On-line monitoring for operational control of water distribution networks

Powell, R.S.

ON-LINE MONITORING FOR OPERATIONAL CONTROL OF WATER DISTRIBUTION NETWORKS

by

R. S. Powell, B.Eng., C.Eng., M.I.E.E., M.Inst.M.C.

A thesis submitted to the Faculty of Science
at
the University of Durham
for the degree of Doctor of Philosophy

May, 1992

This work is dedicated to my mother and to the happy memory of my father.

Acknowledgements

I am indebted to a number of people for their help in producing this work. Foremost among these is Professor Sterling, my former supervisor, whose guidance and enthusiasm was an inspiration. It was Professor Sterling who initially started work in this area in 1970 and it has been my pleasure to work with him in establishing a thriving research group. I also wish to thank Dr. Jim Bumby for kindly standing in as my supervisor after Professor Sterling and I parted company with Durham a year and a half ago.

I am also indebted to the present research group, comprising Dr. Gouping Yu, Dr. Chengchao Xu and Joe Hartley who, with the minimum of help and guidance from me, have succeeded in producing some outstanding theoretical work of immense practical importance in the field of control of water distribution systems. They have thus helped to build the group into a recognised leader in its field and in so doing, supported me in my own endeavours towards this end.

I would like also to thank Jeremy Gann for his patience with my frequent computer operating system ineptitude. I would like to thank Malcolm Irving, Director of the Brunel Institute of Power Systems, and others of his research group for their help, advice and for their continued encouragement, and to thank Maurice Smith for giving me so many practical examples of the convolution theorem in the course of many illuminating discussions, for his friendship and help.

Conducting research as a registered part-time Ph.D. student is quite selfish since it imposes many pressures on family and friends and I would like to apologise to all for my rudeness in turning down many warm invitations of friendship. My family were, as always, totally supportive and understanding of the reduced contact during this period of study and to them all I owe a debt of gratitude and of course to Min, my wife, I will be eternally grateful for her patience during the many weekends and evenings spoilt by my writing this thesis.

To all who contemplate starting a part-time Ph.D. — Don't.

Abstract

This work concerns the concept of on-line monitoring and control for water distribution networks. The problem is simple to state. It is to produce a robust scheme that can continuously provide reliable information about the state of a water network in real-time and over extended periods with the minimum of operator interaction.

This thesis begins by proposing a relational database schema to store 'asset data' for a water distribution network and asserts that asset data should be used as a basis for network modelling. It presents a topology determination algorithm and a demand allocation algorithm so that a mathematical model can be maintained on-line, with operator intervention only necessary to record the change of state of non-telemetered plant items such as switch valves.

In order to provide a reliable on-line model of a distribution system, an investigation has been carried out into the methods available for modelling water networks and in particular, the inherent assumptions in these practices. As a result, new methods have been produced for network element combination and for demand allocation. These methods both support the database approach and enhance the robustness of the system by increasing the range of conditions for which the resulting model is applicable.

For operational control, a new technique for state estimation is proposed which combines the advantages of weighted least squares estimation with those of weighted least absolute values estimation. The proposed method is tolerant to transducer noise and to the presence of large measurement outliers. However, the method is not limited in its application to water networks and could be applied to a wide range of measurement processing problems. Lastly, a new topology based method for processing suspect data is proposed which can determine the likely causes using identifying templates.

Thus a new approach to water network monitoring is proposed via an overall framework into which the various tasks of on-line operational control can be integrated. The exercise has resulted in the production of a core software package which could realistically be used in a control room to facilitate reliable operational control of water distribution systems.

Table of Contents

Declaration and Copyright

The work contained in this thesis has not been previously submitted for consideration for a degree. It is entirely the author's own work.

List of Principal Symbols

Notes: Throughout this thesis the following conventions are used:

1. Scalars are represented by italic letters, e.g. 'Re';
2. Vectors are represented by bold face, lower case letters, e.g. '$\mathbf{z}$';
3. Matrices are represented by bold face, upper case letters, e.g. '$\mathbf{P}$';
4. Iteration counters are written as superscripts in parentheses. e.g. '$x^{(k)}$' refers to the value of x at the k^{th} iteration;
5. Estimated values have a 'hat', e.g. '$\hat{x}$'.

Symbols used in Chapter 2:

x_f — filtered value of an actual measurement x;

a — the filter coefficient;

p — the order of the auto regressive component;

q — the order of the moving average component;

d — the difference order;

Z_t — value of time series at time t;

A_t — value of residual white noise time series at time t;

B — the backshift operator;

$\phi_1 \ldots \phi_p$ — auto regressive weighting parameters;

$\theta_1 \ldots \theta_q$ — moving average weighting parameters;

ρ — the measurement redundancy;

m — the number of measurements;

n — the dimension of the state vector;

$\mathbf{\Sigma}$ — the gain matrix of the state estimator;

$\mathbf{J}$ — the Jacobian matrix;

$\mathbf{R}$ — the measurement covariance matrix;

$\mathbf{z}$ — vector of actual measurements;

S_i^2 — the i^{th} state estimate error covariance;

d_i — the distance of the i^{th} meter to the control centre;

N_k — the number of meters;

β — the upper limit on the variances of the estimation errors.

Additional symbols used in Chapter 4:

L — set of links;

L^{no} — set of link numbers;

L^{type} — type of each link in the set L;

V — set of valves;

$V^{l.no}$ — set of link numbers from the set of valves;

V^{status} — status of each valve in the set V;

CV — set of closed valves;

$CV^{l.no}$ — set of link numbers from the set of closed valves;

ML — set of links that are not in node clusters;

T_p — the potential topology;

T_a — the actual topology;

T_{a_r} — the reduced actual topology;

T_c — the topology of the node clusters;

Additional symbols used in Chapter 5:

q_{ij} — the flow from node i to node j;

h_{ij} — the head loss from node i to node j;

CHW — the Hazen Williams coefficient of the pipe;

D — the pipe diameter;

L — the pipe length;

g_i — the flow inbalance at node i;

d_i — the demand at node i;

u_i — the in-flow at node i;

N — the number of variable head nodes;

F — the number of fixed head nodes;

$\mathbf{x}$ — the state vector;

K — the hydraulic admittance of a pipe (in Hazen Williams terms);

f — the friction factor;

e — the internal roughness;

g — the acceleration due to gravity;

μ — the kinematic viscosity (of water);

v — the velocity of water flowing in the pipe;

ρ — the density (of water);

Re — the Reynold's number;

H — a constant head;

Q_{in} — the flow into a pipe;

Q_L — the distributed demand on a pipe;

ND — the number of consumers along a pipe.

Additional symbols used in Chapter 6:

ϵ — measurement noise vector;

r_i^W — i^{th} weighted residual;

r_i^N — i^{th} normalized residual;

$C(\cdot)$ — cost function;

$\mathbf{e}$ — vector of (unknown) measurement errors;

$\mathbf{S}$ — residual sensitivity matrix;

$\mathbf{P}$ — residual covariance matrix;

$\mathbf{z^0}$ — vector of unknown values equal to the measurements without any errors;

$\mathbf{g}(\cdot)$ — a nonlinear function relating state variables to measurements;

$\mathbf{W}$ — a diagonal weight matrix;

$\mathbf{u}$ — a vector of weights equal to the standard deviations of the measurements;

$\mathbf{H}$ — the weighted Jacobian matrix;

$\hat{\mathbf{z}}$ — vector of measurement estimates;

$\mathbf{\Delta s}$ — vector of weighted measurements;

$\mathbf{D}$ — diagonal weight control matrix.

CHAPTER 1

INTRODUCTION

1.1 Background

Urban water supply networks, as we know them today, were pioneered by the Romans who considered that the provision of a clean water supply was concomitant with the development of a truly civilised society. At the start of the first century A.D., a distribution system was completed in the city of Rome that supplied pressurised water to almost every part of the city [115]†. The importance of clean and reliable water supply has now become paramount. Although, in more recent years, nearly every home in the developed world is directly connected to a water distribution network, the provision of this supply is often taken for granted and the problems associated with the management of it are underestimated.

The consumption of water per capita has been gradually rising in the U.K. since the beginning of this century when it was at about 100 litres/head/day to the current level now of around 130 litres/head/day [5]. With the steady rise in the population, the increase in total demand (from around 4,000 Ml/day in 1900 to nearly 20,000 Ml/day in 1985) is putting a much greater strain on the infrastructure of the water industry.

† A fascinating early account of the system, its management, together with a supply and distribution analysis by Frontinus, Water Commissioner of Rome A.D.97.

Although controls and advice issued during recent droughts have undoubtedly eased the pressure on distribution systems, the projected trend into the foreseeable future indicates a continuing increase in demand for clean water supply [255].

Water distribution systems are now more complex, extensive and hence more difficult to run than ever before. Privatisation has put the spotlight on the water industry, resulting in an increase in public accountability, new government controls and higher customer expectations, the combination of which has lead to more stringent quality requirements and improved levels of service. Furthermore, energy costs have not only been rising but the method of purchasing electricity has become more complicated. Privatisation of the power industry has introduced the possibility of purchasing power in a variety of tariff structures from regional electricity suppliers (in theory, not necessarily the local supplier), from national generators, or directly from the pool, normally through a seller who provides hedging facilities.

Controlling the supply of water in an energy efficient manner whilst maintaining quality, constancy and pressure related goals in an ageing network with ever increasing demand is clearly a complex task. The main aim of the research outlined in this thesis is to simplify at least a part of that task through the provision of reliable, detailed and accurate on-line monitoring techniques.

In this chapter a brief introduction to some of the problems that face the water distribution operations engineers is given in order to provide a background to the overall aims of the work presented in later chapters. Furthermore, the chapter acts as a summary by setting out the structure of the thesis and lastly, by listing the main contributions made by it.

1.2 Pipe Networks

Many existing mains are quite old and hence present day water distribution networks are made up of a mixture of pipe varieties some of which are in very poor condition. As a pipe ages, the effective internal diameter either decreases due to incrustation, which can happen in areas where the water tends to be acidic and capable of carrying dissolved metal salts or it increases due to corrosion. An encrusting pipe causes a rise in upstream pressure and this can lead to excessive pressures elsewhere in the network whereas a corroding pipe will eventually become porous and weak leading to water contamination and loss. Bursts in water pipes can occur for a large number of reasons. Pipes are often laid close to the surface and therefore can be vulnerable to vibrations from road or rail traffic, to frost and to disturbances from earth moving machinery in use by the electrical, gas or telephone utilities. In addition to this, networks are large interconnected structures intolerant to relative movement of individual components. Some networks are in areas liable to subsidence and almost all will endure some land movement whether natural or arising from the different quality of pipe laying practices within a local area leading to differential settlement. In addition to the pipes themselves, networks contain large numbers of fittings such as valves, hydrants, tappings and joints all of which have to be carefully sealed to prevent the loss of water [229].

1.3 Leakage

It is common practice, when studying the usage of a public utility, to analyse the breakdown of the utility by component. When such an analysis is carried out for water consumption, one finds that between 10% and 60% of the total water supplied to the distribution network is left unaccounted for by any user category. A typical breakdown of consumption by component is given in the publications by the National Water Council [183, 184]. Although leakage is often quoted as a percentage, it is generally dealt with in units of $l/property/hour$ since this quantity relates more closely to the demand structure

in terms of domestic dwellings rather than pump or reservoir output. Until recently, the quantity was a somewhat arbitrary value derived as the remainder following the calculation of the estimated consumer demand subtracted from the reservoir and pump outflow (ie the total network supply). Clearly, for a given amount of water flowing into the network, if the consumer demand was over-estimated then the leakage rate would be proportionally lower. Nowadays more detailed studies are normally made with less room for errors of judgement, however, since actual consumer demand is very rarely known, most leakage calculations are still fairly approximate.

The term 'water unaccounted for' comprises two main components, 'wasted' water either lost from the distribution system through leakage or allowed to escape for no useful purpose and meter under-registration, ie measurement errors. In considering the quantity 'waste', it is useful to remember the method of 'waste metering' currently standard practice throughout the water industry [112]. Here, the water flow due to leakage is indistinguishable from that due to base load demand from malfunctioning cisterns or taps left running.

Water loss can occur anywhere in the distribution system including the communication pipes that lead from the supply main to the individual households. This type of waste is considered the user's responsibility and since water meters are not generally installed in U.K. properties it cannot be accurately measured. In general this can be detected in waste and district metering using step tests provided the level of leakage is large enough.

1.4 Unit Costs of Leakage

The justification for expenditure on leakage reduction is normally based on previous experiences of prudent operating practice than on a cost by cost breakdown. This is because an exhaustive analysis of the cost of leakage is nigh impossible. The true cost

can include any or all of the following depending on the size of the burst and how long it has been undetected, however, items vi) — viii) are more difficult to allocate a cost to and are therefore often excluded from the analysis:

i) Treatment and pumping costs of lost water;
ii) Man-power costs in detection;
iii) Network repair costs;
iv) Highway and property repair costs;
v) Capital expenditure acceleration due to artificially increased demand (ie requirement for reservoirs ahead of time and oversized pumps etc);
vi) Poorer public relations especially during hose pipe bans etc.;
vii) Water contamination caused by the ingress of detritus from outside the leaking pipe;
viii) Loss of supply pressure possibly infringing the 'levels of service' guide-lines.

Although the W.R.C.'s "Report 26" [228] does attempt to provide an analysis of leakage costs, it is not difficult to see that individual analyses will differ greatly from one system to the next depending, for instance, on how close the network is to operating at its maximum supply capacity. What is clear is that costs are substantial and are a considerable proportion of the total cost of network operations, an example of these costs are given in [257] and other incentives for leakage control are reported in [197, 226].

1.5 Leakage Detection and Control

Leakage detection methods in use at the present time use waste metering as a method to determine the net night flow into an area or drop tests which determine the reservoir draw down during the night-time periods. The results are compared to the expected night time demand for the area and an assessment is made as to whether this demand is abnormal or not. The burst pipe is normally located using 'step tests' which

cordon off smaller and smaller areas with flow meters until the actual area with the abnormally high demand has been found. This process can take several night-time periods. Once the locality has been found the actual position of the burst is normally determined using listening devices (often a stick, one end of which is pressed on the ground, the other being held against the ear) or with the aid of leak noise correlation devices or sometimes using tracer methods [113, 130]. Slightly more sophisticated methods have been used by Divenot, utilising measurement variance analysis [94] and transient analysis [95].

