PARALLEL SIMULATION OF LARGE-SCALE WATER DISTRIBUTION SYSTEMS

J.K. Hartley, A. Bargiela, R.J. Cant
Real Time Telemetry Systems, Department of Computing,
The Nottingham Trent University, Burton Street, Nottingham NG1 4BU, U.K.
Tel. 0115 - 9418418, Fax. 0115 - 9486518, E-mail: andre@uk.ac.ntu.doc

ABSTRACT

This paper describes the development of parallel algorithms for the processing of telemetry data in the specific context of water distribution networks. Today's increasing complexity of such systems, together with the requirement for smaller discretisation intervals means that the task of simulating large-scale water distribution networks frequently outgrows the computational power of economically viable single processor systems. One solution to this problem is to partition the computational task into concurrently executable sub-tasks and to implement them on a parallel or distributed processing hardware. This paper compares the performance of two parallel state estimators implemented on a transputer platform. The results obtained with these estimators (to be presented in full at the conference) indicate the effectiveness of each technique by reflecting on the convergence rate of the state estimators, the capability of identification/rejection of bad measurements, the consideration of measurement accuracies and the possible recovery from measurement failure situations.

INTRODUCTION

Water distribution systems are large-scale geographically distributed systems that consist of a great number of hydraulic elements. For the purpose of operational control of such systems one needs to derive a simplified model of the system that is sufficiently informative to allow deduction of the network behaviour in response to specific controls. Consequently, water distribution networks are modelled as topological networks of nodes (representing groups of water users) connected by links which represent pipes, pumps or valves in the physical system. The level of aggregation that can be afforded in this process is determined by the required accuracy of system modelling. This implies that frequently even the network model of a realistic system will consist of many hundreds of nodes and links.

Within this context, it is clear that system operators require some assistance in processing the detailed telemetry data about flows and pressures in the network to be able to derive an overall view of system operation. This is particularly true when telemetry data needs to be processed in real time.

An additional concern when dealing with such systems is the presence of errors in telemetered data. These can be caused by transducer or communication line failures and have to be filtered out from the correct data set. Such filtering is feasible only if there are some redundant measurements in the set against which the telemetry data can be checked. It is now widely recognised that the combined data processing and error detection functionality is best provided through the process of state estimation.

Depending on the choice of state variables the mathematical model that underlies the state estimation is constructed to reflect either the flow conservation law or the loop head loss law. If the state variables are the nodal pressures, the flows in the links are expressed as functions of pressures and the model represents mass balance equations in each node.

$$\sum_{j \in \hat{M}_i} f_{ij}(\mathrm{h}) = b_i \qquad i = 1, \ldots, N \qquad (1)$$

where N is a number of network nodes,

$\mathrm{h} = [h_1, \ldots, h_N]^T$ is a state vector of nodal pressures,

$\hat{M}_i$ is a set of nodes incident to node i,

b_i is a node balance.

Alternatively, if the loop flows are state variables, the pressure drops across each link are the functions of flows in the corresponding loop.

$$\sum_{j \in \hat{L}_i} h_{ij}(\mathrm{f}) = 0 \qquad i = 1, \ldots, L \qquad (2)$$

where L is a number of independent loops,

$\mathrm{f} = [f_1, \ldots, f_L]^T$ is a state vector of loop flows,

$\hat{L}_i$ is a set of links forming the loop i.

Although the number of equations in model (1) is greater than in model (2), N>L, the mass balance equations are usually considered better suited for large systems modelling owing to their sparse structure.

The minimum set of equations (1) is then augmented by the equations representing additional flow measurements (expressed as functions of pressures - state variables) and possible direct measurements of pressures. This results in a full set of measurements of size M. The additional measurements provide redundancy needed for error detection. Thus the general mathematical model for state estimation is as follows

$$g(\boldsymbol{x}) = \mathbf{z} + \omega \qquad (3)$$

where g is the non-linear network function,

$\boldsymbol{x}$ is Nx1 dimensional state vector,

$\mathbf{z}$ is Mx1 measurement vector (M>N),

ω is Mx1 vector of measurement errors.

