PROBABILISTIC SIMULATION OF LARGE-SCALE WATER DISTRIBUTION SYSTEMS

Joanna K. Hartley and Andrzej Bargiela
Real Time Telemetry Group
Department of Computing, The Nottingham Trent University
Burton Street, Nottingham NG1 4BU
Tel: +44 (0) 115 9418418 ext 2162, Fax: +44 (0) 115 9486518, Email andre@doc.ntu.ac.uk

ABSTRACT

This paper describes the development of a system that can assist operators of large-scale distribution systems in taking control decisions based on qualitative and quantitative evaluation of the system state. Simulation involves the cross-referencing of system measurements by relating actual meter readings and estimated nodal consumptions to the values calculated from the mathematical model of the system. Meter readings from the distribution systems invariably include a degree of inaccuracy. This inaccuracy and the inexact approximation of nodal consumptions lead to discrepancies within the state estimate. So the simulation algorithm has been further developed to show the effect of these discrepancies in the form of state confidence limits. These confidence limits must be evaluated in real-time for the operators to be able to make any sensible control decisions based on these set of values (Bargiela and Hainsworth 1988a). However, the distribution systems are continually increasing in size which ultimately leads to an increase in size of the simulation procedure. The effect of this increase may be decreased by solving the distribution systems as a set of smaller subsystems in parallel.

INTRODUCTION

The simulation of any distribution system - such as electric power, gas or water distribution systems - requires an insight into the system response to specific operational controls. Simulation can be considered as the pivotal component of any operational software suite - enabling the cross-referencing of system measurements by relating the actual meter readings to the values calculated from the mathematical model of the system. This cross-referencing often makes it possible to identify and localise erroneous measurements and to advise the operator on appropriate remedial action.

Use of the mathematical model of the system requires knowledge of the current state of the distribution system. The state of the system, at any time, is determined from a number of variables and fixed values, such as metered pressure measurements, estimated consumption values and physical links of the system. The solution of the mathematical model of the system depends critically on the number of independent network equations being equal to the number of state variables. So, it is common practice to build in a degree of redundancy into the specification of a telemetry system. This redundancy means that the failure of a single meter does not prevent the computations from proceeding. The optimisation process that takes place - called state estimation - enables the detection and identification of measurement errors which otherwise might have adversely affected the system controls.

State estimation is, however, approximately quadratically dependent on the size of the physical network. As a means of controlling the computational complexity as the size of the distribution systems increase, decentralised control is introduced which dedicates individual computing resources to subsystems of the original system. In the case of distributed systems, the physical structure of the problem can be utilised when decomposing it into a number of smaller subproblems. Each of these subproblems will be solved by a conventional centralised method.

Any state estimation procedure involves solving an over-determined set of equations describing mass-balances in network nodes and pressure-flow relationships:

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

where **x** is a vector of N state variables, called the state vector, which consists of all the nodal pressures and the inflows into the fixed-head nodes; **z** is the measurement vector which consists of M real measurement values and pseudomeasurements; **g** is a non-linear function relating the state variables to the measurements (real and pseudo); and ω is a vector of measurement errors.

The state estimation procedure involves minimising the measurement errors ω by assigning appropriate values to the state estimate **x**. The least absolute values method (2) is considered to be the most suitable minimisation method when applied to the state estimation of water distribution systems.

$$minimise|\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{z}| \qquad (2)$$

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

For real-time distribution system applications, the technique chosen to solve the network equations must be fast, accurate and numerically stable. The Newton-Raphson method is a powerful numerical method for solving systems of non-linear equations offering good computational performance, numerical stability and efficiency.