Section 1.2 highlighted the problems of an ageing distribution system. Of course, high pressures whether transient (water hammer) or sustained, exacerbate the problem of burst occurrence. Transient high pressures are sometimes caused by inexperienced operators closing large valves too quickly or they can occur when one-way valves close or due to other phenomena [250]. Sustained high pressures are often unavoidable at night as the demand for water (and hence the pressure drop from the supply points) decreases. Many schemes have been introduced to control pressures using modulating valves known as PRV's (pressure reducing valves) [24].

1.6 The Need for On-line Monitoring

Leakage control is not the only problem faced by the water supplier and it is certainly not the most serious from the operator's point of view. The operation of a water network on a day to day level is more concerned with the problem of providing customers with an uninterrupted supply of unpolluted water at a pressure above a set minimum. If all the pressures in a distribution system were known, then alarms could inform the operator whenever a pressure violation occurred. Furthermore, this information leads directly to knowledge of the flows, whereupon, if a quality problem occurred, the operator could trace the contamination back to its source. So it is clear that on-line monitoring, leading to a full knowledge of flows and pressures within a distribution

system is a desirable objective even if the operator is only really interested in violations of predetermined levels rather than the precise numerical values.

The principal control that the operator can apply to a network is via the pump schedule. A good operator will produce an energy efficient schedule that balances the security of the system against the operational cost. Since energy management implies control, the industry is increasing its expenditure on I.C.A. (instrumentation, control and automation) schemes [230, 231] and it is now highly relevant to consider more sophisticated monitoring and control systems and to exploit some of the recent methods from control engineering technology.

As the sophistication needed for control increases, more elaborate schemes are required. This has resulted in the installation of telemetry and S.C.A.D.A. (supervisory control and data acquisition) systems as part of the I.C.A. strategy for water distribution systems and has enabled the operators to control entire systems centrally, as reported in a number of publications, such as [25, 33, 36, 45, 77, 93, 123, 161, 163, 168, 200, 211, 253].

The availability of more advanced telemetry schemes has opened up new possibilities in network monitoring and control. Recently, several researchers have formulated the pressure control problem as an optimisation problem [49, 122, 147, 236, 266]. One of the benefits of this type of optimal pressure control is that it can prevent leakage as well as reduce the loss due to existing leaks. All these methods are based on computer models of a water distribution system and on measurements obtained from the system.

A number of monitoring schemes have already been proposed [38, 66, 70–74, 84–89, 118] however, an on-line monitoring system should be adaptable (that is, it should be possible update the parameters of the model as the parameters of the system change) and it should be able to detect and identify instrumentation errors. The systems proposed

in these earlier works do not fully provide these capabilities. Sterling and Bargiela [12, 235], however, have proposed the use of static state estimation for on-line monitoring of water distribution systems and also proposed bad data identification and leak detection methods.

1.7 Thesis organisation

This thesis can be considered to be made up of four main sections. Section 1, comprising Chapters 1 and 2, being introductory in nature, provides the background upon which the thesis is based. Chapter 2, in particular, reviews the suite of software known as 'WASMACS' (Water Systems Monitoring And Control Software) and introduces the current work in that context. The principal idea behind this entire thesis, that of adaptable, robust, on-line monitoring of a water distribution system, is presented in this chapter together with the structure of the proposed overall control system. The software produced in this work is also summarised in Chapter 2.

The next section, comprising Chapters 3 and 4, considers the design and exploitation of a relational database as a means of storing the network data in a readily adaptable form. Chapter 3 outlines a new relational database approach to the upkeep of asset records in a form suitable for on-line modelling; Chapter 4 introduces a new network topology determination method exploiting relational algebra and introduces a novel method for the automatic construction and upkeep of network models.

The 3rd section, comprising Chapter 5 only, is concerned with the accurate mathematical representation of water network flows. In this chapter, various flow models are analysed and novel modifications are introduced to allow reliable automatic combination of serial and parallel network elements without operator intervention. Furthermore, a new method for dealing with lumped consumer demand is presented that allows the records of the demand to reside along with the pipe information rather than be nodally based.

Finally, on-line monitoring is the theme of the 4^{th} section, comprising chapters 6 and 7, wherein the concepts of state estimation are examined and fault detection schemes are presented. Chapter 6 introduces a new robust method of state estimation for on-line monitoring of water distribution networks that is capable of detecting and identifying anomalies in the in-coming measurements and exploits all the functionallity and adaptability of the ideas presented in the foregoing chapters. Chapter 7 presents a new method for anomaly diagnostics which differentiates between measurement errors, leaks and other topological changes.

1.8 Main contributions of this work

This work has encompassed a range of disciplines. The main and principal achievement of the work is the proposal of an integrated scheme for water network on-line monitoring, in order to support other operational control functions. This thesis describes the components of the scheme and proposes several new solutions for the various problems of long term use of such an on-line monitoring system. In particular, the thesis offers:

1.) A schema for relational database design that allows a set of asset records to form the basis of a network model;
2.) A new method for topology determination from asset data;
3.) A method for automating the construction of a network model from asset data and demand data such that demands are automatically lumped at the correct nodes whenever topological changes take place;
4). A new method for converting between pipe resistance formulæ and for combining parallel and series pipes using the Colebrook-White equation;
5.) An analysis of distributed demands and the accuracy of lumped demand assumptions;

6.) A new, robust method for solving the state estimation problem;

7.) A new method for discriminating between bad data types;

8.) A method for diagnosing network faults from measurement errors.

In conclusion, this work represents a realistic step towards reliable on-line monitoring through the application of robust state estimation techniques, underpinned by a reliable flow model and a flexible database. Until now, no state estimation algorithm could be used on-line for any length of time in the water industry because of the inherent inflexibility of traditional methods of modelling topological network data and lack of robustness of the estimation algorithms. As an endorsement of the significance of this work, a major water company has recently signalled its intention to validate the methods outlined, in an on-line field trial.

A significant amount of software has been produced in this research project some of which is discussed in Chapter 2, the remainder in [191].

CHAPTER 2

WATER SYSTEMS MONITORING AND CONTROL SOFTWARE

2.1 Introduction

Computer modelling of water networks has been common practice for a number of years. Used principally in the design of new systems and modifications of existing ones, the analysis is also invariably helpful in discovering a wealth of information about an existing network from its actual topology to the whereabouts of the biggest demands on the system and the discovery of previously undetected leaks. The utility of network analysis packages is such that every water distribution company in this country uses one.

In this work, a move towards automating the on-line analysis of water networks is proposed using on-line state estimation. To understand the background to this work, it is necessary first to present an overall framework into which the proposed techniques can be placed. This is the aim of the present chapter.

2.2 Review of Previous Work

The application of computational techniques in modelling the behaviour of water networks either as a snap-shot or over a 24 hour period has produced many software packages now in use in industry. They can be broadly sub-divided into the categories of

demand forecasting [135–137, 233, 234, 237, 238], network modelling [14, 15, 21, 28, 37, 39, 40, 69, 72, 92, 96, 97, 124, 204, 244, 258] and pump scheduling [109–111, 121, 171, 181, 189, 239–242, 267]. Other less common computational applications include the use of neural networks for demand forecasting [137], state estimation, dynamic network analysis [114], parameter tuning [201–203] and the use of expert systems [192].

Research into demand forecasting during the 70's using a number of techniques such as Fourier analyses and the ARMA method [233, 234, 237, 238] produced algorithms to enable predictions of consumer demand over (typically) a 24 hour period. Using these predictions, pump optimisation (scheduling) programs are able to determine the most cost effective scenario for the forthcoming period based on initial and final reservoir levels, electricity tariffs and a disaggregated network model, e.g. [175]. However, for all but quite small systems, many of the proposed methods were quite unsuitable for real time use. Hence pump scheduling is essentially an off-line technique. As computer power increases this will be less true and more complex techniques will be used in real-time situations.

On-line control of water distribution systems has been reported in several research papers [45, 77, 87, 168, 187]. Of particular note in this country is work by Fallside [109, 110], Sterling et al [236, 239, 241, 242] and Coulbeck [70, 73] all of whom have worked on the optimal control problem as applied to the operation of water networks. A number of network analysis packages such as *WATNET* produced by the Water Research Centre (W.R.C.) or *GINAS :- Graphical Interactive Network Analysis System* [72] have been suggested for use in an on-line mode. Indeed Coulbeck et al [73] have implemented a system that combines with their *GIPOS :- Graphical Interactive Pump Optimisation System* to provide an on-line control facility. In their system, measurements of network in-flows drive the demand forecaster which produces an update on the last demand prediction. From this up-date, the optimiser calculates a new set of pump outputs with the aid of a network model.

2.3 Telemetry

In many areas, the water industry is implementing new instrumentation, automation and control (I.C.A.) schemes which provide operators with overall control of pumping, water distribution and treatment works etc. [25, 31, 45, 91, 163, 168, 187, 200, 232]. As a part of this, supervisory control and data acquisition (SCADA) systems, ie. computer or microprocessor based telemetry systems, have been installed and now process a wealth of on-line data for immediate display via semi- or full graphical displays.

Telemetry computers collect large quantities of data from field transducers either directly or, more commonly, by polling outstations which are connected to a small number of instruments. The possible connections to a telemetry computer, called 'points', can be analogue, digital or status (ie off/on). The telemetry system may permit certain analogue signals to be displayed directly but in more modern systems the computer processes all measurements in a digital form. The information is displayed on purpose built, often semi-graphic consoles. Operator intervention is necessary to issue control commands back to the system depending on the observed state as reported by the telemetry. Many telemetry computers have simple filter algorithms available as built in functions to enable the operator to observe the mean value of a noisy signal. Furthermore, limit checks can also be applied to incoming signals and these checks form the basis of alarm reporting. Alarms often occur in groups and in this case, logic circuits determine the 'up-sequence' which tells the operator the sequence of events following an alarm state.

An example of a filter algorithm commonly used in telemetry systems is the exponential filter which can be expressed as:

$$x_f^{(k+1)} = a x_f^{(k)} + (1-a) x^{(k+1)}, \qquad k = 1, \ldots \infty \tag{2.1}$$

where (k) is the iteration count, x_f refers to filtered data and x refers to unfiltered data. This simple algorithm is trivial to implement in software and therefore is often used in on-line control applications.

Assuming that a telemetry system exists or is intended for a particular water distribution network then a certain amount of useful real-time data will be available however noisy or corrupt. Telemetry computers are not able to determine (except very crudely) which measurements are true values and which are inconsistent with the behaviour of the network since they do not possess a model of the network behaviour. This observation has motivated research into state estimation techniques reported in Chapters 6 & 7 which in turn, has motivated the work presented in the other chapters.

2.4 Outline of WASMACS Modules

2.4.1 Background

The work reported in this thesis represents the central theme in an on-going research project concerned with the design of a complete on-line monitoring and control system for the operation of water distribution systems.

The research encompasses demand forecasting, static and quasi-dynamic water network simulation, parameter estimation for automatic tuning of network analysis models, large scale pump optimisation, optimised valve control for leakage minimisation, state estimation for on-line monitoring, telemetry simulation for measurement noise level studies and sensitivity analysis, meter placement studies, network security analysis, relational database design, expert systems design for prediction of anomalies in demand forecasting, the application of neural networks and the design of very fast fully interactive graphic display systems. This project has resulted in the creation of the suite of software known

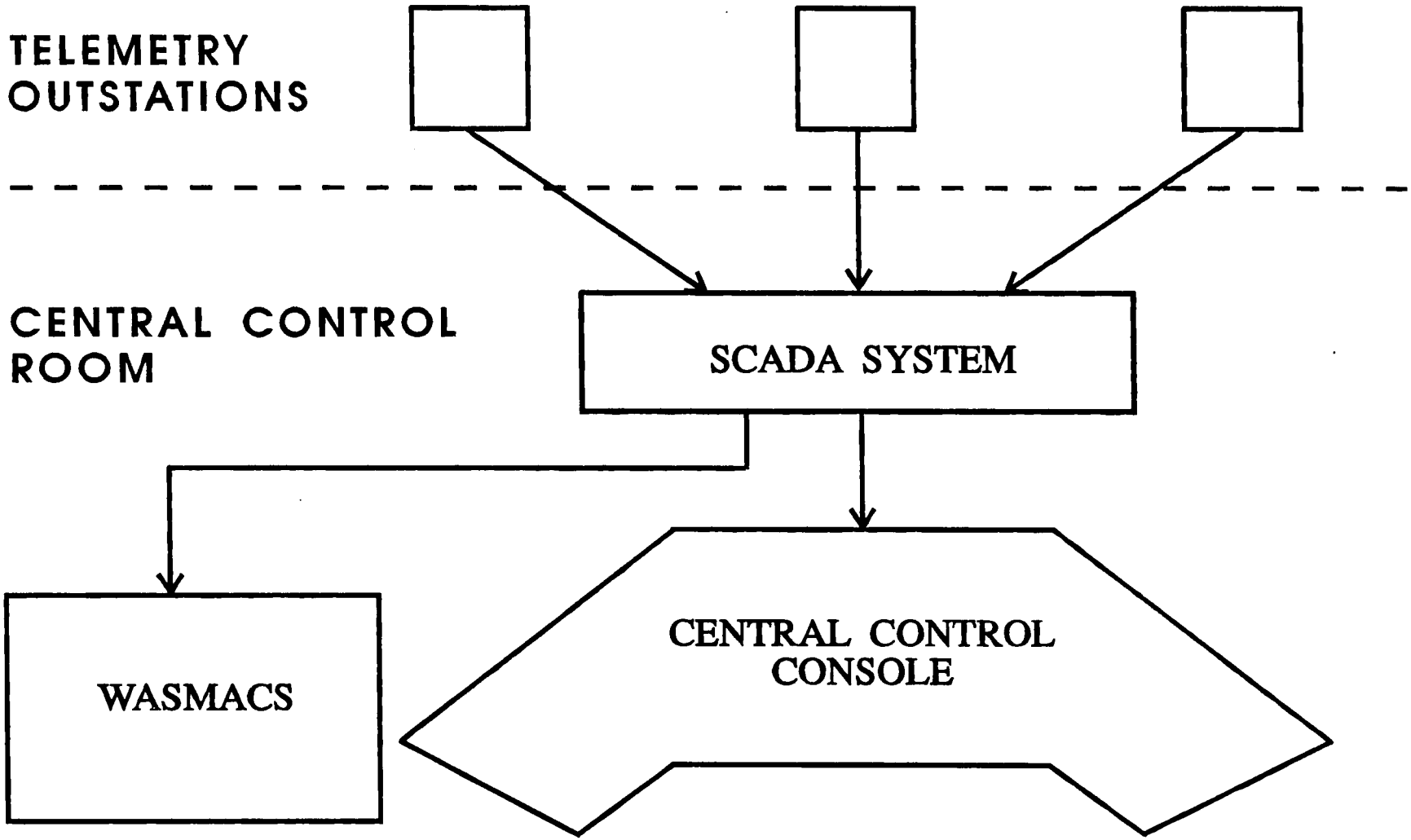

TELEMETRY
OUTSTATIONS
CENTRAL CONTROL
ROOM
SCADA SYSTEM
WASMACS
CENTRAL CONTROL
CONSOLE

Fig. 2.1

as 'WASMACS'. The intention and general philosophy of the research is to develop software for use, in a non-invasive way, in the control room as shown in Fig. 2.1.