The state estimation attempts to satisfy equations (3) while minimising the measurement errors ω

$$\underset{x}{minimise} \qquad \| g(\mathbf{x}) - \mathbf{z} \|_n \qquad (4)$$

Typically 'n' is selected to be 2 so the solution of (4) is a least squares optimisation.

Due to the non-linearity of g($\mathbf{x}$), an iterative approach of solving the line-

arised equations is taken. The Newton-Raphson method obtains the solution to a system of non-linear equations, via iteratively solving a system of linear equations. Primarily g(**x**) is expanded by an initial estimate of the state vector $\boldsymbol{x}_o$ using a Taylor series of first order.

$$\boldsymbol{g}(\boldsymbol{x}_{(k+1)}) = \boldsymbol{g}(\boldsymbol{x}_{(k)}) + \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_{(k)}} \bullet \Delta \boldsymbol{x}_{(k)} \qquad (5)$$

denoting

$$\boldsymbol{g}(\boldsymbol{x}_{k+1}) - \boldsymbol{g}(\boldsymbol{x}_{\boldsymbol{k}}) = \Delta \boldsymbol{z} \qquad (6)$$

$$\boldsymbol{x}_{k+1} - \boldsymbol{x}_{\boldsymbol{k}} = \Delta \boldsymbol{x} \qquad (7)$$

$$\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\boldsymbol{x}_k} = J \qquad (8)$$

the equation that needs to be solved in each iteration is

$$\Delta \boldsymbol{z} = J \bullet \Delta \boldsymbol{x} \qquad (9)$$

which enables the correction to the state vector to be made

$$\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{(k)} + \Delta \boldsymbol{x} \qquad (10)$$

PARALLEL STATE ESTIMATION

The increasing size and complexity of today's water distribution systems leads to an increase in the size of the state estimation problem. By solving a large distribution system as a number of smaller subsystems, the overall size of the state estimation problem is reduced. This evaluation of the entire problem as a number of localised subsystem estimates suggests the natural progression to parallelism of the procedure.

Recent advances in computing hardware point towards the development of distributed parallel programs as a cost-effective scheme (Bargiela et al, 1993). The adopted model of computation is the SPMD (single program, multiple data) structure - where a single program performs the same task on different data. The distributed state estimation of water systems has been developed with the SPMD model of computation in mind. The large-scale networks are subdivided into smaller, more manageable subnetworks by selecting a number of cutlines from the links within the network (Irving and Sterling 1990; Rutowski 1987). The selection of these cutlines are critical to the optimal performance of the state estimator. The locations of the cutlines have an effect on the size and number of subsystems to be solved and the number of selected cutlines is proportional to the complexity of the coordination problem for the subsystems. Each subnetwork is solved independently and then subsequently coordinated with the solution of the neighbouring subnetworks. The coordination procedure must optimise the compatibility of the subsystems at their connecting points (Brdys et al 1990b) in order to achieve the overall solution.

The state estimation algorithms were implemented on a distributed transputer network which provided a convenient mapping between the logical model of computation and the physical hardware. A distribution system tends to grow by adding subnetworks to the original form. Hence, the subdivisions required by the state estimation algorithms are implicitly determined. Thus, by exploitation of the flexibility inherent to the transputer architecture, the addition of an extra processor for each augmented subnetwork results in an increase in computational power and parallelism without any major alteration to the software.

TWO DISTRIBUTED STATE ESTIMATORS

Two distributed state estimators have been developed, both being based on an earlier diakoptical simulation algorithm implemented for use with nonlinear networks (Bargiela 1992; Hosseinzaman and Bargiela 1992) The two state estimation procedures proceed with subdivision of the distribution systems, independent solution of the subsystems on a transputer network and coordination of the subsystem results. The manner in which coordination is achieved characterises the different decomposition methods. However, any decentralised method relies on the solution of a larger number of smaller problems. To take into account the interactions between the subproblems, solution usually proceeds iteratively with information exchange between the subproblems and a coordinating master problem. In the case of the two developed distributed state estimators, each of the subproblems are solved by a conventional centralised method. They are solved individually and the solutions are recombined in some way to achieve the solution of the overall problem.

