NEURAL SIMULATION OF WATER SYSTEMS FOR EFFICIENT STATE ESTIMATION

B. Gabrys and A. 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
E-mail: andre@doc.ntu.ac.uk, bcz@doc.ntu.ac.uk

ABSTRACT

This paper presents a neural network based technique for the solution of a water system state estimation problem. The technique combines a neural linear equations solver with a Newton-Raphson iterations to obtain a solution to an overdetermined set of nonlinear equations.

The algorithm has been applied to a realistic 34-node water network. By changing the values of neural network parameters both the least squares (LS) and least absolute values (LAV) estimates have been obtained and assessed with respect to their sensitivity to measurement errors.

INTRODUCTION

Efficient control of a complex water distribution system requires accurate information about its current operating state. At present in the water industry, modern telemetry hardware systems are being installed to meet these needs. Unfortunately, due to financial constraints, it is not practical to measure all variables of interest. Therefore, the information supplied by the telemetry system must be supplemented by the least accurate predictions of consumptions at the nodes in the network. These predictions are frequently referred to as pseudomeasurements. Measurements and pseudomeasurements are used to calculate flows and pressures in the distribution network through the use of a state estimators which provide a means of reconciling the discrepancies between the mathematical model of the system and the input data (Sterling and Bargiela 1984; Hainsworth 1988). Over the last decade state estimators gradually became the key utility for the implementation of monitoring and control of large scale public utility systems such as water, gas or electric power distribution systems.

With the increasing complexity of modern water distribution systems there is a need for efficient state estimators which will form a basis for the implementation of real time control of these systems. Among the potential algorithms and techniques for state estimation neural network based estimators are of great interest because of their massively parallel nature and consequent computational efficiency. While the full potential of neural networks for mathematical optimization can only be realised with appropriate computing hardware, their performance has been assessed through the simulation studies.

Neuron-like architectures were simulated using MATLAB and SIMULINK programs. The networks have been configured to produce state estimates of the water system according to the least squares and the least absolute values optimality criterias. The results indicate that when implemented in VLSI technology (Cichocki and Unbehauen 1992a, 1992b), the computation time of state estimation task will not be influenced by the size of the problem (assuming the match of the network and the problem size) and the final solution will be found within the time of order of a hundred nanoseconds.

WATER SYSTEM MODEL AND ESTIMATION METHOD

The process of state estimation requires a mathematical model of a water distribution network. The nonlinear head-flow functions describing network elements are used to express mass-balances in each node of the physical system, and to represent the specific measurements that are being taken. This can be expressed as follows:

\[ z = g(x) + \omega \]  

where:
- \( z \) - measurement vector
- \( g(x) \) - nonlinear functions describing system
- \( \omega \) - vector of measurement inconsistency

The state estimation can be expressed as a problem of minimization of discrepancies between the actual measurements and the values calculated from the mathematical model. Using the least squares criterion the state estimation problem can be expressed as:

\[ \min_x E_2(x) = \frac{1}{2} (z - g(x))^T W (z - g(x)) \]  

Similarly, using the least absolute values criterion the state estimation is expressed as:

\[ \min_x E_1(x) = \sum_i w_i |z_i - g(x)_i| \]  

where:
- \( w \in \mathbb{R}^m \) - measurement weight vector
- \( W = \text{diag} [w_1, w_2, \ldots, w_m] \) - measurement weight matrix

While the least squares method (Cichocki and Unbehauen 1992a; Golub 1986; Gill at al. 1981) is optimal for a Gaussian distribution of the noise, the assumption that the set of measurements or observations has a Gaussian error distribution is frequently unrealistic due to various sources of errors such as instrument errors, modelling errors, sampling errors and human errors. In order to reduce the influence of large errors, the least absolute values criterion (3) can usefully be adopted (Sterling and Bargiela 1984; Cichocki and Unbehauen 1992b).

The proposed solution of the state estimation problem (2) or (3) is based on the Newton method. Expanding \( g(x) \) by an initial guess of the state vector \( x^{(0)} \), using a first-order Taylor series and
defining \( z^{(0)} = g(x^{(0)}) \), we obtain
\[
\begin{align*}
\Delta z & = z^{(0)} + \Delta z \\
g(x) & = g(x^{(0)}) + J(x^{(0)}) \Delta x
\end{align*}
\] (4)

After this linearisation we obtain the following set of equations:
\[
J(x^{(k)}) \Delta x = z - g(x^{(k)})
\] (6)

where:
\[
J(x^{(k)}) \in \mathbb{R}^{m \times n} - \text{Jacobian matrix evaluated at } x^{(k)}
\]
k = 0, 1, ... - step of the estimation process

Equations (2) and (3) can be therefore expressed as
\[
\min_{\Delta x} E_2(\Delta x) = \frac{1}{2} \left[ \Delta z - J(x^{(k)}) \Delta x \right]^T W \left[ \Delta z - J(x^{(k)}) \Delta x \right]
\] (7)

and
\[
\min_{\Delta x} E_1(\Delta x) = w^T [\Delta z - J(x^{(k)}) \Delta x]
\] (8)

These overdetermined sets of linear equations form the basis for the construction of a neural network which is presented in the following section.