In this section, some of the relevant software modules which make up the WASMACS suite will be introduced in order to explain the framework of the overall on-line control strategy.

2.4.2 Demand Forecasting

The objective of water distribution control is to satisfy consumer demands wherever and whenever they occur. In the long term, an accurate prediction of the maximum load gives an indication of any expansion needed in the system. In the short term, the knowledge of the variation of water consumption makes it possible for the operator to take appropriate control actions. In the case of computer assisted control, short-term predictions provide the necessary information upon which optimisation depends, the more accurate and reliable the demand forecast, the more accurate and reliable the optimisation.

The method used is based on the ARIMA modelling technique. After stationorisation of the demand data, the parameters of the model are estimated, and the suitability of the model structure is checked. The model is then expanded in time in order to produce predictions. Parameters of the model are updated every time a new demand measurement becomes available. The general form of the ARIMA equation being:

$$(1 - \phi_1 B - \phi_2 B^2 \ldots - \phi_p B^p)(1 - B^d) Z_t = (1 - \theta_1 B - \theta_2 B^2 \ldots - \theta_q B^q) A_t \quad (2.2)$$

Where:

p is the order of the Auto Regressive component;

q is the order of the Moving Average component;

d is the difference order which account for the trends in the time series data that have a period longer that the span of the time series;

Z_t = value of time series at time t;

A_t = value of residual white noise time series at time t;

B = the backshift operator, where

$$BZ_t = Z_{t-1} \tag{2.3}$$

$$B^d Z_t = Z_{t-d} \tag{2.4}$$

$\phi_1 \ldots \phi_p$ = Auto Regressive weighting parameters;

$\theta_1 \ldots \theta_q$ = Moving Average weighting parameters.

Although this method of demand forecasting is normally highly accurate, demands in water supply areas are prone to large fluctuations caused by calendar based events, weather based events and changes in the network. Knowledge pertaining to these effects can be stored in a knowledge base and used to alter the demand forecast to provide a more reliable prediction of the actual demand.

An alternative, and more robust method developed by this author and his Ph.D. student, uses a simulation of a neural network in an algorithm known as a single layer neural network [137]. The advantage of the neural network approach is that the inputs to the network can include weather data such as temperature, rainfall and sunshine hours. The method employed for on-line use is to first categorise the preceding days into 4 'day types' which describe a range of general weather conditions and base the forecast on that information so that the operator does not have to guess temperature, rainfall and sunshine hours for the day of the prediction.

A fully colour graphic interface is provided since in demand forecasting, it is considered important to give the operator as much information as possible. Only then can the operator make informed decisions about the current evolution of the demand as

it varies over the preceding week (for instance). The user can select between weekly forecasts, daily forecasts, stepping forward or backward a day at a time to investigate either new forecasts or examine the prediction error when the actual demand data has been received. Details from the knowledge base such as average temperature, sunshine hours, rainfall etc. are also shown graphically where appropriate [135–137].

2.4.3 Telemetry and Distribution System Simulation

The static system simulator has principally been developed to provide a facility for operational control policy studies without the difficulties inherent in carrying out such studies on actual water systems. The simulator provides realistic data about the system performance by modelling telemetry malfunction in addition to normal measurement noise. In its present form, the package allows operator intervention to change the status of any control valve in the network, as well as to introduce additional nodes representing leakages. The simulator also enables the level of measurement noise to be varied and it allows gross measurement errors to be introduced arbitrarily into the telemetered data base. There are three programs in this suite: the network simulator; the telemetry simulator; and the operator's interaction program. The three programs were designed as a test bed upon which the other algorithms (e.g. state estimators) developed in this work could be validated. The overall structure is shown in Fig. 2.2.

The simulator calculates an exact state of the water system by solving the square set of non-linear mass-balance equations using a Newton-Raphson iterative procedure. The flow model can either be the Hazen-Williams equation or the Colebrook-White equation. High computational efficiency has been obtained through the use of sparse matrix techniques. Network parameters can be stored in a relational database or can be read from files. A more detailed description of the equations involved is presented in Chapter 5 and in [254].

WATER SYSTEM SIMULATOR PROGRAM INTERACTION

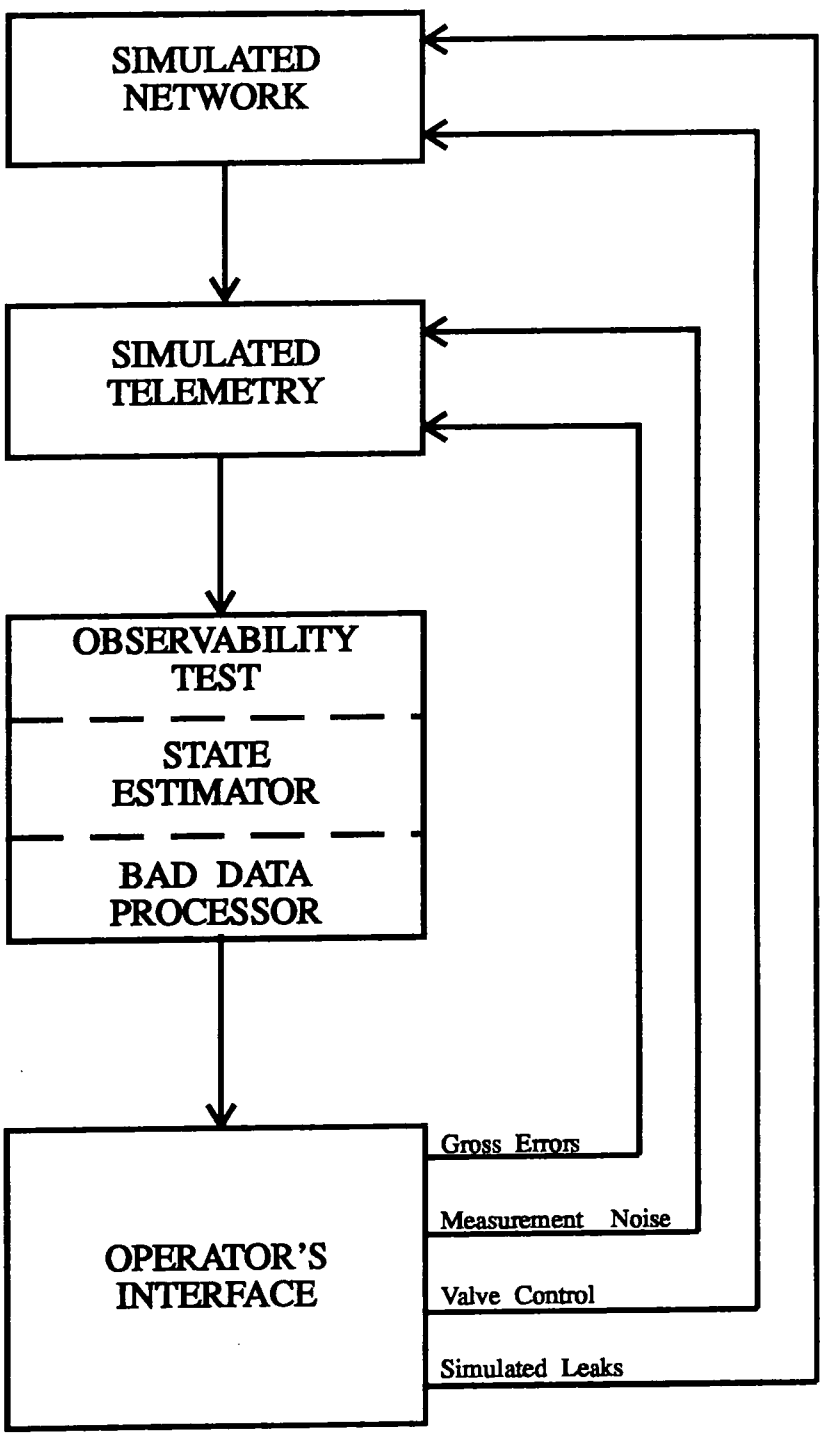


Figure. 2.2

2.4.4 Meter Placement

Very few water distribution networks have sufficient on-line measurement systems to monitor the network state in real time. However, in many cases, some instrumentation is available. For example, pumping stations and reservoirs are normally reasonably well instrumented and metering points are often available for direct installation of portable instruments for data logging or minimum night flow monitoring. Such metering points could in fact be utilised in an on-line monitoring scheme. In order that a network is state observable, a necessary (but not sufficient) condition is that

$$\rho = \frac{m}{n} \geq 1 \tag{2.5}$$

where ρ is the global measurement redundancy, m is the number of measurements and n is the number of state variables. A useful alternative is to define the redundancy in terms of the surplus measurements, ie $m - n$, since this indicates the maximum number of bad data that can theoretically be detected.

The majority of measurements used by a state estimator can, in fact, be nodal load (demand) measurements obtained on a house count basis without incurring installation costs. These 'measurements' (sometimes termed pseudomeasurements [12]) together with on-line pressure and flow measurements from the field telemetry make up the measurement vector $\mathbf{z}$. The physical location of these measurement points is crucial to the performance of any state estimator [150] and provides the sufficient condition for observability, namely that the matrix,

$$\mathbf{\Sigma} = (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-\mathbf{1}}\mathbf{J}) \tag{2.6}$$

is non-singular or, equivalently, that the Jacobian, $\mathbf{J}$, is of full rank. (The terms of Eqn. 2.6 and its derivation are given in Appendix 2). A number of authors have proposed methods for observability determination based on network topological algorithms [13, 56, 57, 116, 153, 179]. The reference [116] departs form the other methods, which are

largely variations on the same theme, by using the programming language PROLOG in an exceedingly compact algorithm.

The observability problem consists essentially in determining whether the measurements currently available to the state estimator provide sufficient information to allow the computation of the state estimates. The meter placement problem extends this concept by attempting to achieve a metering system design which will guarantee reliable estimates even in the event of metering or telemetry failures. Optimal meter placement therefore consists in determining the positions of a limited number of meters whilst maintaining the metering and installation costs to a minimum [1, 133].

Generally, the addition of a meter to an already observable network will increase the accuracy of the state estimates. This is because the basic information used at the nodal level in any simulation or state estimation program are the nodal demands which are inherently very unreliable measurements since it is virtually impossible to predict the general behaviour of small populations of consumers. The new meter installation, on the other hand, constitutes a cost which should be kept to a minimum. The objective of optimal meter placement is therefore to minimise the variance of the state estimates whilst minimising the length of the connection from the control centre and also minimising the total number of installations. This is therefore a multi-objective minimisation problem and is solved by systematically improving the measurement system starting from one that contains only demand measurements. The model used to define an optimal measuring system for state estimation can be stated as follows:

$$\min C_1 = \sum_{i=1}^{n} S_i^2 \tag{2.7}$$

$$\min C_2 = \sum_{j \in \Omega_T} d_j \tag{2.8}$$

$$\min C_3 = \quad N_k \tag{2.9}$$

subject to:

$$S_i^2 \leq \beta_i \quad i = 1, \ldots, n \tag{2.10}$$

for a system with n state variables and m measurements and where d_j represents the distance from the j^{th} meter to the control room, N_k is the total number of instruments used, and

$$S_i^2 = \quad \Sigma_{ii} \tag{2.11}$$

and Σ is as defined in Appendix 2. $\beta$is the upper bound on the variances as defined by the user.

2.4.5 Valve Control

The primary aim of a water distribution control system is to maintain sufficient pressure to ensure that all demands, wherever and whenever they occur, can be met. The idealised objective of network operation is to keep the pressure of the water, relative to ground level, in each individual node constant at a predefined level. This set of pressures is often referred to as an optimal head profile. However, owing to the head/flow relationships in the network the optimal head profile can only be maintained in a few nodes of the network, whilst in the others, the operational pressure remains higher. As the complexity of a distribution network grows, the task of achieving an optimum pressure becomes more and more difficult and the average over-pressure tends to increase. This in turn results in an increased energy cost, increased volume of distributed leakages and higher risk of major bursts, particularly during the night period when the pressure rises due to decrease of consumer demand. The volume of leakages in complex networks is, on average, approximately 25 – 30% of the total production and consequently represents the main potential for improvement of water distribution system economy.

Pressure control is possible through the use of valves (pressure reducing valves or PRV's) installed in the pipe network. However, computation of the optimum (time-

varying) valve settings is a relatively difficult task due to the high dimensionality of the optimisation problem and nonlinearity of the network model. To overcome this, a method which is based on successive linear programming using the sparse revised Simplex technique was developed which takes full advantage of the sparse structure of the problem to achieve low solution times. The problem is posed rather like the state estimation problem but instead of minimising errors between the observed and computed measurements, the algorithm minimises the errors between the observed and desired head profile by adjusting valve settings within its network model. This work was initially developed by Bargiela [236] but has been slightly refined as reported in two papers [49, 266]. Similar work has also been undertaken by Jowitt et al [122, 147].