Application of such decomposition-coordination techniques, to the state estimation problem, have already been shown to be enhanced by parallelism (Osiadacz and Salimi 1988). This form of decomposition favours implementation on a parallel processing hardware, giving rise to an approximately n-fold improvement in the computational efficiency. However, an improvement is seen even when the task is solved on a sequential computer, that is the subsystem solutions are generated in succession rather than in parallel.

By applying these distributed parallel state estimators to water distribution networks, the operators of the system will have available the estimated state of the system in real-time. So they will have information concerning all of the pressures and the flows throughout the network at each period in time. Thus, any alterations to be made to the system, such as the opening of a valve or the starting-up of a pump, can be decided upon with more reliable information about the true state of the entire network.

The Two-Phase Algorithm - State Estimation By Initial Extraction Of Redundant Measurements

The first state estimation procedure progresses in two phases. Firstly the solution to a minimum measurement set problem is obtained. Then the effects of the remaining measurements on the state estimate are considered. This is achieved by splitting the components of the network equations (3) into appropriate segments - the minimum measurement set and the remaining redundant measurements:

$$\boldsymbol{g}(\boldsymbol{x}) = [\hat{\boldsymbol{g}}(\boldsymbol{x}), \tilde{\boldsymbol{g}}(\boldsymbol{x})]^{\boldsymbol{T}} \qquad (11)$$

$$\boldsymbol{z} = [\hat{\boldsymbol{z}}, \tilde{\boldsymbol{z}}]^{T} \qquad (12)$$

$$\omega = [\hat{\omega}, \tilde{\omega}]^{T} \qquad (13)$$

where $\hat{\boldsymbol{g}}(\boldsymbol{x})$ is N equations (the minimal measurement set),

$\hat{\boldsymbol{z}}$ is Nx1 vector,

$\hat{\omega}$ is Nx1 vector,

and $\tilde{\boldsymbol{g}}(\boldsymbol{x})$ is M-N equations (the redundant measurements),

$\tilde{\boldsymbol{z}}$ is (M-N)x1 vector,

ω is (M-N)x1 vector.

Thus equation (3) may be written in the form

$$\begin{bmatrix} \hat{\boldsymbol{g}}(\boldsymbol{x}) \\ \tilde{\boldsymbol{g}}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{z}} \\ \tilde{\boldsymbol{z}} \end{bmatrix} + \begin{bmatrix} \hat{\omega} \\ \tilde{\omega} \end{bmatrix} \qquad (14)$$

The initial N equations (depicting the minimum measurement set $\hat{\boldsymbol{z}}$) are solved by splitting the network into k smaller partitions and solving these subnetworks in parallel on a distributed architecture of processors.

$$\begin{bmatrix} \hat{\boldsymbol{g}}_1(\boldsymbol{x}) \\ \hat{\boldsymbol{g}}_2(\boldsymbol{x}) \\ \cdots \\ \hat{\boldsymbol{g}}_s(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \hat{\boldsymbol{z}}_1 \\ \hat{\boldsymbol{z}}_2 \\ \cdots \\ \hat{\boldsymbol{z}}_s \end{bmatrix} + \begin{bmatrix} \hat{\omega}_1 \\ \hat{\omega}_2 \\ \cdots \\ \hat{\omega}_s \end{bmatrix} \qquad (15)$$

In the same way that equation (3) progresses to equation (9), the linearised form of equation (15) is written as

$$\begin{bmatrix} \Delta\hat{\boldsymbol{z}}_1 \\ \Delta\hat{\boldsymbol{z}}_2 \\ \cdots \\ \Delta\hat{\boldsymbol{z}}_s \end{bmatrix} = \hat{J} \bullet \Delta\boldsymbol{x} \qquad (16)$$

where

$$\left[\frac{\partial\hat{\boldsymbol{g}}_1}{\partial\boldsymbol{x}} \; \frac{\partial\hat{\boldsymbol{g}}_2}{\partial\boldsymbol{x}} \; \cdots \; \frac{\partial\hat{\boldsymbol{g}}_s}{\partial\boldsymbol{x}}\right]^T = \hat{J} \qquad (17)$$