Since the measurement equations (1) are nonlinear, the solution to (2) or (3) is an iterative process with the consecutive state estimates calculated by under-relaxation of the linear solution
\[
x^{(k+1)} = x^{(k)} + \gamma \Delta x^{(k)}, \quad k=0,1,....
\] (9)

If all elements of \( \Delta x \) in k-th iteration are lower or equal to a predefined convergence accuracy, the iteration procedure stops. Otherwise, a new correction vector is calculated using equation (6) with \( x^{(k+1)} \) instead of \( x^{(k)} \) and suitable neural network.

**NEURAL NETWORKS SOLVING SETS OF LINEAR EQUATIONS**

The minimisation problems described by (7) and (8) can be generalised as follows:
\[
\min_{\Delta x} E(\Delta x) = \sum_{i=1}^{m} \sigma_i [r_i(\Delta x)]
\] (10)

where:
\[
E \text{ is a general cost (energy) function} \\
A = J(x^{(k)}), \quad b = z - g(x^{(k)})
\]
\[
r_i(\Delta x) = \sigma_i^T \Delta x - b_i \quad \text{is the vector of residuals} \\
\sigma_i [r_i] \quad \text{represents a suitably chosen convex functions.}
\]

In a special case when \( \sigma_i (r_i) = \frac{r_i^2}{2} \) we obtain the standard least-squares criterion (7) and for \( \sigma_i (r_i) = |r_i| \) we have the least absolute values criterion (8).

The minimisation of the energy function described by (10) by standard gradient descent methods leads to the following system of nonlinear differential equations:
\[
\frac{d \Delta x}{dt} = -\mu_j \sum_{i=1}^{m} a_{ij} \left( f_i \left( \sum_{k=1}^{n} a_{ik} \Delta x_k - b_i \right) \right)
\] (11)

or in compact matrix form
\[
\frac{d \Delta x}{dt} = -\mu A^T f(\Delta x)
\] (12)

where:
\[
\mu_j > 0 \quad \text{is the learning parameter;}
\]
\[
f_i (r_i (\Delta x)) = \frac{\partial \sigma_i (r_i)}{\partial r_i} \quad \text{- activation function}
\]

a) for \( \sigma_i (r_i) = \frac{r_i^2}{2} \) we have \( f_i (r_i (\Delta x)) = r_i (\Delta x) \) - linear activation function

b) for \( \sigma_i (r_i) = |r_i| \) we have \( f_i (r_i (\Delta x)) = sign \{ r_i (\Delta x) \} \) - signum activation function

To cater for both least squares and least absolute values criteria the logistic function can be defined as follows:
\[
\sigma_L (r_i) = \frac{\beta}{\alpha} \ln \left( \cosh \left( \alpha r_i \right) \right)
\] (13)

where \( \alpha > 0, \beta > 0 \) are the problem dependant parameters and
\[
f_i (r_i) = \frac{\partial \sigma_L (r_i)}{\partial r_i} = \frac{\beta}{\alpha} \ln \left( \cosh \left( \alpha r_i \right) \right) = \beta \tanh \left( \alpha r_i \right)
\] (14)

is the sigmoid activation function.

Choosing \( \beta \) large with \( \alpha = 1/\beta \) the sigmoid activation function approximates \( f_i (r_i (\Delta x)) \) over a wide range, so the network solves the system of linear equations in the least squares sense. On the other hand, taking \( \beta = 1 \) and \( \alpha \) large we obtain an approximation of \( f_i (r_i (\Delta x)) = sign \{ r_i (\Delta x) \} \), so the network solves the least absolute values problem.

The system of differential equations (12) has been implemented using SIMULINK, by the artificial neural network (ANN) shown in Figure 1.

![Figure 1: ANN for solving a system of linear equations (6) based on the system of differential equations (12) with optional activation functions (fr) (implementation in MATLAB and SIMULINK)](image-url)
COMPUTER SIMULATION RESULTS

The performance of the proposed methods for water-system state estimation was tested on the realistic 34-node network (42 state variables) depicted at the Figure 2.