2.4.6 Pressure Control Using Both Pumps and Valves

Using valve control only it is possible to effect this type of pressure regulation but only at a cost. Since no attempt is made to reduce the output of the pumps, the action of the pressure controlling valves is to throttle the flow and hence dissipate some of the energy from the pumps. A natural progression from valve control is to implement a combined optimal control system that utilises any available variable speed pumps as well as the control valves.

The minimisation of the difference between the observed and the reference pressures is achieved by adjusting both valve and pump parameters in a linear programming model. There is much similarity with the method outlined in Section 2.4.5 for valve only control except that in this method the control variables include pump speed. In the algorithm, numerical weighting coefficients are used to ensure that the pumps are not subjected to undue cycling [266].

2.4.7 Discrete Time Simulation

Although a static analysis suite is provided, it is often necessary to carry out extended period simulations to validate operational control policies, to carry out design exercises and to determine limits of system security.

The discrete time or 'dynamic' simulator (strictly speaking, 'quasi-dynamic' since pressure transients in pipelines are not computed) performs an extended time period simulation calculating all network flows and pressures as in the static simulator but integrating reservoir levels and resolving pump equations at each step to derive the new boundary conditions for the next time step. Demand profiles are stored individually in the relational data base, as are pump profiles. Demand information is stored on a per-half hour basis whereas pump profiles are stored in activity/time-of-day couplets. Models of fixed and variable speed pumps are available together with variable throttle, level switched valves, control valves, pressure switched valves etc. The time step interval can be varied during the simulation and pump curves can be interpolated from measurement data.

2.4.8 Pump Scheduling

Pump scheduling is a dynamic optimisation problem that must take account of system storage, time-varying electricity tariff structure and operational (hydraulic) constraints in order to minimise the overall operational costs for a particular control period. The problem must be formulated and solved so that given the forecasted demands for the coming 24 hour period (say), the initial and final reservoir conditions, the unit and maximum electricity charges and the set of hydraulic network constraints, an optimised programme of pump activity is found. Hence the actual optimisation is non-linear and the constraints are non-linear. Unfortunately, the problem as stated is also non-convex, due to the need to include the maximum electricity tariff. The inclusion of discontinuous variables representing fixed speed pumps, one way valves etc. also cause non-convexity.

In the WASMACS suite, a new non-linear optimisation algorithm has been designed employing the generalised reduced gradient (GRG) method and a non-linear network sensitivity analysis. Thus, the scheduling problem encompasses the entire network operation problem rather than being a simplification or ignoring the non-linear hydraulic constraints. The only simplification that is currently made is in the treatment of the monthly maximum demand charge. To take into account of this, a hybrid model is used in which the cost of the monthly maximum demand is weighted recursively into the overall day's operational cost function. The algorithm has been designed so that iterations always proceed in a feasible solution space and thus will find better and better (local) optimal solutions.

2.4.9 Security Assessment

Security assessment for water supply and distribution systems is split into two levels: the lower level is concerned with distribution network operation and levels of service whereas the higher level considers the security of bulk supply. The discrete time simulator is used to model the system behaviour beyond the normal range of operations, by modelling failures of system components and identifying the areas where violations of statutory pressures has taken place. An interactive graphics facility is being developed to permit the system operator to select and evaluate any potential contingency.

Security study at the higher level aims to determine the minimum storage of service reservoirs to safeguard the bulk supplies against some anticipated crisis events. The effect of different failure events on the supplies is identified by network simulation and the occurrence of such events are assessed by probability analysis. Based on the dynamic relationship between the failure probability of supply facilities, the availability of pumping capacity and the diurnal demand profile, a time-varying minimum storage level for the service reservoirs can be determined. For different levels of risk, a set of 'control curves' can be produced to provide the system operator with secure operating guide-lines [265].

This system is currently under development and is designed to be used in conjunction with the optimisation suite.

2.5 Proposed Control System

The major components of the suite of software described above are designed to work in a composite control loop as shown in Fig 2.3. This is described as follows. The measurements from transducers in the field are validated in the state estimator and any errors detected are removed as described in Chapter 6. The state vector is then made available to the pressure control package outlined above in Section 2.4.5. Disaggregated demand data is passed to the demand forecasting module which calculates the next 24 hour demand profile, principally for use in the pump scheduling package and the security assessment package. Output from the pump scheduling program is passed to the pressure control program which uses the current state vector to determine an incremental, optimal adjustment for the valves and pumps in the network. The control signals derived from this last package are passed to the telemetry system.

The proposed system therefore comprises a cascade control loop with the longer term controller (pump scheduler) computing the set points for the faster controller (pressure control) which controls the pressure changes resulting from short term fluctuations in demand not predicted by the demand forecasting system. The outer loop is concerned with the diurnal system behaviour whereas the inner loop calculates a snap-shot optimisation based on the current state of the network.

Underpining each of these packages and providing a number of different services is the relational database. The service provided to the state estimator (which, in this sense is nearly identical to the pressure control program) is detailed in Chapters 3 and 4. The pump optimisation, security assessment and demand forecasting systems all exploit the relational database and use it for model storage, pump and valve profiles, demand profiles

FULL ON-LINE CONTROL LOOP

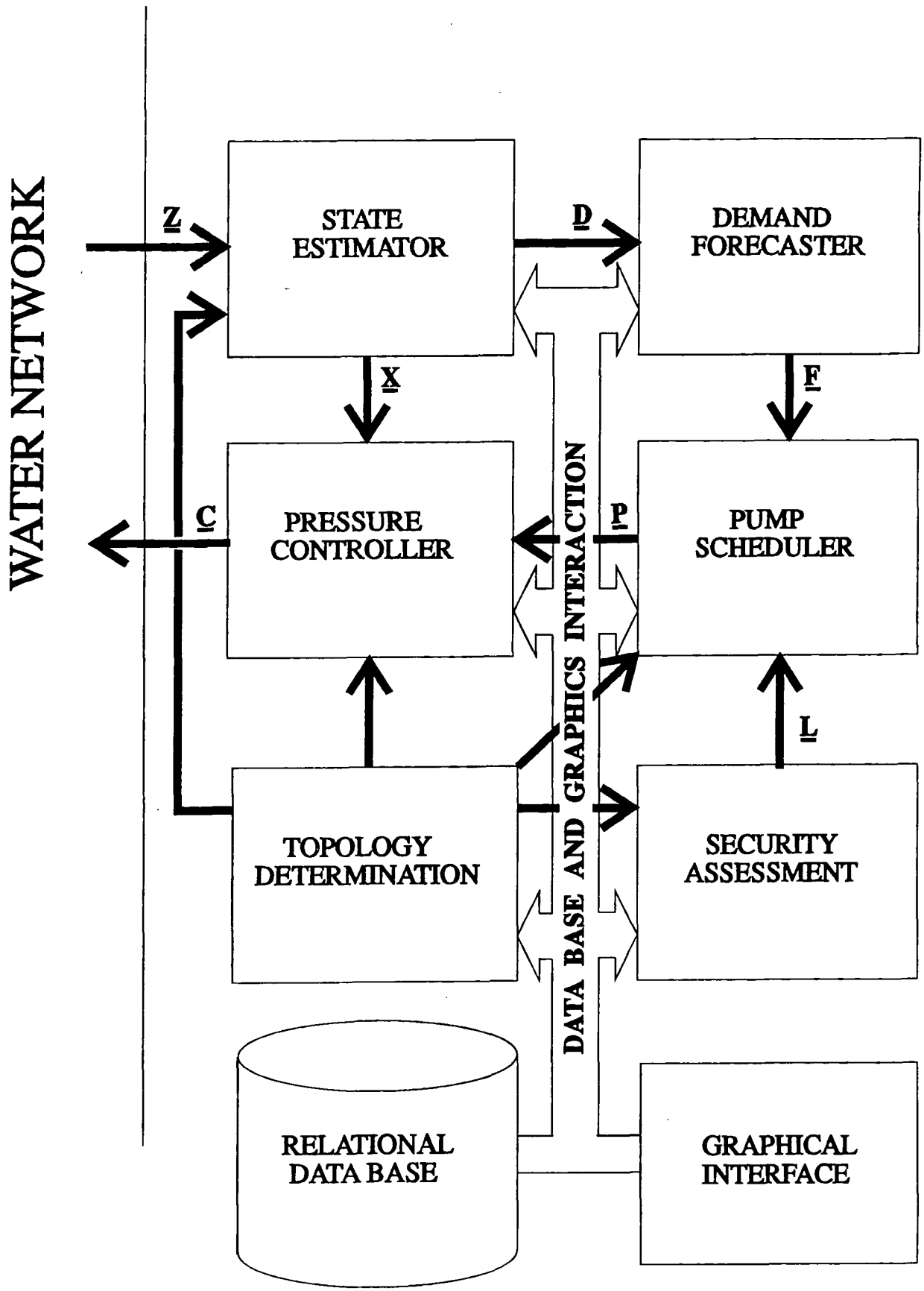


 FIG. 2.3

for a range of user types (e.g. industrial, commercial, agricultural and domestic) and a host of other data.

In the chapters which follow, only the components essential to on-line monitoring are discussed whereas the remaining components, considered outside the scope of this thesis, are reported in the references already given.

2.6 Conclusions

The macroscopic theme of this work has now been presented. As stated earlier, the objectives of water systems control is to provide a continuous balance between water supply and demand at a minimum cost without violating predefined limits on head, water quality or the amount of water stored in the system for emergency demands. This can only be achieved optimally if two things are known at all times: the system's current state and the system's predicted state. Therefore a pre-requisite to optimal control is optimal observation, which also has its own pre-requisites. To build an optimal observer one needs a representative model of the system and for this model to be representative, it has to be kept constantly up to date.

Although some of the other packages mentioned in the previous section have been developed by the author with the help of others they are not the main concern of this thesis, instead, the principal concern of this work is that part of the control loop which represents the provision of a continuous and accurate determination of the system's state vector together with the implicit problems of the maintenance of the observer model.

CHAPTER 3

FLEXIBLE ON-LINE DATA STORAGE

3.1 Introduction

The primary function of the algorithms developed in this research is to provide a database of live, validated information about a distribution network and its associated equipment. The techniques to provide this information are presented in the chapters 5,6 and 7, however it is necessary first to consider how that data should be managed.

Water distribution networks are highly complex and a vast amount of data is necessary to describe them accurately. Some data are static (e.g. the location of a pipe), some data are slowly or infrequently changing (e.g. the internal roughness of a pipe, or the network topology) and some data are dynamic (e.g. the measurements). The fact that a significant portion of the network data is constantly changing is often ignored in the design of simulation and analysis software and this can result in the stored information in use by these algorithms becoming out of date and eventually bearing little relation to the actual physical plant. This becomes a serious matter in on-line control schemes if the methods used for control need to rely on an accurate mathematical model of the system.

In storing large quantities of data it is normally convenient to partition them into categories such as tables, however, it is crucial to ensure that data stored in a multitude

of tables are self consistent and that access to subsets of this data is straight forward. Efficient methods of data management that provide access to both parametric data and dynamic data for a distribution network are therefore necessary.

The term 'database' is frequently used to mean a collection of (ordered) data stored on a computer where access and update permission is granted to an arbitrary number of separate tasks. Recently a number of proprietary software systems, called Database Management Systems (DBMS), have become available for the maintenance of large amounts of data for which the user interface and management tools have been designed by the supplier. It is important to recognise the difference between the two areas of research under the name of 'database design'. The first is the design of the DBMS, this includes the design of the user interface, the method of processing requests, data handling at the logical (file) level and data handling at the physical (disk) level. The second area is the design of a database *schema*. The schema is the form in which the user perceives the data to be stored, schema design involves an understanding of the data model and of the functional dependencies of the data within the model. In the present work we are only concerned with this latter problem and not with the database management system design. Two proprietary DBMS's have been used to support this work, namely 'RDB' produced by Digital Equipment Corporation and 'Ingres' produced by Relational Technology Inc. [81].

This chapter introduces the design of a relational database which is shown to result in an efficient mode of operation although the access times are significantly increased in comparison to an earlier 'common block' approach. It is argued that the benefits arising from the database approach outweigh the increased access times. The chapter is organised as follows: a review of recent work in the application of database technology is given in Section 3.2; Section 3.3 discusses the method of sharing data in a database through the use of common areas of computer memory; Section 3.4 reviews current database technology; Section 3.5 presents the design philosophy and criteria for a relational database schema

design to support an on-line monitoring and control facility; in Section 3.6 the actual design is presented and the various tables that make up the schema are listed; in Section 3.7 the concepts of database optimisation are considered; Section 3.8 explains the method of accessing the data in the database and outlines the data control problem for programs designed for on-line operation; performance testing is reported in Section 3.9 using water network related database transactions; finally the relational database approach is summarised and conclusions are drawn in Section 3.10.

NB. In this chapter and the next, `typewriter script` is used when referring to a specific database operation, such as `insert` or when referring to a table or attribute within a table, for example `Pipes` or `Pipes.length` respectively. Table names are always given a capital first letter whereas attribute names are written entirely in lower case. The table name is omitted, e.g. `length`, whenever it is clear which table is being referred to from the context.

3.2 Review of Recent Work

Although work on relational DBMS design had been widely published in the early and middle 1970's, e.g. [30, 60–63, 80, 253], much of the database schema design undertaken in that decade for utility applications (which was mainly limited to power distribution systems) was proceeding independently of these early ideas and hence practically all of the systems proposed were non-relational. The then more popular CODASYL approach [59] was having greater impact on design methodologies and evidence of this bifurcation in design approaches is still apparent today.

The major design problems and ideal requirements of a database intended to underpin the operation of a large energy management system (E.M.S.) were identified by Anderson and Smith [9]. In their paper a number of the traditional problems of database management system design such as read/write locking, data security etc. are introduced

along with certain requirements unique to the E.M.S. application area. These additional requirements included the need for fast database access times, data sharing between a multitude of users or tasks and the ability to access a database history to determine the scenario leading up to and during particular events of interest. Their paper outlines the design of both an inverted list database management system and a schema to overcome the problems mentioned.