The subnetworks are divided by removing any interconnecting pipes, the removal of these pipes must be considered when solving the subnetworks and when coordinating the results. So, in order to partition the N equations into s disjoint segments, equation (16) is written as

$$\begin{bmatrix} \Delta\hat{\boldsymbol{z}}' \\ \boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} \hat{J}' & -C_{\psi\alpha} \\ C_{\alpha\psi} & K \end{bmatrix} \bullet \begin{bmatrix} \Delta\hat{\boldsymbol{x}} \\ \Delta\boldsymbol{y} \end{bmatrix} \qquad (18)$$

where $\hat{J}' = diag\left[\frac{\partial\hat{\boldsymbol{g}}_1}{\partial\boldsymbol{x}} \; \frac{\partial\hat{\boldsymbol{g}}_2}{\partial\boldsymbol{x}} \; \cdots \; \frac{\partial\hat{\boldsymbol{g}}_s}{\partial\boldsymbol{x}}\right]$ with

$\frac{\partial\hat{\boldsymbol{g}}_1}{\partial\boldsymbol{x}} \in R^{N_1 \times N}, \ldots, \frac{\partial\hat{\boldsymbol{g}}_s}{\partial\boldsymbol{x}} \in R^{N_s \times N}, \sum_{i=1}^{s} N_i = N$, this maps

directly onto the Jacobian matrix $\hat{J}$ except for the elements corresponding to the removed pipes

$\Delta\hat{\boldsymbol{z}}' = [\Delta\hat{\boldsymbol{z}}'_1, \Delta\hat{\boldsymbol{z}}'_2, \ldots, \Delta\hat{\boldsymbol{z}}'_s]^T$ denoting the right-handside of equation (16) changed to take into account the compensating flows from the removed pipes,

Y is an augmented vector of zeroes for relevant pressure balances and values of net mass imports/exports for each subnetwork, ,pipes,

$C_{\alpha\psi} = C_{\psi\alpha}^T$ is a connectivity matrix between the subnetworks and the removed pipes,

K is a coordinating matrix,

$\Delta\boldsymbol{x} = [\Delta x_1, \Delta\boldsymbol{x}_2, \ldots, \Delta\boldsymbol{x}_s]^T$ depicting the coordinated results of the subnetwork solutions, and

$\Delta\boldsymbol{y}$ represents the coordinating variables (increments to the reference pressures in the subsystems and the flows between the subsystems).

The state estimate $\hat{\boldsymbol{z}}$ based on the N measurements may thus be determined using the equation

$$\hat{\boldsymbol{x}}_{(k+1)} = \hat{\boldsymbol{x}}_{(k)} + \Delta\hat{\boldsymbol{x}} \qquad (19)$$

(see equation (10)).

Subsequently, on solution and coordination of these minimum measurement subsets, the redundant measurements $\tilde{\boldsymbol{z}}$ are processed yielding the final state estimate for the whole system. This is in effect an application of the standard recursive least squares technique.

The algorithm begins with the calculation of the covariance matrix, $P_o = [\hat{J}^T R^{-1} \hat{J}]^{-1}$, where R is a diagonal NxN matrix.

Taking the final estimate derived from the equation (19) to be the initial estimate for equation (20), the measurements from the $\tilde{\boldsymbol{g}}(\boldsymbol{x})$ set can be processed one at a time to obtain the final estimate **x.** The basic equations are

$$\boldsymbol{x}_{(k)} = \boldsymbol{x}_{(k-1)} + \boldsymbol{W}_{(k)} [\tilde{\boldsymbol{z}}_{(k)} - \boldsymbol{j}_{(k)} \boldsymbol{x}_{(k-1)}] \qquad (20)$$

$$P_{(k)} = [I - \boldsymbol{W}_{(k)} \boldsymbol{j}_{(k)}] P_{(k-1)} \qquad (21)$$

$$\boldsymbol{W}_{(k)} = P_{(k-1)} (\boldsymbol{j}_{(k)})^T [\sigma^2_{(k)} + \boldsymbol{j}_{(k)} P_{(k-1)} (\boldsymbol{j}_{(k)})^T]^{-1} \qquad (22)$$

where $\boldsymbol{x}_{(k)}$ is the state estimate after processing k of the measurements in $\tilde{\boldsymbol{z}}$,

$P_{(k)}$ is the state covariance matrix corresponding to $\boldsymbol{x}_{(k)}$,

$\tilde{\boldsymbol{z}}_{(k)}$ is the k-th measurement in $\tilde{\boldsymbol{z}}$,

$\boldsymbol{j}_{(k)}$ is the linearised k-th measurement equation vector in $\tilde{\boldsymbol{z}}$,

$\sigma^2_{(k)}$ is the variance of noise on the k-th measurement in $\tilde{\boldsymbol{z}}$,

$\boldsymbol{W}_{(k)}$ is the weighting vector for the k-th measurement in $\tilde{\boldsymbol{z}}$.