A complete definition of network parameters are contained in (Sterling and Bargiela 1984). In order to achieve sufficient measurements redundancy (defined as a ratio of the number of measurements and pseudomeasurements to the number of state variables), the set of the mass balance equations was augmented by a number of several flow and pressure measurements.

Two sets of measurements were processed having redundancy ratios 1.74 and 1.4.

The effect of ‘bad data’ measurements was simulated by introduction of systematic gross errors in head and flow measurements.

The specification of these errors are given in the following examples.

**Example 1:**

Introduced gross errors
head in node 22 = 42.59 [m Aq] (exact value = 46.59 [m Aq])
load in node 8 = -0.025 [m³/s] (exact value = -0.075 [m³/s])

**Example 2:**

Introduced gross errors:
head in node 22 = 42.59 [m Aq] (exact value = 46.59 [m Aq])
head in node 29 = 35.70 [m Aq] (exact value = 31.70 [m Aq])
head in node 30 = 48.58 [m Aq] (exact value = 43.58 [m Aq])
load in node 8 = -0.025 [m³/s] (exact value = -0.075 [m³/s])

Table 1 and Table 2 show the state estimates calculated for redundancy ratios 1.74 and 1.4 respectively. The neural network (Figure 1) has been implemented in SIMULINK. The state vector shown in column 2 of Table 1 and Table 2 is the state vector obtained by exact network simulation and is referred to as a vector of reference values. State estimates of LS and LAV methods calculated for data not including gross errors are presented in column 3 and 4 of Table 1 and Table 2 respectively. State estimates calculated for data including gross errors are presented in columns 5, 6, 7 and 8.

**LAV (Least Absolute Values) method.**

Table 1 (columns 6 and 8) shows the results of examples in which higher measurement redundancy has been used. Table 2 (columns 6 and 8) shows the corresponding results for lower measurement redundancy. Comparison of these results indicates that a smaller number of equations (measurements) was sufficient for accurate estimation with a specific pattern of measurements considered. However, an increased number of measurements contributes mainly to an improved reliability of the estimation and ensures the rejection of a larger spectrum of errors. In conclusion, the LAV problem solution is median solution and passes through at least n (n - number of state variables) of the m data points (measurements). The feature of producing interpolatory fits that approximate closely most of the data while neglecting gross errors is an extremely useful property of the LAV criterion. Provided sufficient basic measurements are available, the LAV estimator...
can then act as filter for incoming data

**LS (Least Squares) method.**

The ordinary LS problem solution is the mean solution since it tries to satisfy all the equations in the set, but usually this solution will not solve exactly any of these equations. The results shown in columns 5 and 7 of Table 1 and Table 2 are very good example of the influence of gross errors on a LS state estimation. A measurement containing gross error has the biggest effect on the state estimation process, as discussed in this paper, will be accomplished in a time of order of hundred microseconds. Consequently the state estimation process, as the main cause of using of state estimation methods is insufficient number of measurements.

Figures 3 and 4 illustrate the convergence of the estimation process (variables x1 and x8 respectively) for the LS and LAV estimators.

All simulations have been carried out on Sun Workstation using SIMULINK (Dynamic System Simulation Software) and MATLAB (High-Performance Numeric Computation and Visualization Software) programs. Various integration algorithms have been used for many different values of the parameters $\gamma$, $\alpha$, $\beta$ and $\mu$. The results presented in Tables have been obtained for $\gamma = 0.6$, $\mu = 10^6$. Gear integration algorithm. Parameters $\alpha$ and $\beta$ have been set as follows: $\alpha = 0.1$, $\beta = 10$ for LS and $\alpha = 500$, $\beta = 1$ for LAV estimators.

**CONCLUSIONS**

It has been found, through the simulation study, that a neural network based state estimator provides an efficient means of water system state estimation. While the LS estimates have shown to be strongly affected by any change in the measurement vector, the LAV estimates proved to be resistant to large changes in the data. This is a very useful property when the known data in the measurement vector are contaminated with occasional gross errors.

The main restriction of a VLSI implementation of neural networks is the number of connections between the processing units on a chip. It is envisaged that the future use of optical cross-connecting will enable the implementation of arbitrary large ANN. Consequently the state estimation process, as discussed in this paper, will be accomplished in a time of order of hundred microseconds.