Hierarchical database structures closely match the natural order of objects in large scale physical systems and this inherent structure is exploited by many researchers in the field [138, 208, 209]. Ben-Yaacov, [23] introduces a 'fixed' hierarchy of four levels comprising: the database system name; the major objects within the system (e.g. busbars, lines etc); the individual components of those major objects and lastly, the parametric and measurement data pertaining to these components. The advantage of this restrictive approach is there is that the query language is straight forward to construct and the access times are kept to a minimum. Lehman and Mattioni [154], provide an overview of the hierarchical, network and relational database models, arguing that there will be a trend towards the use of proprietary database management systems in the future rather than 'home grown' types.

The 'PRIMO' database system [151] mimics the relational model (but fails to meet Codd's requirements [64, 65], as listed in Appendix 1, for a fully relational database). However, the approach is based on two dimensional tables (arrays) which are queried using row – column ordinates. A set of these arrays make up a schema and any number of schemas can be handled in the overall system. Because of the simplicity of the system it is considered to be suitable for real-time applications.

In [17] and [18], Baumann and Bayer introduce a hierarchy for the time dependant nature of data in an on-line monitoring and control system and also discuss the various operational tasks ranging from on-line control to long term planning. Their observations

lead to recommendations regarding the different treatment of slowly varying and quickly varying data in order not to loose the utility of database systems on the one hand and not to compromise the real-time requirements on the other.

Schwab et al [215] argue that the many unused features that are present in commercial database systems add an unnecessary system overhead. Instead, they propose a semi-relational system that has been designed with performance in mind. The ability to join relations or form views is not provided in their system. A fully relational scheme was developed by Dambourg et al [79] and applied to the problem of (off-line) protection system design for power systems. In that approach a commercial system was used along with a variety of network analysis software. More recently Gann [117, 118] has investigated the applicability of two commercially available relational database management systems in conjunction with discrete time simulations of power distribution networks and found that even with a large (72Mbyte) RAM disk, the access times did not suggest that relational databases should totally replace other, more time efficient, methods of storing data. Further notes on power systems database design can be found in [214].

3.3 Data Sharing Using Task Common Memory

An efficient but inflexible means of allowing two or more tasks to communicate is via shareable areas of computer memory. This can be achieved by the use of identical FORTRAN77 named task common blocks in the tasks concerned and the 'memory resident' installation of those common blocks. When a shareable area of memory is allocated to two or more tasks, the physical addresses of the identifier names defined as common to the separate tasks are identical. The method is computationally efficient because at run time there is no actual transfer of data from one task to the next since both tasks simply use the same addresses for the shared data. Since there is no computational overhead, there is a very significant improvement over (say) the use of a shared file to pass the data especially if the quantity of shared data is large. However, since the structures to do this

are static data structures defined prior to linking the object modules, it is impossible to change them at run time. Consequently, the programmer is tempted to define areas which are much larger than is needed to avoid subsequent re-sizing and re-installation.

Furthermore, task common blocks are not defined in common high level languages such as ANSI FORTRAN77 and are not available under all operating systems. In any event, the method of installation is operating system dependent and hence code to do this is not portable. Another draw back of this method of data sharing is that it is not straight forward to mix programming languages and this constrains the programmer to use one chosen language only. Since the data structure is fixed it is also difficult to add third party software modules that should have access to the database. A final problem with this type of structure is ensuring data consistency. There is an inherent danger of inadvertently overwriting the data although this can be theoretically avoided by using a system of flags passed between the tasks.

3.4 Current Database Technology

There are three basic data structures employed by database management systems, namely:

- Hierarchical databases;
- Network databases;
- Inverted list databases (of which relational databases are a subset).

Hierarchical data can be viewed as a tree structure with an implied superiority and dependence for each record stored in the database determined purely by its position in the tree. The tree begins at its root node (the record stored there being the only one without a superior or *parent* record) and ends at its leaf nodes (whose records have no dependants or *child* records). At each node the record comprises three parts: the pointer (or link)

to the parent record, the data itself and the pointers to the child records. A hierarchical database is a collection (or forest) of such tree structures. Note that the structure of the hierarchical database is determined by the data itself and that deletion of nodes within the database can result in the so called *triggered deletion* of all the dependants.

By way of an example, consider the hierarchical data model shown in Fig. 3.1. This structure shows the logical dependence of each instance of each data item. For example, a given instrument, "Flow meter 1" is to be found in "Pumpstation B", in "District B", in "Zone B", etc. Hence the search:

```
Select all instruments in Pumpstation B
```

would therefore appear trivial and would not consume excessive computer time in execution. However, the following request would almost certainly be disproportionately more time consuming:

```
Select all flow measurements
```

This second request would involve the search of a large part of the database since it is the case that instruments can "belong to" any arbitrary piece of equipment and hence it would therefore be necessary to search every equipment node to see if there were instruments present currently reporting values related to flow. These two examples show that the database structure should be influenced by the intended usage of the database if fast data retrieval is required and that an elegantly designed database (from the point of view that it matches the real world data dependencies) may produce a system that requires complicated and time consuming queries when unforeseen operations are necessary. It therefore follows that to design a hierarchical database perfectly it is necessary to consider all the anticipated queries before settling on a particular structure, a requirement that is virtually impossible to fulfil.

Hierarchical Data Structure

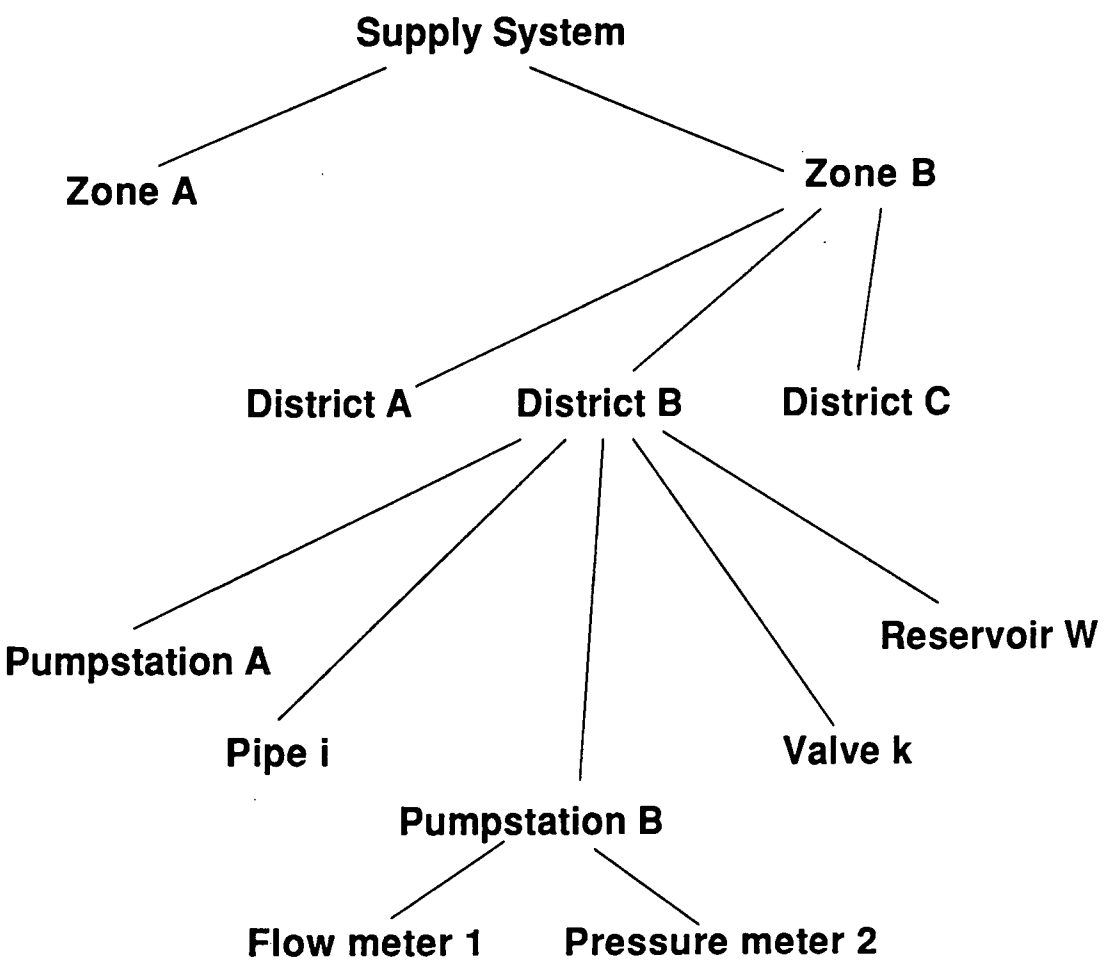


Figure 3.1

The network data model is quite similar to the hierarchical system but here the tree is replaced by a graph. One can view a hierarchical database as a vertical structure and a network database as horizontal. Whereas in the hierarchical database the links do not themselves carry any information, in a network database they often do. Links, in a network database, can be of two types: information carrying and non-information carrying. Information carrying links specify inter-record relations such as '... is a member of ...' or '... is located in ...' and are necessary when there is more than one path from one record to the next. The data definition model for a network database takes the form of a digraph. Complex search algorithms are required for the most efficient traversal of a network of records. The systems usually employ an explicit query language to determine the path to take after examining each record [82, 83, 208, 209].

In relational database systems, data is stored in what appears to the user as two dimensional tables, referred to as relations. The major difference between this approach and that of a hierarchical or network data model is that each data element stored is a simple data item. That is, pointers or links to other data items are not stored explicitly but are implicit in the content of the database, this is in contrast to a hierarchical or network model where this information is in the database structure. The advantage of this type of linkage is that application programs can access a specific record directly while application programs which need to exploit the parent, child and sibling relationships have to be provided with the connection information [215].

The main advantage of the relational approach is its structural neutrality [154] and this also highlights another major difference between the relational approach and the hierarchical or network approaches. Namely that the structure of the relational database does not reflect the structure in the data model. Instead, the structure of the data model is described purely by the information stored i.e. the relations of the database. Hence the designer of a schema for a relational database need not worry about the likely queries to the database and need only consider the meaning of the data required. This facilitates

a flexibility to modify the structure that is not shared by the hierarchical and network database systems, since in these latter approaches the designer must provide for all desired interconnections between items when the database is first designed [31].

Another consideration common to all database systems is that of data integrity. This can be achieved by ensuring that all the transactions are disk based rather than RAM based, then loss of power does not mean loss of data. Further integrity can be built into the hardware, e.g. mirrored disks etc. Bond [29] has developed a relational database system that uses a multi-computer network to ensure integrity without the need to purchase extra hardware.

Codd's initial design of a relational database was influenced by the idea of a universal data sublanguage based on predicate calculus to query the database [60, 61, 62], although it was not until 1974 that Chamberlin and Boyce [41] actually proposed the first database sublanguage which was then called SEQUEL. Software houses producing DBMS's were also designing query languages and out of this work came an attempt to produce a standard query language to allow users to create schemas and transfer them to different DBMS's. In 1980 the language SQL (Structured Query Language) emerged out of a process of continual revision starting with SEQUEL and leading to the intermediate languages SQUARE and SEQUEL2 [42, 207]. SQL has now been accepted as an international standard by the American National Standards Institution (ANSI X3.136-1986) [8] but despite this few DBMS designers have adhered strictly to the standard and most offer extensions to it.

SQL can deal with the components of a database at various levels. It contains statements that can be used to define the structure of the data, called Data Definition Language (DDL), it has statements for data manipulation, known as Data Manipulation Language (DML) and finally it contains statements for the control of *transactions* caused by the DML. This latter sublanguage is called DCL (Data Control Language) [141].

3.5 Design Philosophy

The flexibility provided by the relational database itself combined with the power of 'embedded SQL' (i.e. SQL embedded in a host language such as FORTRAN) enables a new approach to be taken in handling the data produced or used by an on-line monitoring system for water distribution networks. In this section a design philosophy that attempts to maximise the potential benefits of such a system is presented. In addition, a number of terms that arise in current database work will also be introduced to clarify the explanation of the actual design proposed in Section 3.6.

The main objective of the complete on-line monitoring system proposed in this work is to provide the plant operator with reliable time varying plant information and to avoid burdening him or her with the additional duties of plant model maintenance. To satisfy the second part of this aim the data required for storage in the database must exactly reflect the physical system and be stored in a form that can be used directly by the analysis code. This combined goal presents one of the main difficulties in the schema design which, to date, does not appear to have been solved by other researchers. The primary objectives of the database design can now be stated.

Main Aims:

1.0 The complete system (database + analysis programs) must be able to respond quickly enough to ensure that the information provided is always relevant (ie real time operation is required);

2.0 The system should exploit the desirable features of the relational data model in providing consistency and flexibility of data manipulation;

3.0 The system must be able to determine the network topology automatically;

4.0 The system must keep track of the consumer demand location as the status of switching valves change;

5.0 It should be possible to 'replay' events that occur in normal network operations.

3.5.1 Data Requirements

As shown in Fig. 3.2, we can consider each module in terms of its category either as a data input module, for example the SCADA system, as a data processing module such as the state estimator or finally as a data presentation system, i.e. the H.C.I. (Human Computer Interface). As an example of some of the data required in this on-line monitoring system, consider the requirements of the state estimator which has inputs of on-line measurements of flow, pressure, disaggregated demand estimates, estimates of the measurement variances, parametric information and topological information. The outputs take the form of validated measurements, validated demand estimates, all network flows and pressures and a list of 'error information' derived by the state estimator. Much of the output is to be displayed graphically so that graphical coordinate information is also necessary.

As is shown later in this chapter, the design of the database schema is such that other packages of the water systems monitoring and control suite which have been developed in earlier work [191] can share the same data access methods as the above packages. Since the underlying philosophy of the design is to store the data in the most simple format possible it is only necessary to consider the data itself and not the way in which it will eventually be used. Therefore there is no need to consider the function of any individual packages in this chapter.