The Newton-Raphson method obtains the solution to a system of non-linear equations, by iteratively solving a system of linear equations. Primarily $\boldsymbol{g}(\boldsymbol{x})$ is expanded by an initial estimate of the state vector $\boldsymbol{x_0}$ using a Taylor series of first order, leading to the iterative equation

$$\boldsymbol{g}(\boldsymbol{x}_{(k+1)}) = \boldsymbol{g}(\boldsymbol{x}_{(k)}) + \left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x} = \boldsymbol{x}_{(k)}} \cdot \Delta \boldsymbol{x}_{(k)} \qquad (3)$$

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

By denoting

$$\boldsymbol{g}(\boldsymbol{x}_{(\boldsymbol{k}+\boldsymbol{1})}) - \boldsymbol{g}(\boldsymbol{x}_{(\boldsymbol{k})}) = \Delta\boldsymbol{z} \qquad (4)$$

$$\boldsymbol{x}_{(\boldsymbol{k}+\boldsymbol{1})} - \boldsymbol{x}_{(\boldsymbol{k})} = \Delta\boldsymbol{x} \qquad (5)$$

$$\left.\frac{\partial \boldsymbol{g}}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x} = \boldsymbol{x}_{(\boldsymbol{k})}} = J \qquad (6)$$

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

$$\Delta\boldsymbol{z} = J \cdot \Delta\boldsymbol{x} \qquad (7)$$

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

$$\boldsymbol{x}_{(\boldsymbol{k}+\boldsymbol{1})} = \boldsymbol{x}_{(\boldsymbol{k})} + \Delta\boldsymbol{x} \qquad (8)$$

Such a traditional, deterministic state esimator can provide misleading information. The estimate for the state of the system is dependent on the pseudo- and metered measurement data, but this data may be inaccurate. The values of the metered measurements are generally accurate to within a small percentage error, while the errors of the pseudomeasurements (estimated consumption values) can be more extreme. A large number of meters within the distribution network is, therefore, desirable to counteract the effect of the pseudomeasurement errors, but the cost is too great.

Pseudomeasurements, along with an affordable number of metered measurements, are thus used when determining the state of the system. However, use of the inaccurate pseudomeasurements still has a major impact on the uncertainty of the final state estimate. So, if the estimates are to be used as a basis for making control decisions it is necessary to know by how much they are in error. For this reason, it is essential that the state estimate variables are described in terms of their upper and lower bounds, ensuring an ease of understanding by indicating the potential error associated with each variable.

Probabilistic state estimation solves a non-linear set of equations (1), resulting in an interval state estimate. A series of compromise solutions are considered, which take into account the existing relation between different measurements with their discrepancies. This enables the distribution systems to be viewed with representation of every possible outcome - a necessary requirement for on-line monitoring.

DISTRIBUTED PROBABILISTIC SIMULATION

In any procedure of simulation, the difficulty of satisfying the conflicting requirements on speed and accuracy increases with the system's size (El-Keib and Singh 1992; Lin et al. 1989). If the network is too complex the memory and computational requirements become excessive for real-time analysis. The general consensus is that to combat such problems the system must be split into subsystems, introducing the notion of distributed simulators, via parallel programming. The parallel structure of the algorithms enables modular augmentation of the processing hardware (transputer network) in line with the increase of the size of the simulated systems.

So, with the ever-increasing size of distribution networks, probabilistic simulation is best implemented as a distributed procedure. The large-scale networks are subdivided into smaller, more manageable subnetworks (Irving and Sterling 1990; Rutowski 1987) by cutting a number of links (Figure 1). The subnetworks are updated to compensate for the flow lost in the cut links. Each updated 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. 1990) in order to achieve the overall solution.

Figure 1. Partitioning the network

a) original system

b) equivalent partitioned system

The algorithm used is based on an earlier diakoptical simulation algorithm implemented for use with nonlinear networks (Bargiela 1992; Hosseinzaman and Bargiela 1992). The distributed probabilistic simulator then progresses to sensitivity analysis of the state variables, which is the same as our earlier sequential algorithm (Bargiela and Hainsworth 1988b; Hainsworth 1988). The upper and lower bounds of the calculated state vector are determined by relating the sensitivity of the state vector to the measurement vector. The outcome of this post-processing is a probabilistic state vector described by confidence intervals.