**REFERENCES**


---

![Table 1: 34-node-system state estimates (73 equations; redundancy ratio=1.74)](image)

1-34: nodal heads (m Aq) at nodes 1-34; 35-42: fixed-head nodes in/out flows (m /s) at nodes 27-34

LS - least squares method; LAV - least absolute values method
<table>
<thead>
<tr>
<th>State Variable</th>
<th>Exact value</th>
<th>LS</th>
<th>LAV</th>
<th>LS (Ex.1)</th>
<th>LAV (Ex.1)</th>
<th>LS (Ex.2)</th>
<th>LAV (Ex.2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>32.638</td>
<td>32.617</td>
<td>32.661</td>
<td>32.606</td>
<td>32.662</td>
<td>33.697</td>
<td>32.673</td>
</tr>
<tr>
<td>2</td>
<td>43.749</td>
<td>43.750</td>
<td>43.745</td>
<td>43.599</td>
<td>46.810</td>
<td>46.616</td>
<td>46.616</td>
</tr>
<tr>
<td>3</td>
<td>46.041</td>
<td>46.076</td>
<td>46.052</td>
<td>46.023</td>
<td>46.027</td>
<td>46.027</td>
<td>46.027</td>
</tr>
<tr>
<td>4</td>
<td>46.618</td>
<td>46.650</td>
<td>46.637</td>
<td>46.634</td>
<td>46.467</td>
<td>46.616</td>
<td>46.616</td>
</tr>
<tr>
<td>5</td>
<td>43.265</td>
<td>43.254</td>
<td>43.260</td>
<td>43.263</td>
<td>43.204</td>
<td>43.269</td>
<td>43.269</td>
</tr>
<tr>
<td>6</td>
<td>43.024</td>
<td>42.971</td>
<td>42.975</td>
<td>42.977</td>
<td>43.150</td>
<td>43.001</td>
<td>43.150</td>
</tr>
<tr>
<td>7</td>
<td>42.402</td>
<td>42.362</td>
<td>42.380</td>
<td>42.391</td>
<td>43.929</td>
<td>42.390</td>
<td>42.390</td>
</tr>
<tr>
<td>8</td>
<td>42.130</td>
<td>42.094</td>
<td>42.106</td>
<td>42.130</td>
<td>43.924</td>
<td>42.130</td>
<td>42.130</td>
</tr>
<tr>
<td>9</td>
<td>43.798</td>
<td>43.788</td>
<td>43.864</td>
<td>43.779</td>
<td>44.683</td>
<td>43.815</td>
<td>44.683</td>
</tr>
<tr>
<td>10</td>
<td>47.950</td>
<td>47.946</td>
<td>47.729</td>
<td>47.946</td>
<td>48.048</td>
<td>47.947</td>
<td>48.048</td>
</tr>
<tr>
<td>11</td>
<td>44.664</td>
<td>44.681</td>
<td>44.670</td>
<td>44.670</td>
<td>45.010</td>
<td>44.670</td>
<td>45.010</td>
</tr>
<tr>
<td>12</td>
<td>44.004</td>
<td>44.017</td>
<td>44.058</td>
<td>44.067</td>
<td>44.658</td>
<td>44.039</td>
<td>44.658</td>
</tr>
<tr>
<td>13</td>
<td>49.274</td>
<td>49.306</td>
<td>49.303</td>
<td>49.306</td>
<td>49.703</td>
<td>49.314</td>
<td>49.703</td>
</tr>
<tr>
<td>14</td>
<td>49.099</td>
<td>49.095</td>
<td>49.052</td>
<td>49.325</td>
<td>49.208</td>
<td>49.097</td>
<td>49.208</td>
</tr>
<tr>
<td>15</td>
<td>49.057</td>
<td>49.054</td>
<td>49.059</td>
<td>48.893</td>
<td>49.050</td>
<td>49.050</td>
<td>49.050</td>
</tr>
<tr>
<td>16</td>
<td>49.298</td>
<td>49.321</td>
<td>49.324</td>
<td>49.327</td>
<td>49.733</td>
<td>49.366</td>
<td>49.733</td>
</tr>
<tr>
<td>17</td>
<td>47.970</td>
<td>47.967</td>
<td>47.972</td>
<td>47.969</td>
<td>48.148</td>
<td>47.970</td>
<td>48.148</td>
</tr>
<tr>
<td>18</td>
<td>49.338</td>
<td>49.348</td>
<td>49.346</td>
<td>49.348</td>
<td>49.766</td>
<td>49.358</td>