There are two basic data categories in each module of the WASMACS suite:

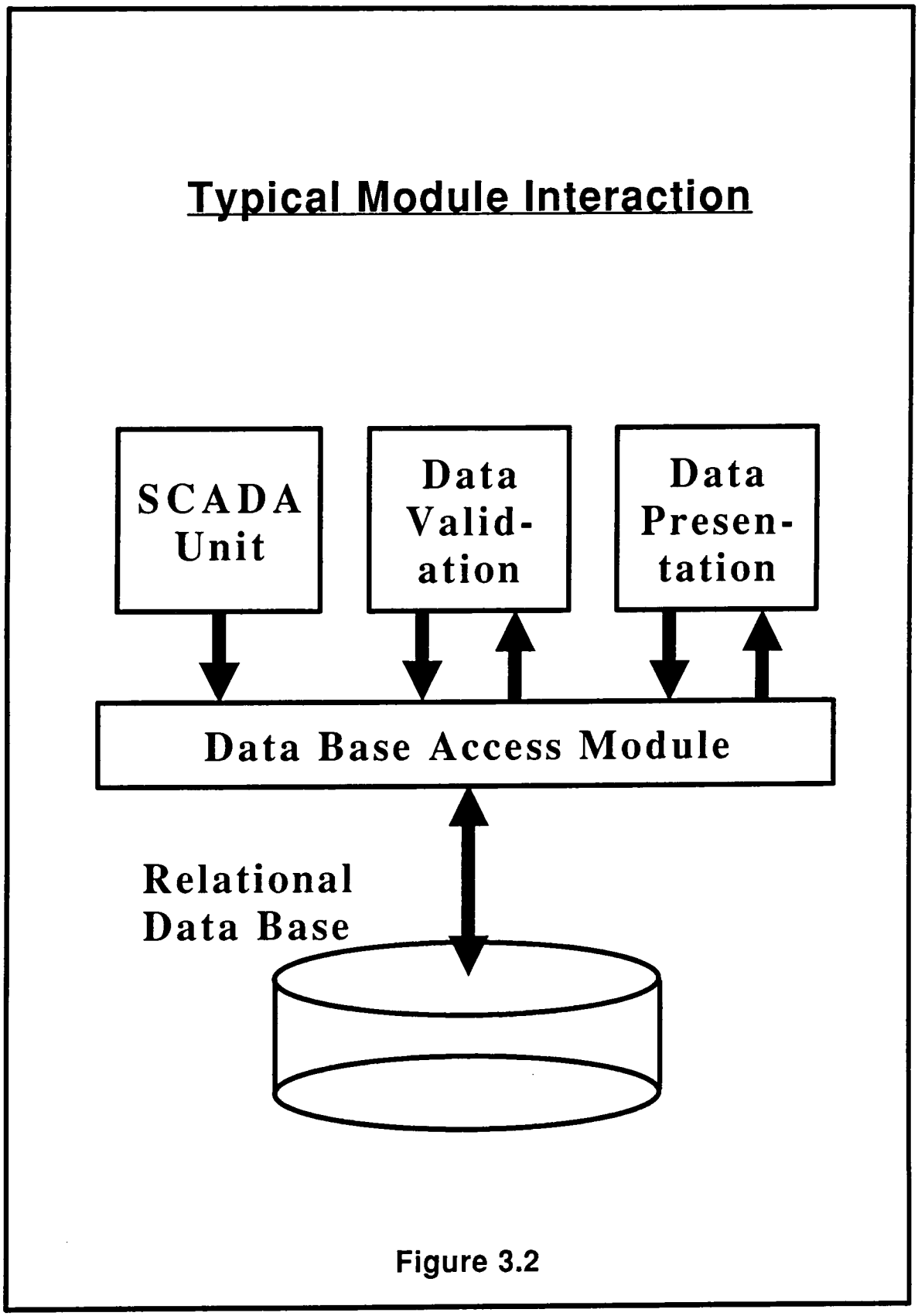

Typical Module Interaction
SCADA Unit
Data Valid-ation
Data Presen-tation
Data Base Access Module
Relational Data Base

Figure 3.2

Data local to each task:

– Algorithm control data;

– Local variables.

Global data, shared by two or more tasks:

– Parametric (or system) data;

– Measurement data;

– Results data.

The first (obvious) conclusion is that data in the first group need never be stored in the database. The second group of data is global in nature and more than one task or user may require a right of access to it. Therefore it is this set of data which is considered for the purposes of database design.

The database will hold all the system data necessary for the analysis routines to function and additional data that is relevant in general to the system under control. The data stored will belong to one or more of the following classes:

1.0 *Asset Data*: Data relating to the general assets of the water company, comprising:

1.1 *Objects* or *Static Data*: Data relating to specific physical objects:

1.1.1 *Parametric Data*: Data that is used by one or more analysis routines;

1.1.2 *Benign Data*: Data that is irrelevant to the model but is there to aid the operator;

1.2 *Dynamic Data*: Data that is normally time varying;

1.3 *Topological Data*: The data that describes the possible topological structure of the network;

1.4 *Past Data*: Data describing the past state of the static and dynamic data;

2.0 *Meta Data*: Data about the data in the database.

It is worth noting here the distinction between 'information' stored in the database and 'data'. Date [82] makes this subtle distinction for an arbitrary database but in our case the dichotomy is easier to see and moreover it clarifies the approach taken in this particular schema design. The term 'information' refers to the *meaning* of the data stored. It is this meaning of the data that imposes the structure of the design by virtue of the relationships between some data items and others. For example, the status of a valve (open or shut) is on the one hand atomic data (ie data in its simplest form) but on the other hand 'status' conveys topological information via the valve's relationship with the objects to which it is joined.

An 'event' is defined here as a change to some data value in the database or the insertion of new data into the database. An event occurs when a valve changes state or when new measurement data is inserted into the measurements table. It is easy to see that events carry a temporal attribute since they occur at a given moment in time known as *valid time*. Objects must be given this attribute since they may change state at a particular time, from 'on-line' to 'off-line', for instance. Alternatively, it might be necessary to record when a tuple was updated in the database by (say) the operator. This version of time is called *transaction time* and is often used in the DBMS journal which keeps a record of the changes as they are made. The third type of 'time' frequently used in databases is referred to as *user defined time* and is simply some arbitrary time such as the time (or date) that a piece of plant is scheduled for maintenance. In re-running a previously stored situation it is important to re-build the precise system that was current at the instant of interest and hence the need for this temporal data. This will be discussed further in Section 3.6.2. The requirement to store temporal data has lead to the design of

so called temporal databases [10, 82, 169, 226], however, these systems are the subject of current research and no working system seems to be available at present. In this work, a design is proposed that uses these three types of time unaided by the constructs of temporal databases and in so doing justifies, to a certain extent, the use of the term temporal database as a description of the present schema.

(Each object stored in the database which has a graphical representation is given an attribute that allows linkage with a part of the database which is used purely for the graphics program. Graphical attributes are not stored in in the database since a relational database is far too slow to hold such data. This information has been omitted from the tables presented in this thesis since it is not directly relevant to the research. Appendix 3 describes the graphics package.)

3.5.2 Schema Design

Much man-power is currently expended during distribution system modelling exercises, in the water industry, when producing a set of self consistent data values that represent the plant in a form suitable for the data processing software [68]. This exercise comprises:

1.0 Obtaining the accurate parametric values for the plant items to be modelled;
2.0 Deciding which plant items are relevant to the model and which are not. For example:
    2.1 The question of whether an item is connected to the main body of the plant;
    2.2 Whether (typically) the flow through a plant item is hydraulically significant.
    2.3 Which plant items have no bearing on the plant model (e.g. communication pipes, various fittings etc.)
3.0 The process of lumping distributed parameters such as domestic consumer demand.

While it is not possible to fully automate the first exercise (although it is possible to automatically tune given values [201–203]) the second exercise could be automated to some degree and the third can be completely automated.

In exercise 2.1, the status of the plant or of the switching valves connected to the plant will determine whether an item of plant is on-line or not. Since the status of switching valves can be stored in a suitable table it is possible to determine which plant items are hydraulically connected to the main plant simply by examining that table. The problem outlined in 2.2 is more difficult [48] and no automatic scheme is attempted here but instead the database operator can specify a parameter that determines whether or not the plant item is to form part of the mathematical model. Problem 2.3 is again operator determined but is also implicit in the database design. Note that the hydraulic effect of fittings and appurtenances is normally accounted for in the determination of the hydraulic resistance of a pipe.

The hydraulic effect of lumping demand is investigated further in Chapter 5 but the procedure can be automated straight forwardly where isolating valves are closed using the schema design itself as will be shown in Section 3.6 and Chapter 4.

This general discussion on data requirements and the distribution system modelling process has clarified the uses of the database schema. It is now possible to proceed with an optimal (in database terms) schema design and hence the general philosophy of the database design is stated as follows:

1.0 The highest degree of normalisation should be employed in the design of each table where it does not compromise the efficiency and robustness of the overall design. Normalisation is described in Appendix 1;

2.0 Referential integrity should be imposed on the set of tables but the tables should be designed so that a minimum of inter-table links or triggers is required;

3.0 The database schema should be determined by the meaning of the information stored as data and the degree of normalisation must not be constrained or compromised by any one particular intended usage;

4.0 Data duplication should not be allowed since this would lead to the possibility of inconsistency;

Objectives 1 – 3 (above) ensure that the overall design is as open as possible so that any future programs can make use of the data stored without having to redesign any of the tables. This schema design philosophy will result in a parametric description being stored in its most basic and natural form for each physical plant item relevant to the totality of the system under control. This leads to the observation that the database model will be a *logical copy of the actual physical system.* Of course, this introduces the operational constraint that the database must be kept up to date when the physical system changes in the course of normal network operations. (A special human-computer interface exists to aid the operator which is described in Appendix 3).

3.6 Schema Design Implementation

In this section the actual implementation of a schema design that follows the philosophy outlined in the previous section is presented. The basic design is in two parts: tables and views. The tables store asset data, dynamic data, past data and meta data. Views are used to provide subsets of table data or subsets of combinations of table data. A view is practically the same as a table except that it does not exist in its own right, since it is derived from other tables (or other views). This section describes the format of the tables in the database schema, however, since views are more relevant to the operation

of the database (although they are strictly part of its design), they are not discussed in this chapter and are left until Chapter 4.

The design of the base tables in the schema can be divided into three categories. These include tables for storing asset data, tables for storing past data and tables for storing meta data. However, the majority of the tables in the finished schema merely contain data which is directly applicable to the object in question and once the design philosophy explained below is understood no further detailed explanation of those tables is necessary.

3.6.1 Asset Data

To appreciate the philosophy behind this schema design it is not necessary to review each table in the database and so only a small subset of the complete set of tables will be explained in detail. For example, consider the `Pipes` relation:

Table name: Pipes	
Attribute	**Data Type**
pipe_no	integer
link_no	integer
zone_i	integer
length	real
diameter	real
chw	real
roughness	real
material	char string
min_press	real
max_press	real
max_velocity	real
no_connections	integer
population	integer
last_maint	date
sched_maint	date
os_ref	char string
name	char string
valid_time	date
trans_time	date

Table 3.1

Note that a pipe is not assigned a graph theoretical direction, ie no information is given about the two nodes which, in a mathematical model, the pipe would lie between. This is because, in reality, a number of pipes of different material or diameter may be laid end to end between two nodes. In order to store the fact that a pipe is often part of a physical system that connects two nodes, the `link_no` attribute is specified. `Pipes.link_no` is a foreign key to the primary key `Links.link_no` and is therefore

explained in the description of the attributes in the Links table. The zone_i attribute signifies an area determined by pressure controlling devices such as pumps or valves, therefore a pipe can be a member of one, and only one zone. The population attribute refers to the number of consumers distributed along the pipe. The explanation of the use of trans_time and valid_time attributes is left until Section 3.6.2. Additional, benign data such as ordnance survey reference, a descriptive name, the post code of the street and the pipe's maintenance history (as recommended in [52]) is also provided as well as the parametric data which itself does not need any further explanation.

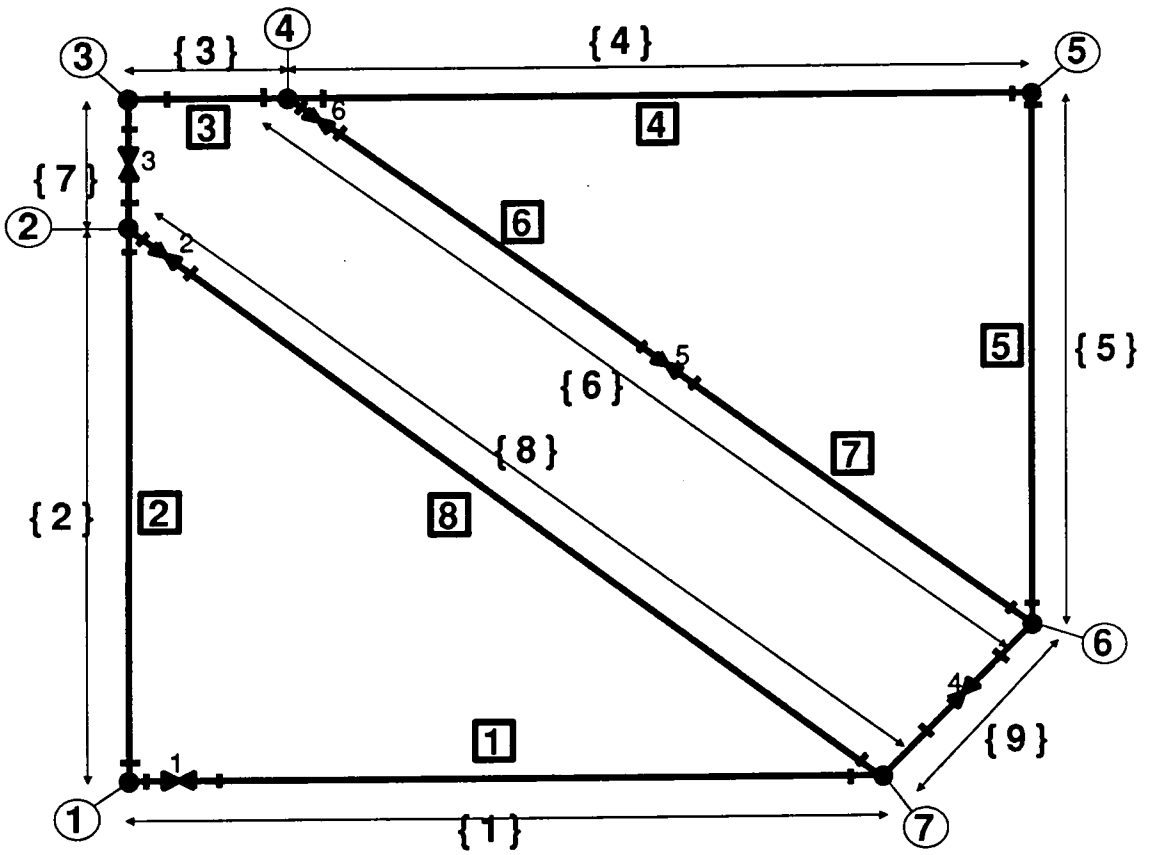
The Links table (below) is an example of topological data:

Table name: Links	
Attribute	**Data Type**
link_no	integer
node_i	integer
node_f	integer
trans_time	date
valid_time	date
type	char string

Table 3.2

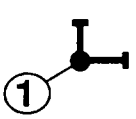
A link is a path between any two adjacent nodes in the network, which, in the mathematical model will form one equation relating head difference to flow. In the database, it is given a designation depending on the most hydraulically significant element which it contains. Pumps and PRV's are equally significant so that if a pump and a PRV are both present, they should be modelled separately. The possible values that can be assigned to Links.type are:- 'pump', 'valve', 'pipe' or 'node'. Pipes are always part of pumps and valves so the first two designations imply that pipes exist which are

Example Network



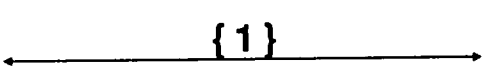
Key:

Pipe and pipe number



Node and node number

Valve and valve number



Link and link number

Figure 3.3

connected to either the pump or the valve and which share the same `link_no` but which are not modelled separately.