Since the solution of the mathematical model of the systems depends critically on the number of measurements being equal to the number of state variables, it is common practice to introduce a degree of redundancy into the specification of a telemetry system. This redundancy means that the failure of a single meter does not prevent the computations from proceeding. The optimisation process (Hartley and Bargiela 1995), which results in the state estimate of the network, also enables the detection and identification of measurement errors which otherwise might have adversely affected the system controls.

THE DISTRIBUTED PROBABILISTIC STATE ESTIMATION ALGORITHM

The first task of the algorithm is to divide the network into subnetworks to be later distributed on a transputer platform. So, the overdetermined set of M equations (1) are partitioned into s subnetworks:

$$\begin{bmatrix} \boldsymbol{g_1}(\boldsymbol{x}) \\ \boldsymbol{g_2}(\boldsymbol{x}) \\ \dots \\ \boldsymbol{g_s}(\boldsymbol{x}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{z_1} \\ \boldsymbol{z_2} \\ \dots \\ \boldsymbol{z_s} \end{bmatrix} + \begin{bmatrix} \omega_1 \\ \omega_2 \\ \dots \\ \omega_s \end{bmatrix} \qquad (9)$$

With utilisation of the Newton-Raphson method, with results similar to that of equations (3) to (7), the linearised form of this equation is:

$$\begin{bmatrix} \Delta\boldsymbol{z_1} \\ \Delta\boldsymbol{z_2} \\ \dots \\ \Delta\boldsymbol{z_s} \end{bmatrix} = J \cdot \Delta\boldsymbol{z} \qquad (10)$$

where

$$\begin{bmatrix} \frac{\partial \boldsymbol{g_1}}{\partial \boldsymbol{x}} & \frac{\partial \boldsymbol{g_2}}{\partial \boldsymbol{x}} & \dots & \frac{\partial \boldsymbol{g_s}}{\partial \boldsymbol{x}} \end{bmatrix}^T = J \qquad (11)$$

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

The resulting equation to be solved in order to determine the coordinated state estimate of these *s* partitions is the matrix

$$\begin{bmatrix} \Delta\boldsymbol{z}^p \\ \boldsymbol{Y} \end{bmatrix} = \begin{bmatrix} J' & -C_{\alpha\beta} \\ C_{\alpha\beta} & K \end{bmatrix} \begin{bmatrix} \Delta\boldsymbol{x} \\ \Delta\boldsymbol{y} \end{bmatrix} \qquad (12)$$

where $\Delta\boldsymbol{z}^p = \begin{bmatrix} \Delta\boldsymbol{z}^p{}_{\boldsymbol{1}} & \Delta\boldsymbol{z}^p{}_{\boldsymbol{2}} & \dots & \Delta\boldsymbol{z}^p{}_{\boldsymbol{s}} \end{bmatrix}^{\boldsymbol{T}}$ denotes the right-hand side of equation (10) being changed to take into account the compensating flows from the removed pipes,

$\boldsymbol{Y}$ is an augmented vector of zeroes, for the relevant pressure balances, and values of net mass imports/exports, for each subnetwork,

$J' = diag \begin{bmatrix} \frac{\partial \boldsymbol{g_1}}{\partial \boldsymbol{x_1}} & \frac{\partial \boldsymbol{g_2}}{\partial \boldsymbol{x_2}} & \dots & \frac{\partial \boldsymbol{g_s}}{\partial \boldsymbol{x_s}} \end{bmatrix}$ maps directly onto the Jacobian matrix *J* except for the elements corresponding to the removed pipes,

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

K is a coordinating matrix,

$\Delta\boldsymbol{x} = \begin{bmatrix} \Delta\boldsymbol{x_1} & \Delta\boldsymbol{x_2} & \dots & \Delta\boldsymbol{x_s} \end{bmatrix}^{\boldsymbol{T}}$ depicts the coordinated results of the subnetwork solutions, and

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

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

$$\boldsymbol{x}_{(\boldsymbol{k}+\boldsymbol{1})} = \boldsymbol{x}_{(\boldsymbol{k})} + \Delta\boldsymbol{x} \qquad (13)$$