The general strucure of the distributed state estimation algorithm using the post-processing of redundant measurements is given in Figure 1 and the algorithm may be summarised as follows:

1. Read in the system description data.

2. Form subsystem data packets by removal of a number of links. Update the data packets with compensating flows. Send the packets to individual simulators, distributed across a network of transputers.

3. Calculate each of the uncoordinated subsystem solutions by linear regression to the square set of linearised equations. Implement an incremental improvement to the initial solution, using the iterative Newton-Raphson procedure.

4. Collate and coordinate the partial solutions, taking into account the effect of alterations necessary 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 then STOP otherwise repeat from Step 2.

6. Calculate the initial estimate covariance matrix - indicating the

relationship between each of the derived state variables.

7. Update the state estimate by processing one measurement from the set of redundant measurements, $\tilde{\boldsymbol{z}}$, with the appropriate weighting.

8. Update the state covariance matrix to take into account the effect of this redundant measurement.

9. Modify the measurement weights, by utilisation of the current covariance matrix.

10. Repeat 7,8,9 for every measurement in the set of redundant measurements.

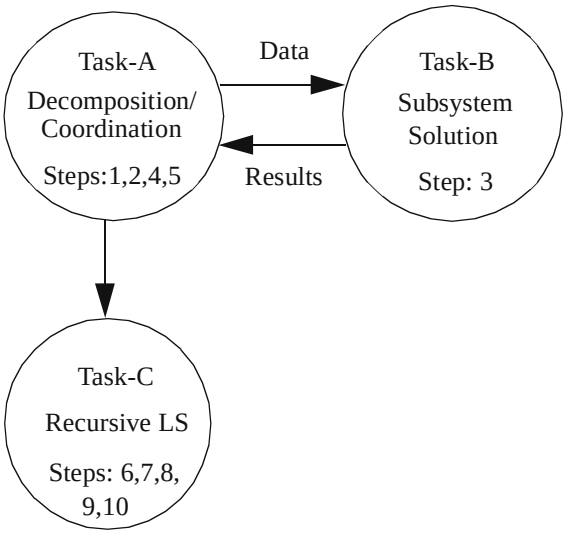

Figure 1. Program Structure of the Two Phase Algorithm

The error rejection capability of the proposed method for state estimation of a physical system has been tested on a 130-node network, subdivided into two subnetworks, each consisting of 65 nodes. The minimal measurement set includes the combined values of the consumption inflow at each node. The data is obtained by exact network simulation. This original set of measurements is subsequently increased by the introduction of ten head measurements and ten flow measurements. The effect of any *bad data* measurements is simulated by modification of measured values. However, processing redundant measurements one-at-a-time raises questions about the ability of the algorithm to detect measurement errors. Other than this limitation, the algorithm's error rejection capability is seen to be both reliable and accurate throughout the testing.

The One-Phase Algorithm - State Estimation By Inclusion Of Redundant Measurements

The second algorithm processes the overdetermined set of equations for each of the subnetworks. So all M equations of (3) are partitioned into s subnetworks

$$\begin{bmatrix} \boldsymbol{g_1}(\boldsymbol{x}) \\ \boldsymbol{g_2}(\boldsymbol{x}) \\ \cdots \\ \boldsymbol{g_s}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ \cdots \\ \boldsymbol{z}_s \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \omega_2 \\ \cdots \\ \omega_s \end{bmatrix} \qquad (23)$$

