24) can be linearised about x_0 , resulting in

$$z_\gamma^p + M_{\gamma\beta} \cdot \Delta f_\beta = 0 \quad (25)$$

where

$$z_\gamma^p = z_\gamma + M_{\gamma\beta} \cdot f_\beta(x_0) \quad (26)$$

The equations (18), (23) and (25) constitute the model of the nonlinear network system to be solved in a distributed mode. The equations can be solved simultaneously by means of an augmented matrix

$$\begin{bmatrix} z_0^p - g(x_0) \\ 0 \\ z_\gamma^p \end{bmatrix} = \begin{bmatrix} J^p & -C_{\alpha\beta} & 0 \\ C_{\beta\alpha} & H & M_{\beta\gamma} \\ 0 & -M_{\gamma\beta} & 0 \end{bmatrix} \begin{bmatrix} \Delta x^p \\ \Delta f^\beta \\ \Delta x_\gamma \end{bmatrix} \quad (27)$$

The uncoordinated subsystems solutions $\Delta \tilde{x}^p$ are obtained first by solving

$$J^p \cdot \Delta \tilde{x}^p = z_0^p - g(x_0) \quad (28)$$

to give

$$\Delta \tilde{x}^p = [(J^p)^T W J^p]^{-1} (J^p)^T W (z_0^p - g(x_0)) \quad (29)$$

These resultant uncoordinated subsystem solutions may then be coordinated by introducing corrections associated with the subnetwork reference pressures and the inter-subnetwork flows. With the pre-calculation of equation (29), the augmented matrix equation (27) may therefore be reformulated as

$$\begin{bmatrix} -C_{\beta\alpha} \Delta \tilde{x}^p \\ z_\gamma^p \end{bmatrix} = \begin{bmatrix} H + C_{\beta\alpha} [(J^p)^T W J^p]^{-1} & (J^p)^T W C_{\alpha\beta} M_{\beta\gamma} \\ -M_{\gamma\beta} & 0 \end{bmatrix} \begin{bmatrix} \Delta f^\beta \\ \Delta x_\gamma \end{bmatrix} \quad (30)$$

This set of equations includes the alteration required of the cut-pipe flows, Δf^β . So, combining this solution to those generated by equations (20) and (29), the iterative update to the calculated pressure is derived

$$\Delta x^p = \Delta \tilde{x}^p + [(J^p)^T W J^p]^{-1} (J^p)^T W C_{\alpha\beta} \Delta f_\beta \quad (31)$$

and augmentation of alterations to any generated subnetwork reference pressures gives

$$\Delta \mathbf{x} = \Delta \mathbf{x}^p + K_{\alpha\gamma} \Delta \mathbf{x}_\gamma \quad (32)$$

where

$$K_{\alpha\gamma} = \begin{cases} 1, & \text{if node } \alpha \text{ holds a generated reference pressure of subnetwork } \gamma \\ 0, & \text{otherwise} \end{cases} \quad (33)$$

resulting in the equation (9):

$$\mathbf{x}_{(k+1)} - \mathbf{x}_{(k)} = \Delta \mathbf{x}$$

Once the point state estimate has been derived, the algorithm then determines the associated bounds [3,5]. The upper and lower bounds \mathbf{x}^u and \mathbf{x}^l are found with utilisation of the sensitivity matrix $(J^T J)^{-1} J^T$. This $n \times n$ matrix is the pseudo-inverse of the Jacobian matrix, J . It is so called because the (i, j) th element J_{ij} of the matrix relates the sensitivity of the i th element of the state vector x_i to the j th element of the measurement vector z_j . The two vectors of upper and lower bounds are calculated using the equations:

$$\mathbf{x}^u = \mathbf{x} + |(J^T J)^{-1} (J^T) \cdot \Delta \mathbf{z}| \quad (34)$$

$$\mathbf{x}^l = \mathbf{x} - |(J^T J)^{-1} (J^T) \cdot \Delta \mathbf{z}| \quad (35)$$

where $\Delta \mathbf{z}$ is the measurement error vector of the pseudo- and telemetered measurements.

With calculation of the upper and lower bounds, the approximate linearised state uncertainty set, containing the elements \mathbf{x} has been determined. The set is described as approximate because some of the vectors $\Delta \mathbf{z}$ may be inconsistent, and therefore redundant. The intervals, \mathbf{x}^l and \mathbf{x}^u , derived from this approximate linearised state uncertainty set will thus not be the smallest possible. However, this approximate set contains the whole of the true linearised state uncertainty set, so there is the necessary assurance that these intervals will never rule out any feasible state vector from the true state uncertainty box [5].

This distributed probabilistic state estimation algorithm may be summarised as follows:

1. Read-in the system data consisting partially of an overdetermined set of measurements (13).
2. Form subsystem data packets by removal of a number of links. Update the data packets with compensating flows. Send the packets to individual estimators, distributed across a network of transputers (16).