The designation 'node' is crucial to the present schema design. A link can be assigned the type 'node' by the user of the system to signify that it is part of a node cluster normally assumed to operate at one pressure. In other words, the 'node' designation is applied whenever the link is considered not to contain any hydraulically significant elements. Fig. 3.3 has two such clusters of nodes, they are numbers 2,3,4 and 6,7, with link numbers 3,7 and 9. Even though links 7 and 9 contain switch valves they are still designated 'nodes'. When the valves are open, the links are used to combine the nodes in the cluster, whereas if they are shut, they prevent the nodes in the cluster to which they are joined from being combined. This is the key to topology determination as will be shown in Chapter 4.

The basis for demand allocation and topology determination is provided in the following table. This is the `Isolating Valves` table. The benign data in the actual `Isolating Valves` table has been omitted for clarity:

Table name:	`Isolating Valves`
Attribute	**Data Type**
`valve_no`	`integer`
`link_no`	`integer`
`load_i`	`integer`
`load_f`	`integer`
`status`	`char string`
⋮	⋮

Table 3.3

Every element in the network through which water flows is given a link number. Since a valve, like a pipe, is a member of a link, it is given a `link_no` attribute. The loads refer to the fraction of the population (in the link) on each side of the valve. This effectively conveys information as to where in the link each valve is situated. The use of this information is explained further in Chapter 4.

Details of consumer demand is held in a table which relates the population in a given area to particular pipe. The `Pipes.population` attribute is a summation of this information for each pipe in the table:

Table name:	`Demandnodes`
Attribute	**Data Type**
`node_no`	`integer`
`zone_no`	`integer`
`elevation`	`real`
`min_press`	`real`
`population`	`integer`
⋮	⋮
`os_ref`	`char string`
`name`	`char string`
`valid_time`	`date`
`trans_time`	`date`

Table 3.4

In this table, `node_no` is the primary key. The purpose of the table is give meaning to the actuality of a node. Note there are no link numbers, since the primary key is enough to connect with the `Links` table. The most important information here perhaps is `elevation`, since this is used together with any pressure measurement that might be related to the tuple in question.

3.6.2 Past Data

One consideration that appears to be ignored by other network modelling systems is the design of a well defined method for dealing with past data. In order to handle past data in a coherent way, the history of network changes must be recorded in the database so that past events can be analysed as they occurred. This requires that any changes of state to any of the current network objects, any addition of new objects or any deletion of old objects must be recorded in the database tables together with the time that the change occurred.

At this point it is necessary to understand the usage of the `trans_time` (transaction time) and `valid_time` attributes since they occur in all the tables. Transaction time refers to the age of each tuple in the database. It can be used to describe when an event occurred in the database itself and can therefore be used to determine the evolution of the database. This is in contrast to valid time which refers to the temporal nature of the information that the database holds. For example, if a pump has been started at some time in the past but somehow this fact had not been automatically entered into the database, then the model must be updated accordingly such that when an analysis is carried out on the past state of the network, the correct pump status is automatically included. To include the pump status correctly it would be necessary to search the database for the state of the network which was valid at a particular time. To use transaction time for this search would lead to an error since the valid time for the pump start would, in this case, precede the transaction time. The same obviously applies to other types of change to the physical network that are not automatically relayed back to the central control computer.

Furthermore, valid time is only relevant to parametric data and not benign data because, to reiterate, valid time refers to the state of the physical system. Such attributes as `Pipes.sched_maint` (the scheduled maintenance date for a pipe) do not affect

the state of the pipe and hence no change to the valid_time attribute would be made following a change to this attribute, although the transaction time would be altered.

Hence, each asset data table contains the trans_time and valid_time attributes. For static data tables (descriptions of physical objects) the attributes of the second table are duplicates of the first table except that there is an extra column attribute called change. This attribute refers to one of two possible events in the current data table: either an update or a deletion.

The past data tables store the old tuples together with the type of change that occurred and the time (transaction time) of this change. Valid time is simply the old valid time (the new valid time is held in the current data table). The sequence of changes that cause the updating of the past data tables is accomplished automatically within the database by the use of *triggers*. A trigger is specified at the time of the database definition and instructs the DBMS to execute a predefined alteration to data in a table following some change to a related table. In this case, every time an alteration is made to a static data table, a subsequent addition (via the SQL insert operator is made to the related past data table.

A trigger is created at the time the schema is created and is normally employed to ensure referential integrity. The following is an example of the creation of a trigger. The trigger is given a name: 'move_deletes_on_demand_nodes' and it can be instantiated so that it is carried out before or after the transaction named has taken place. In this example the named transaction is delete and the source table is demandnodes while the receiving table is the past data version of that table, namely past_demandnodes. The action calls for the DBMS to insert all the stored information about a tuple, referenced by the primary key, in to the past_demandnodes table before a deletion takes place. To do this the DBMS creates a temporary table which has been called d and which is known as an alias for the demandnodes table.

```
create trigger move_deletes_on_demandnodes
before delete on demandnodes
    (insert into past_demandnodes
        (node_no,
        zone_no,
        elevation,
            ⋮
        valid_time,
        trans_time)
    (select * from demandnodes d
    where
    d.node_no=demandnodes.node_no))
for each row
    (update pastdata set change=''delete''
    where
    past_demandnodes.node_no=demandnode.node_no)
for each row;
```

With dynamic data, large sets of data are inserted into the database at once and because of the need to allow this to occur quickly, additional past data tables are not used and the `update` transaction is not permitted. This approach allows faster operation because it avoids the need to wait while a large number of triggered updates are executed to ensure that the past data tables are correctly updated. Taking measurements as an example, it can easily be seen that it is not desirable to allow use of the SQL `update` operation. The dynamic past data tables are therefore not mirrored but instead, whenever new data is available it is inserted into the table together with its valid time. The retrieval operation defaults to selecting the most recent set of measurements from this

table. Queries on past data can easily be made by defining a 'most recent since ...' qualifier. Additionally, it will occasionally be necessary to issue a `delete` instruction to prune the table to prevent the database from becoming too large. Valid time is available directly from the measurements provided they are 'time-tagged' in the field outstations before being sent to the SCADA computer, otherwise the valid time can simply be derived from the database computer's own clock. With dynamic data, it is unlikely that transaction time and valid time will differ much from each other but in this work it is proposed that both be stored as the difference gives an indication of the degree of dynamic data skew.

3.6.3 Meta Data

In not relying on an invariant mathematical model of the distribution network, this proposed system faces the inevitable overhead of checking the database tables at each new re-calculation (or re-estimation) to ensure that the model is up to date. However, this additional overhead can be minimised using a table of meta data designed to allow each routine to determine whether any changes relevant to it have occurred in the database during the previous time period.

The `Change` table comprises a column for the module name and a set of further columns for each table in the database. The `module` column contains the name of each module that uses the database. Therefore each routine has exactly one row to inspect at each new time step, furthermore, if a new module is added to the software suite, it is simply necessary to insert a new row into the `Change` table. The other columns contain the transaction time when the table referred to by the column name was last updated. The analysis routine need only initiate a select transaction on a table if a change has been detected. If benign data is changed in a static data table, this does not affect the transaction times held by the `Change` table. (Although the database journal contains a

record of the recent changes made to the database, the user does not have control over it and so it is of limited use.)

The Change table can be kept up to date automatically using triggers that update the table name attributes with a new transaction time whenever an alteration is made to a base table.

Table name: Change	
Attribute	**Data Type**
module	char string
pipes	date
links	date
⋮	⋮
⋮	⋮

Table 3.5

3.7 Optimisation

An advantage with using a relational data base system is that optimisation can be carried out at a high level. There are two aspects to this problem: query optimisation and storage optimisation. The DBMS carries out its own optimisation internally to some extent, normally well enough that the user will not do better manually. However, since in some circumstances, it is possible to know in advance the type of query most frequently used or the most time-critical query, then one can employ some of the constructs to improve the performance of the completed schema.

An important consideration often overlooked in this area is the resources that the computer has at its disposal. In modern multi-user operating systems the operating system dynamically allocates its resources such as CPU time, RAM and cache to the various tasks

it is servicing at any one time. Each task belongs to a user who has various restricted allocations made to him/her when the account is set up. Furthermore, a system process may have artificial restrictions imposed upon it during the creation of its own account. Lastly, and perhaps most significantly, when the computer system was initially configured, the tasks to be run on it were probably not known and therefore the area on the disk that the operating system uses for swapping processes in and out of its RAM, may not have been correctly sized. In a large database application such as this, large amounts of data are manipulated in memory, possibly involving the creation of temporary tables. It is therefore crucial that the maximum resources are provided, including the disk swap area, to ensure good performance from the database itself. It is the author's experience that this can have a far greater impact on efficiency than any of the tools provided within the DBMS.

3.8 Database Access and Data Control

3.8.1 Database Access Library

In order that pre-existing software is able to exploit the database environment described here, the access routines are entirely self contained. This minimises the amount of alteration necessary to any existing code. To achieve this goal in the current work, a number of FORTRAN subroutines were written which call SQL modules which in turn access the database. This set of subroutines and modules constitutes the 'Database Access Library' (DBAL).

3.8.2 Control of On-Line Data

The problem of dealing with on-line data which is quickly time variant needs to be considered carefully since it is crucial to the efficient operation of the data management system. As can be seen from the test results shown in Section 3.9, the database imposes

a serious overhead in terms of access and retrieval times. In a system designed for the control room, it would be undesirable to have to wait until all the measurement data, estimation data, and demand data had been written to the database before it was available to the operator. This delay can be expressed as follows:

Let the time at which the measurements become available be t_0, then the estimates will only be available at time

$$t_0 + \Delta t$$

where

$$\Delta t = \sum t_{db}, t_{est}$$

and,

$$t_{db} = \sum t_{db\_get\_model}, t_{db\_get\_meas}, \text{etc} \ldots$$

$$t_{est} = \sum t_{build\_model}, t_{converge}, t_{write\_data}$$

Where $t_{db\_get\_model}$ is the sum of all the stages necessary to retrieve the latest model from the database and $t_{db\_get\_meas}$ refers to the retrieval of the latest measurements from the database.

An alternative is to place dynamic data into the database but make it directly available to the analysis modules at the same time through the use of task common blocks (in the case of FORTRAN). In order maintain data consistency, each measurement has to be time stamped and placed in the database using `insert` where the measurement's tag number forms a composite primary key together with it s valid time data. The `insert` operation results in a complete history of measurements being stored as explained in Section 3.7.

3.9 Functional and Performance Analysis

A number of tests were carried out with the aid of a specially designed program that stores duplicates of a given network in the database. The components of the duplicate networks are numbered in a way that guarantees uniqueness across the whole set of networks thus formed. Although the resulting network is non-connected, the numbering scheme allows it to be treated as one large network without loss of generality.

To investigate the performance of the data retrieval and storage procedures most effectively, it was necessary to only transfer the data essential for the model building exercise. When the simulator or state estimator are required to run, it will not be necessary to transfer all of the data stored against the relevant objects in the database. Such data fields that are not required include place names, maintenance dates, material etc. (i.e. benign data). In fact only parametric data is required. One could say that transactions executed on behalf of the on-line (or quasi on-line) programs are performance critical transactions whereas those carried out for the purposes of ad hoc queries are not performance critical. For this reason the performance tests conducted measured in detail the timings of performance critical transactions only.