Thus similarly to equation (9) and equation (16), the linearised form of (23) is achieved

$$\begin{bmatrix} \Delta\boldsymbol{z}_1 \\ \Delta\boldsymbol{z}_2 \\ \cdots \\ \Delta\boldsymbol{z}_s \end{bmatrix} = J \bullet \Delta\boldsymbol{x} \qquad (24)$$

where

$$\begin{bmatrix} \frac{\partial \boldsymbol{g_1}}{\partial \boldsymbol{x}} & \frac{\partial \boldsymbol{g_2}}{\partial \boldsymbol{x}} & \cdots & \frac{\partial \boldsymbol{g_k}}{\partial \boldsymbol{x}} \end{bmatrix}^T = J \qquad (25)$$

with $\frac{\partial \boldsymbol{g_1}}{\partial \boldsymbol{x}} \in R^{M_1 \times N}, \ldots, \frac{\partial \boldsymbol{g_s}}{\partial \boldsymbol{x}} \in R^{M_s \times N}, \sum_{i=1}^{s} M_i = M$

In a similar fashion to the first algorithm, the resulting equation to be solved in order to determine the coordinated state estimate of these s partitions is the matrix

$$\begin{bmatrix} \Delta z' \\ \boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} J' & -C_{\psi\alpha} \\ C_{\alpha\psi} & K \end{bmatrix} \bullet \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \qquad (26)$$

(see equation (18)).
The different elements of this equation include

$\Delta\boldsymbol{z'} = [\Delta\boldsymbol{z'}_1, \Delta\boldsymbol{z'}_2, \ldots, \Delta\boldsymbol{z'}_k]^T$ denoting the right-handside of equation (24) changed to take into account the compensating flows from the removed pipes,

$J' = diag\left[\frac{\partial \boldsymbol{g_1}}{\partial \boldsymbol{x_1}} \frac{\partial \boldsymbol{g_2}}{\partial \boldsymbol{x_2}} \cdots \frac{\partial \boldsymbol{g_s}}{\partial \boldsymbol{x_s}}\right]$, this maps directly onto the Jacobian matrix J except for the elements corresponding to removed pipes,

Y is an augmented vector of zeroes for relevant pressure balances and values of net mass imports/exports for each subnetwork,
,pipes,

$C_{\alpha\psi} = C_{\psi\alpha}^T$ is a connectivity matrix between the subnetworks and the removed pipes,

K is a coordinating matrix,

$\Delta\boldsymbol{x} = [\Delta x_1, \Delta\boldsymbol{x}_2, \ldots, \Delta\boldsymbol{x}_s]^T$ depicting the coordinated results of the subnetwork solutions, and

$\Delta\boldsymbol{y}$ represents the coordinating variables (increments to the reference pressures in the subsystems and the flows between the subsystems).

The subnetwork state estimation tasks are thus solved concurrently and are coordinated, to give the final solution of the state estimate.

$$\boldsymbol{x}_{(k+1)} = \boldsymbol{x}_{(k)} + \Delta\boldsymbol{x} \qquad (27)$$

There is no need for post-processing of redundant measurements as these are considered during the state estimation procedure of the subnetworks, so it involves one-phase only.

This one-phase algorithm can be summarised as follows:

1. Read in the system description data.

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.

3. Calculate each of the uncoordinated subsystem solutions by weighted least squares regression to the linearised set of equations.

$$\begin{bmatrix} 0 & -I & J \\ -I & W & 0 \\ J^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \psi \\ \Delta x \end{bmatrix} = \begin{bmatrix} \Delta z \\ 0 \\ 0 \end{bmatrix}$$

where W is the weight matrix $\frac{1}{\Delta x}$.

Implement an incremental improvement to the initial solution, using the iterative Newton-Raphson procedure.

4. Collate and coordinate the partial solutions, taking into account the effect of alterations necessary 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 then STOP otherwise repeat from Step 2.

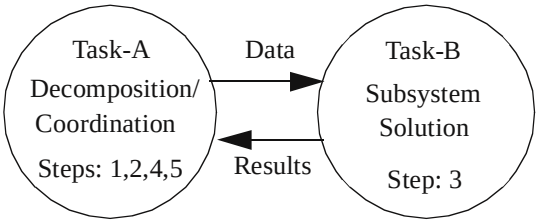

Figure 2. Program Structure of the One-Phase Algorithm.