3. Calculate each of the uncoordinated subsystem solutions by least absolute values regression to the linearised set of equations (29).
4. Collate and coordinate the partial solutions, taking into account the effect of necessary alterations to each of the subsystem's reference pressure and the effect of inter-subnetwork flows.
5. If the coordinated corrections from Step 4 are less than a given threshold value (i.e. the least absolute values solution has been found) then go to Step 6, otherwise repeat from Step 2.
6. Calculate the upper and lower bounds of the derived point state estimate (34) and (35).

5. RESULTS

The algorithm has been tested on a 130-node test water distribution network [6]. Meters are selectively placed throughout the entire network to measure the pressure and flow values. However, due to cost, the number of telemeasurements is limited. So pseudomeasurements — estimates of nodal consumption — are also used to minimise the overall meter cost. These consumption values are, however, of a very fluctuating nature giving rise to discrepancies.

The distributed probabilistic state estimation algorithm firstly splits the network into two subnetworks, with nodes numbered 1–65 and 66–130 respectively. The algorithm then proceeds to determine the state estimate of the entire network by individual subnetwork solution on two processors followed by coordination on the root processor. Once this has been achieved the algorithm determines the individual bounds of the state estimate elements.

In Example 1, the algorithm is tested on a network which has two measurement points, reference head measurements are taken at nodes 65 and 130 (one head measurement in each subnetwork), the accuracy of the meters is 200 Nm^{-2} .

In Example 2 more meters are added to the network. The 130-node network has ten meters placed within it. Five of the meters are head meters (each with an accuracy of 200 Nm^{-2}), and the other five meters are flow meters (each with an accuracy of $0.005 \text{ m}^3 \text{ s}^{-1}$). Table I shows the increased accuracy of the state estimate due to the inclusion of extra head and flow meters, as would be expected.

P denotes a pressure measurement reading at the node, with an accuracy of 200 Nm^{-2}

F denotes a flow measurement reading in a pipe adjoining the node, with accuracy $0.005 \text{ m}^3 \text{ s}^{-1}$

TABLE I Pressure bounds at selected network nodes

<i>Node</i>	<i>Example 1</i>	<i>Example 2</i>	<i>Node</i>	<i>Example 1</i>	<i>Example 2</i>
6	57430	8160 F	53	55720	6900 F
7	39830	6300 F	57	19790	200 P
9	57510	8240 F	63	41030	3170 F
19	124570	14780 F	65	200 P	200 P
21	115830	9170 F	77	42980	200 P
24	50480	200 P	80	11080	660 F
31	75050	3840 F	82	11340	270 F
33	100380	200 P	84	24820	14970 F
34	49470	1210 F	86	38940	22340 F
38	49360	1320 F	89	63350	200 P
39	54970	200 PF	95	11390	200 P
45	55370	540 F	97	60900	13350 F
47	53490	1710 F	115	53560	200 P
51	54380	5470 F	130	200 P	200 P

The algorithm has also been tested for computational efficiency, on three different 130-node networks. The network was partitioned into two, three, and then four, subnetworks of approximately equal size. The subnetworks were each solved individually on separate processors. The convergence times for the networks are shown in Table II.

The total solution time is an amalgamation of partitioning the system data for the subsystems, the total subsystems solution time, the time taken to coordinate these subsystem solutions and the time taken to evaluate the related upper and lower bounds.

The subsystem solution involves an iterative procedure to evaluate the solution of the subsystem followed by manipulation of the subsystem data

TABLE II Convergence times of the algorithm when applied to four different 130-node networks, each partitioned into two, three and four subnetworks.

	<i>Number of nodes</i>	<i>Number of cutlines</i>	<i>Total subsystems solution time (s)</i>	<i>Number of subsystem iterations</i>	<i>Coordination time (s)</i>	<i>Confidence limit analysis time (s)</i>	<i>Total solution time (s)</i>
Network 1	65,65	4	10.7	1,1	0.03	32.7	44.0
	43, 43, 44	17	5.4	1, 1, 1	0.08	32.6	38.7
	32, 32, 33, 33	24	5.0	1, 1, 1, 1	0.19	33.3	39.0
Network 2	65, 65	3	8.7	1, 1	0.03	32.1	41.3
	43, 43, 44	17	5.4	1, 1, 1	0.08	32.0	38.0
	32, 32, 33, 33	23	3.6	1, 1, 2, 5	0.17	32.1	36.5
Network 3	65, 65	4	8.7	1, 2	0.03	31.2	40.5
	43, 43, 44	15	4.0	1, 1, 3	0.07	31.6	36.2
	32, 32, 33, 33	24	9.0	1, 1, 5, 22	0.23	31.8	41.5
Network 4	65, 65	3	5.7	2, 2	0.03	29.9	36.2
	43, 43, 44	16	4.4	1, 1, 4	0.07	29.9	35.0
	32, 32, 33, 33	21	3.3	1, 1, 2, 4	0.15	29.7	33.8