<td>49.766</td>
</tr>
<tr>
<td>19</td>
<td>49.029</td>
<td>49.049</td>
<td>49.043</td>
<td>48.591</td>
<td>49.045</td>
<td>49.082</td>
<td>49.045</td>
</tr>
<tr>
<td>20</td>
<td>46.618</td>
<td>46.655</td>
<td>46.637</td>
<td>46.634</td>
<td>46.489</td>
<td>46.615</td>
<td>46.489</td>
</tr>
<tr>
<td>21</td>
<td>45.623</td>
<td>45.651</td>
<td>45.639</td>
<td>45.620</td>
<td>45.803</td>
<td>45.635</td>
<td>45.803</td>
</tr>
<tr>
<td>22</td>
<td>46.588</td>
<td>46.628</td>
<td>46.607</td>
<td>46.632</td>
<td>46.609</td>
<td>46.631</td>
<td>46.609</td>
</tr>
<tr>
<td>24</td>
<td>43.249</td>
<td>43.224</td>
<td>43.226</td>
<td>43.115</td>
<td>44.247</td>
<td>43.252</td>
<td>44.247</td>
</tr>
<tr>
<td>25</td>
<td>42.532</td>
<td>42.487</td>
<td>42.496</td>
<td>42.933</td>
<td>44.005</td>
<td>42.520</td>
<td>44.005</td>
</tr>
<tr>
<td>26</td>
<td>32.086</td>
<td>32.102</td>
<td>32.104</td>
<td>32.106</td>
<td>33.230</td>
<td>32.115</td>
<td>33.230</td>
</tr>
<tr>
<td>29</td>
<td>31.692</td>
<td>31.702</td>
<td>31.704</td>
<td>31.706</td>
<td>32.793</td>
<td>31.718</td>
<td>32.793</td>
</tr>
<tr>
<td>30</td>
<td>43.582</td>
<td>43.604</td>
<td>43.603</td>
<td>43.590</td>
<td>44.392</td>
<td>43.624</td>
<td>44.392</td>
</tr>
<tr>
<td>31</td>
<td>44.188</td>
<td>44.201</td>
<td>44.198</td>
<td>43.526</td>
<td>44.205</td>
<td>44.201</td>
<td>43.526</td>
</tr>
<tr>
<td>32</td>
<td>45.710</td>
<td>45.747</td>
<td>45.724</td>
<td>45.929</td>
<td>45.720</td>
<td>45.720</td>
<td>45.720</td>
</tr>
<tr>
<td>35</td>
<td>0.0723</td>
<td>0.0722</td>
<td>0.0723</td>
<td>0.0719</td>
<td>0.0716</td>
<td>0.0723</td>
<td>0.0716</td>
</tr>
<tr>
<td>36</td>
<td>0.0927</td>
<td>0.0925</td>
<td>0.0886</td>
<td>0.0926</td>
<td>0.0874</td>
<td>0.0927</td>
<td>0.0874</td>
</tr>
<tr>
<td>37</td>
<td>-0.0229</td>
<td>-0.0229</td>
<td>-0.0229</td>
<td>-0.0229</td>
<td>-0.0170</td>
<td>-0.0229</td>
<td>-0.0170</td>
</tr>
<tr>
<td>38</td>
<td>0.0519</td>
<td>0.0523</td>
<td>0.0523</td>
<td>0.0554</td>
<td>0.0524</td>
<td>0.0522</td>
<td>0.0524</td>
</tr>
<tr>
<td>39</td>
<td>-0.0391</td>
<td>-0.0391</td>
<td>-0.0391</td>
<td>-0.0391</td>
<td>-0.0427</td>
<td>-0.0393</td>
<td>-0.0427</td>
</tr>
<tr>
<td>40</td>
<td>0.0254</td>
<td>0.0261</td>
<td>0.0257</td>
<td>0.0249</td>
<td>0.0254</td>
<td>0.0254</td>
<td>0.0254</td>
</tr>
<tr>
<td>41</td>
<td>0.0614</td>
<td>0.0614</td>
<td>0.0614</td>
<td>0.0619</td>
<td>0.0633</td>
<td>0.0614</td>
<td>0.0619</td>
</tr>
<tr>
<td>42</td>
<td>0.1061</td>
<td>0.1063</td>
<td>0.1063</td>
<td>0.1073</td>
<td>0.1063</td>
<td>0.1062</td>
<td>0.1063</td>
</tr>
</tbody>
</table>

Table 2: 34-node-system state estimates (59 equations; redundancy ratio 1.4)

1-34: nodal heads (m Aq) at nodes 1-34;
35-42: fixed-head nodes in/out flows (m/s) at nodes 27-34
LS - least squares method; LAV - least absolute values method