Once the point state estimate has been derived, the algorithm then determines the associated bounds (Bargiela and Hainsworth 1988b; Hainsworth 1988). The upper and lower bounds $\boldsymbol{x^u}$ and $\boldsymbol{x}^l$ are found with utilisation of the sensitivity matrix $(J^TJ)^{-1}J^T$. 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 $\boldsymbol{x_i}$ to the *j*-th element of the measurement vector $\boldsymbol{z_j}$. The two vectors of upper and lower bounds are calculated one element at a time using the equations:

$$\boldsymbol{x^u} = \boldsymbol{x} + max[(J^TJ)^{-1}(J^T) \cdot \Delta\boldsymbol{z},\boldsymbol{0}] \qquad (14)$$

$$\boldsymbol{x^l} = \boldsymbol{x} + min[(J^TJ)^{-1}(J^T) \cdot \Delta\boldsymbol{z},\boldsymbol{0}] \qquad (15)$$

where $\Delta\boldsymbol{z}$ 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 $\boldsymbol{x}$ has been determined. The set is described as approximate because some of the vectors $\Delta\boldsymbol{z}$ may be inconsistent, and therefore redundant. The intervals, $\boldsymbol{x^l}$ and $\boldsymbol{x^u}$, 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 (Hainsworth 1988).

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

1. Read-in the system data consisting partially of an over-determined set of measurements (equation (9)).

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 (equation (12)).

3. Calculate each of the uncoordinated subsystem solutions by least absolute values regression to the linearised set of equations. Implement an incremental improvement to the initial solution, using the iterative Newton-Raphson procedure (equation (13)).

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 (equations (14) and (15)).

RESULTS

The algorithm has been tested on a 130-node network. 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 0.02 mAq.

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 0.02 mAq), and the other five meters are flow meters (each with an accuracy of $0.005 \mathrm{m}^3 \mathrm{s}^{-1}$). Table 1 shows the increased accuracy of the state estimate due to the inclusion of extra head and flow meters, as would be expected.

Table 1. Pressure bounds at selected network nodes

P denotes a pressure measurement reading at the node, with an accuracy of 0.02 mAq

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

Node	Example 1	Example 2	Node	Example 1	Example 2
6	5.743	0.816 F	53	5.572	0.690 F
7	3.983	0.630 F	57	1.979	0.020 P
9	5.751	0.824 F	63	4.103	0.317 F
19	12.457	1.478 F	65	0.020 P	0.020 P
21	11.583	0.917 F	77	4.298	0.020 P
24	5.048	0.020 P	80	1.108	0.066 F
31	7.505	0.384 F	82	1.134	0.027 F
33	10.038	0.020 P	84	2.482	1.497 F
34	4.947	0.121 F	86	3.894	2.234 F
38	4.936	0.132 F	89	6.335	0.020 P
39	5.497	0.020 PF	95	1.139	0.020 P
45	5.537	0.054 F	97	6.090	1.335 F
47	5.349	0.171 F	115	5.356	0.020 P
51	5.438	0.547 F	130	0.020 P	0.020 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 2.

Table 2. Convergence times of the algorithm when applied to four different networks, each partitioned into two, three and four subnetworks.

Number of nodes	Number of cutlines	Total subsystems solution time (s)	Total solution time (s)
65×2	4	10.87	44.30
$43 \times 2, 44$	17	5.46	38.98
$32 \times 2,$ 33×2	24	5.02	38.99
65×2	3	8.78	41.93
$43 \times 2, 44$	17	5.46	38.00
$32 \times 2,$ 33×2	23	3.65	36.64
65×2	4	8.49	40.56
$43 \times 2, 44$	15	3.99	37.74
$32 \times 2,$ 33×2	24	9.08	41.52
65×2	3	5.68	35.36
$43 \times 2, 44$	16	4.41	33.62
$32 \times 2,$ 33×2	21	3.41	33.11

These results show that, in general, the solution time of the subsystems decreases as the number of partitions increases. This is, however, dependent on the way in which the networks are partitioned. If the partitioning causes one of the subnetworks to include a number of conflicting measurements, the subsystems solution time will naturally increase (as in the case of the third network when divided into four - see Table 2).