COMPARISON OF THE ALGORITHMS

While the idea of a decomposed solution of large-scale linear systems has been well researched, the decomposition of non-linear systems remains a relatively unexplored area. The development of earlier work on decomposed simulation of nonlinear networks, concerned with the processing of redundant measurements, has been discussed. Both state estimation algorithms enable identification/rejection of bad measurements, consideration of measurement accuracies, recovery from measurement failure situations etc.. None of the above are feasible with simple load-flow simulations, ie. measurement sets with no redundant measurements.

Due to the contrasting approaches within the decomposition stages of the two algorithms, both the solution and coordination stages are also inherently different when the two algorithms are compared. The first algorithm solves each subnetwork by utilisation of an $N_i \times N_i$ Jacobian matrix, whilst the second algorithm uses a $(2M_i + N_i) \times (2M_i + N_i)$ augmented matrix formulation. This method of state estimation by initial extraction of redundant measurements has thus been proven to have a faster convergence rate than the second method of state estimation by inclusion of redundant measurements. However, as far as accuracy is concerned, (even with the inclusion of inaccurate measurements) the second method can be relied upon - this in contrast to the first method where the solution is dependant on the order of the measurements within the data file. The two estimation methods will be compared and contrasted in greater detail at the conference.

In order to achieve an efficient and scalable state estimation it was decided to implement the algorithms on a transputer platform. The transputer platform has proven to be a suitable implementation vehicle for the distributed state estimation algorithm. It is worth noting, however, that the algorithm structure implies that, if possible, Task-C of the first algorithm should be executed on the highest performance node.

REFERENCES

Bargiela, A. 1992. "Nonlinear Network Tearing Algorithm for Transputer System Implementation.", *Proc. TAPA '92*, (Nov.): 19-24.

Bargiela, A., A.D.S. Argile, J.K. Hartley, W. Walton. 1993. "Parallel Processing for Probabilistic Decision Support in Water Distribution Systems." *ITA SERC Seminar,* Brunel University, (Sept.): 255-64.

Brdys, M.A., N. Abdullah, P.D. Roberts. 1990a. "Hierarchical Adaptive Techniques for Optimizing Control of Large-Scale Steady-State Systems: Optimality, Iterative Strategies, and their Convergence." *IMA Journal of Math. Control & Info.*, Vol. 7: 199-233.

Brdys, M.A., N. Abdullah, P.D. Roberts. 1990b. "Augmented Model-Based Double Iterative Loop Techniques for Hierarchical Control of Complex Industrial Processes." *Int. J. Control,* Vol. 52, No. 3: 549-570.

Chen, S., B. Ulanicki, J.P. Rance, B. Coulbeck. 1993. "A General Purpose Optimizer for Large Scale Water Supply and Distribution Systems." *Proc. Int. Conf. on Integrated Computer Applications in Water Supply*, (Sept.): 239-58.

Coulbeck, B., M.A. Brdys, C.-H. Orr, J.P. Rance. 1988. "A Hierarchical Approach to Optimized Control of Water-Distribution Systems: Part II." *Optimal Control Applications & Methods*, Vol. 9,: 109-26.

Van Cutsem, T., M. Ribbens-Pavella. 1983. "Critical Survey of Hierarchical Methods for State Estimation of Electric Power Systems." *IEEE Trans. on Power Apparatus and Systems,* Vol. PAS-102, No. 10 (Oct.): 3415-23.

Fallside, F., B. Perry. 1975. "Hierarchical Optimisation of a Water Supply Network." *IEE Proc.*, Vol. 122, No. 2 (Feb.): 202-208.

Hosseinzaman, A., A. Bargiela. 1992. "Parallel Simulation of Nonlinear Networks using Diakoptics." *Proc. PACTA '92* (Sept.).

Irving, M.R., M.J.H. Sterling. 1990. "Optimal Network Tearing Using Simulated Annealing." *IEE Proceedings C (Generation, Transmission and Distribution)* , Vol. 137, Iss. 1 (Jan.):,69-72.

Osiadacz, A.J., M.A. Salimi. 1988. "Hierarchical Dynamical Simulation of Gas Flow in Networks." *Civ. Engng. Syst* , Vol. 5 (Dec.): 199-205.

Rutowski J. "An Efficient Algorithm for Branch Tearing Large Scale Networks." *Circuit Theory and Design 87* (Sept.): 693-8.