to be used as part of the coordination process. Table II shows that, in general, the solution time of the subsystems decreases as the number of partitions increases. The total subsystems solution time is dependent on the way in which the networks are partitioned, as well as being dependent on the size of the subsystem. If the partitioning causes one of the subnetworks to include a number of conflicting measurements, the number of iterations taken to solve the subsystem will increase. An increase in the number of iterations taken will inevitably increase the subsystem's solution time (as in the case of the third network when divided into four, Table II shows that one subsystem requires 22 iterations before a solution is found). The coordination procedure is also sensitive to the chosen network partitioning. The coordination time is dependent on the number of cutlines (as shown in Table II). The time taken to evaluate the bounds is, however, independent of the partitioning as it is implemented on the entire network's minimal measurement set, so it remains approximately constant for each of the partitionings of the four networks. Confidence limit analysis constitutes a large proportion of the total solution time. This suggests that a large saving of time has been made by parallelising the state estimation procedure, and also suggests that a further saving can be made by the introduction of parallelisation of the confidence limit analysis procedure.

With consideration of the above factors, the dependence of the subsystem solution time on the size of the subsystem will normally result in the total solution time of the distributed probabilistic state estimation algorithm decreasing as the number of partitions increases and the number of cutlines decreases.

6. CONCLUSIONS

The parallel state estimation algorithm has been found to be an effective means of overcoming the problems associated with the computational complexity of the state estimators. The algorithm requires sending only small amounts of data between the subsystems and the coordinating task, thus making it suitable for distributed processor implementation. The use of the algorithm is particularly effective when the number of subsystems equals the number of distributed computing nodes, allowing simultaneous solution of all the subsystems.

The subsequent quantification of the inaccuracy of calculated system states, caused by input data uncertainty, led to an efficient and more informative distributed state estimator. The results have indicated the effectiveness of the method by reflecting on the convergence rate and the general effect caused

by inaccurate consumption estimates. It is expected that the full advantage of parallel state estimation will be gained when dealing with large networks of several hundred nodes.

References

- [1] Bargiela, A. (1992). Nonlinear Network Tearing Algorithm for Transputer System Implementation, *Proc. TAPA '92*, Nov., pp 19-24.
- [2] Bargiela, A. and Hainsworth, G. D. (1988). Telemetry System Design and On-Line Decision Support with 'TCLAS' Software, *Computer Applications in Water Supply*, 2, pp 43-58.
- [3] Bargiela, A. and Hainsworth, G. D. (1988). Pressure and Flow Uncertainty in Water systems, *Jnl. of Water Resources, Planning and Management*, 2, pp 43-58.
- [4] Brdys, M. A., Abdullah, N. and Roberts, P. D. (1990). Augmented Model-Based Double Iterative Loop Techniques for Hierarchical Control of Complex Industrial Processes, *Int. J. Control*, 52(3), pp 549-570.
- [5] Hainsworth, G. D. (1988). Measurement Uncertainty in Water Distribution Telemetry Systems, *PhD thesis*, Nottingham Polytechnic.
- [6] Hartley, J. K. (1996). Parallel Algorithms for Fuzzy Data Processing with Application to Water Systems, *PhD thesis*, The Nottingham Trent University.
- [7] Hartley, J. K. and Bargiela, A. (1995). Parallel Simulation of Large-Scale Water Distribution Systems, *Proc. of ESM '95*, June, pp 723-727.
- [8] Hosseinzaman, A. and Bargiela, A. (1992). Parallel Simulation of Nonlinear Networks using Diakoptics, *Proc. PACTA '92*, Sept.
- [9] El-Keib, A. A. and Singh, H. (1992). Fast Linear Programming State Estimation Using the Dual Formulation, *Transactions on Power Systems*, May, 7(2), pp 620-628.
- [10] Irving, M. R. and Sterling, M. J. H. (1990). Optimal Network Tearing Using Simulated Annealing, *IEE Proc. C (Generation, Transmission and Distribution)*, Jan., 137(1), pp 69-72.
- [11] Kron, G. (1963). 'Diakoptics: The Piecewise Solution to Large Scale Systems', MacDonald.
- [12] Lin, J., Li, L. A., Wan, B. W. and Roberts, P. D. (1989). Triple Iterative Loop Technique for Optimizing control of Large-Scale Steady-State Systems, *Int. J. Systems Sci.*, 20(11), pp 2107-2124.
- [13] Rutowski, J. (1987). An Efficient Algorithm for Branch Tearing Large Scale Networks, *Circuit Theory and Design '87*, Sept., pp 693-698.