The total solution time is an amalgamation of the total subsystems solution time, the time taken to coordinate these subsystem solutions and the time taken to evaluate the related bounds. The coordination time is dependent on the number of cutlines, while the time taken to evaluate the bounds is independent of the partitioning as it is implemented on the network as a whole. Consequently, the total solution time of the distributed probabilistic state estimation algorithm will normally decrease as the number of partitions increases and the number of cutlines decreases.

CONCLUSIONS

The use of the probabilistic state estimator introduces quantification of the inaccuracy of calculated system states caused by input data uncertainty. The inclusion of the distributed diakoptical state estimator results in a decrease in the amount of time required for the solution of the calculation of these states' bounds. The outcome, as a whole, is the development of a more efficient and more informative state estimator to be implemented on a distributed hardware architecture.

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 has been discussed how the inclusion of these effects can give the system operator a fuller picture of the state of the distribution system.

This probabilistic state estimation procedure has been successfully developed and applied to a test water distribution network model (Hartley 1996). The sensitivity matrix method is suitable for a number of applications, such as any on-line decision support procedure, which requires a confidence limit analysis procedure that can produce uncertainty bounds in real-time.

REFERENCES

BARGIELA, A. “Nonlinear Network Tearing Algorithm for Transputer System Implementation”, *Proc. TAPA ‘92*, pp 19-24, Nov. 1992.

BARGIELA, A., HAINSWORTH, G.D. “Telemetry System Design and On-Line Decision Support with ‘TCLAS’ Software”, *Computer Applications in Water Supply*, Vol.2, pp 43-58, 1988a.

BARGIELA, A., HAINSWORTH, G.D. “Pressure and Flow Uncertainty in Water systems”, *Jnl. of Water Resources, Planning and Management,* Vol. 2, pp 43-58, 1988b.

BRDYS, M.A., ABDULLAH, N., ROBERTS, P.D. “Augmented Model-Based Double Iterative Loop Techniques for Hierarchical Control of Complex Industrial Processes”, *Int. J. Control,* Vol. 52, No. 3, pp 549-570, 1990.

HAINSWORTH, G.D. ‘MEASUREMENT UNCERTAINTY IN WATER DISTRIBUTION TELEMETRY SYSTEMS’, PhD thesis, Nottingham Polytechnic, 1988.

HARTLEY, J.K. ‘PARALLEL ALGORITHMS FOR FUZZY DATA PROCESSING WITH APPLICATION TO WATER SYSTEMS’, PhD thesis, The Nottingham Trent University, 1996

HARTLEY, J.K., BARGIELA, A. “Parallel Simulation of Large-Scale Water Distribution Systems”, *Proc. of ESM ‘95*, pp 723-727, June 1995.

HOSSEINZAMAN, A., BARGIELA, A. “Parallel Simulation of Nonlinear Networks using Diakoptics”, *Proc. PACTA ‘92*, Sept. 1992.

EL-KEIB, A.A., SINGH, H. “Fast Linear Programming State Estimation Using the Dual Formulation”, *Transactions on Power Systems*, Vol. 7, No. 2, pp 620-628, May 1992.

IRVING, M.R., STERLING, M.J.H. “Optimal Network Tearing Using Simulated Annealing”, *IEE Proc. C (Generation, Transmission and Distribution)*, Vol. 137, Iss.1, pp 69-72, Jan. 1990.

LIN, J., LI, L.A., WAN, B.W., ROBERTS, P.D. “Triple Iterative Loop Technique for Optimizing control of Large-Scale Steady-State Systems”, *Int. J. Systems Sci.*, Vol. 20, No. 11, pp 2107-2124, 1989.

RUTOWSKI, J. “An Efficient Algorithm for Branch Tearing Large Scale Networks”, *Circuit Theory and Design ‘87*, pp 693-698, Sept. 1987.