The tests were undertaken for a variety of database transactions. A program was written that read the network data from a file and loaded it into the database and then read the data from the database. The subroutines used to make the transactions were:

`DB_ADD_PIPES`: To add a set of pipe parameters

`DB_ADD_LINKS`: To add a set of links

`DB_ADD_VALVES`: To add a set of isolating valves

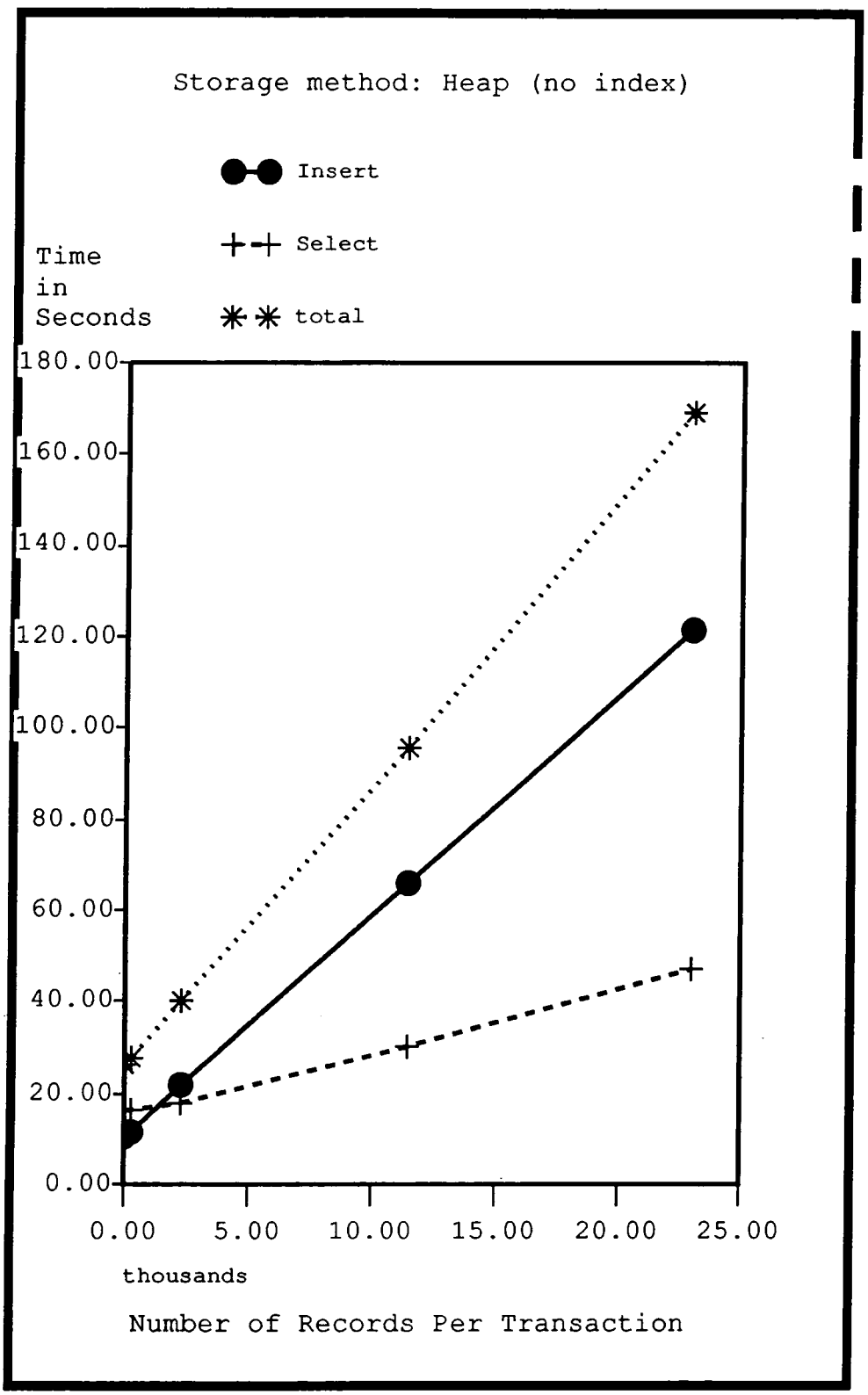

Storage method: Heap (no index)
Insert
Select
total
Time
in
Seconds
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
0.00
5.00
10.00
15.00
20.00
25.00
thousands
Number of Records Per Transaction

Figure 3.4

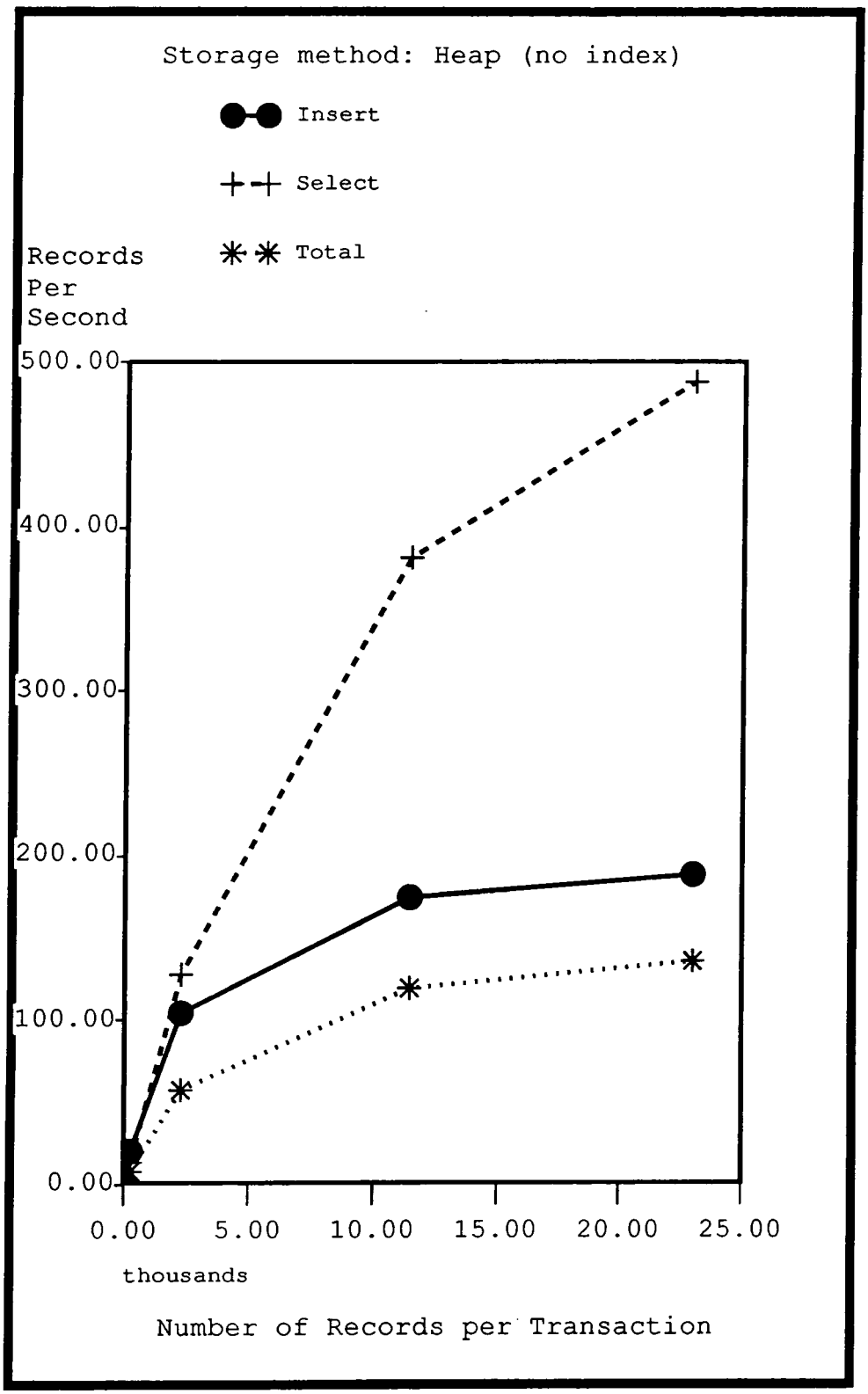

Storage method: Heap (no index)
Insert
Select
Total
Records
Per
Second
500.00
400.00
300.00
200.00
100.00
0.00
0.00
5.00
10.00
15.00
20.00
25.00
thousands
Number of Records per Transaction

Figure 3.5

DB_GET_PIPES: To retrieve all information from the `pipes` relation

DB_GET_LINKS: To retrieve all information from the `links` relation

DB_GET_VALVES: To retrieve all information from the `valves` relation

Each _ADD_ subroutine is designed to place data in a database table and each _GET_ subroutine retrieves data from a table.

The test were conducted on a VAX-station 3100, Model 30 computer with 8 Mbytes of RAM. The operating system, VMS-5.2, dynamically allocates memory and cache. While it is difficult to determine precisely what is going on, VMS appears to give up its ownership of the memory during heavy processing but when that processing has finished, VMS accounts for about 5–6 Mbytes of RAM.

Each test was conducted as follows: first of all the largest 'network' (comprising 1000 duplicates of Fig 3.3) was loaded into the database using `insert` operations. Secondly, that data was retrieved using `select` operations. This was done four times but the elapsed times of the first pass were ignored for reasons explained below. For each of the other three passes the elapsed time of the total insert operations and those of the total retrieve operations were averaged to give two timing results per pass (corresponding to the solid and the dashed lines respectively in Fig 3.4 & 3.5). Then smaller data sets were loaded and retrieved in a similar fashion (with three passes each, all averaged) until the last pass, which was for the singleton network case.

The reason it was done in this order was due to the way VMS allocated resources. Starting from small size transactions and moving to bigger ones, VMS appeared to be recalculating the needs of the task repeatedly, and the results were not consistent. By starting with a large size network VMS allocated judicious amounts of resource and

performance did not deteriorate in the way it did by going from small to large. The very first pass was ignored because the elapsed time here included the initial loading of the database image and the attachment overhead to journal files etc. Since we are considering this application for on-line use, the start-up overhead is irrelevant.

The results show that there is a nearly linear relationship between the number of records processed and elapsed time. Since 1000 copies of the network shown in Fig. 3.3 represent a 7000 node network, it was felt that this was a reasonable upper limit. At the low end, the overhead from the database can clearly be identified by the fact that the curves do not pass through zero. However, performance increases with number of transactions as can be seen by observing the record processing rate as shown in Fig 3.4.

3.10 Conclusions

This chapter has presented a new approach for the data handling functions necessary for an on-line monitoring system. The philosophy is general, in that it need not specifically apply to water networks and is directly applicable to other networks such as gas or electrical systems. The approach is also generally applicable to other large scale monitoring operations.

The overall goal has been to represent data as it really is, without making concessions to the way it might be used. The only major difference between the schema designed in this work and a likely schema for a company asset data base is the inclusion of the graphics attribute in certain tables and the inclusion of the `link_no` attribute in others. This point is important since it means that without much modification, an asset database could be used as the database underpinning network analysis. Although this itself is useful, the real benefit is that since it is the company's assets that are being monitored, it is crucial that the data used in the monitoring algorithm is consistant with

the data held in the asset database. With this approach the two processes can be directly joined.

The approach also provides a systematic procedure for the modification of the data stored so that data consistency is achieved at all times. Linking this data to other applications such as graphics has already been done thus proving that the database can be used as a single repository supporting multi-task access. This removes the problem of data inconsistency and allows the operator to manipulate the database via a meaningful graphics screen. Once the operator is involved in the data upkeep in this way, it is likely that the data held in the database will be better maintained.

This data maintenance can now be carried out by personnel who have no knowledge of the way the analytical routines work and all that is required is that they maintain their logical copy of the physical system. As has been mentioned before, a human-computer interface has been designed in other work to make the data maintenance task easier. It is not expected that any user of this type of system would interact directly with the database, instead all the interaction would be via the H.C.I. which calls a set of subroutines, some of which have been tested above.

The performance tests show that there is a fairly severe overhead imposed by the use of the database in terms of the transaction times. This is an absolute increase over the common block method mentioned earlier because data transfer via common blocks is instantaneous. However, common blocks have to be loaded from files and every time a change occurs in the network the data files would have to be edited and re-loaded. Of course it is not possible to compare the edit time against the time it would take to make a comparable change to the database but since in editing files one would need to adhere to the strict guide-lines imposed by the particular analysis routine, it is quite likely that the database approach will actually speed up the overall data maintenance task even for a network analyst. For an operator, without direct experience of loading data files, the

speed-up offered by the proposed approach would be much greater. Furthermore, the database will not suffer the mental fatigue of the human when it comes to 7000 node systems and so transcription or typing errors would be reduced.

CHAPTER 4

AUTOMATED NETWORK MODEL DERIVATION

4.1 Introduction

The method of state estimation outlined in Chapter 6 of this work requires a mathematical model of the water distribution network. The network itself comprises a wide variety of physical objects which include pipes, pumps, valves and reservoirs. The data necessary to describe this collection of objects is not limited to the objects' physical description but must also include a description of the interconnection of these objects. Facts about some objects change the interconnection information directly and these changes occur in normal network operation. Since it is the physical system that is of interest to the network operator rather than some mathematical abstraction of it, the information provided must always relate directly to the physical system and any operator action must therefore be in terms of the physical system rather than the mathematical model.

An on-line monitoring scheme with a state estimator providing the important data validation function must have a mechanism that ensures the mathematical model employed represents the physical system as accurately as possible, at all times, otherwise the processed measurement data will be meaningless. Exactly the same argument applies to any other analytical routine that relies upon the mathematical model. Although a description of all the elements of a network stored in a database might be complete in

so far as every physical item is completely described, such data is not guaranteed to be in a form that can be used directly by the analysis software. It is therefore necessary to consider ways in which the data held in the on-line database can be used to automatically build a mathematical model suitable for network analysis.

This chapter is presented in 7 sections. Section 4.2 reviews the approaches taken in power system topology determination since this is closely analogous to one of the main problems confronted by the present chapter. Section 4.3 introduces the problem of automatic model generation more formally and outlines the two main aspects of the problem. Section 4.4 presents new techniques which exploit the relational database of Chapter 3 to solve the problems posed in Section 4.3. The complete model building phase is then described in Section 4.5. In Section 4.6 tests are undertaken to show that the new algorithms are functionally correct and to determine the performance overhead imposed by the routines for various network models. The chapter is summarised and conclusions are drawn in Section 4.7.

4.2 Review of Previous Work

The problem of topology determination at the nodal level is similar to the problem of determining the topology of a substation in an electrical power system network. A water network normally has a number of groups of nodes each treated as single demand nodes for mathematical modelling purposes comprising a number of pipes joined together on a manifold containing isolating valves, as shown in Fig. 4.1 (from [229]) and schematically in Fig. 4.2 in which the range of available topologies is illustrated. This structure is termed a 'node cluster' in this work and is somewhat analogous to a power system substation. A power system, on the other hand, comprises many substations within which different transmission lines are connected together via bus bars and switches. Since topology determination in power systems is so similar to the problem

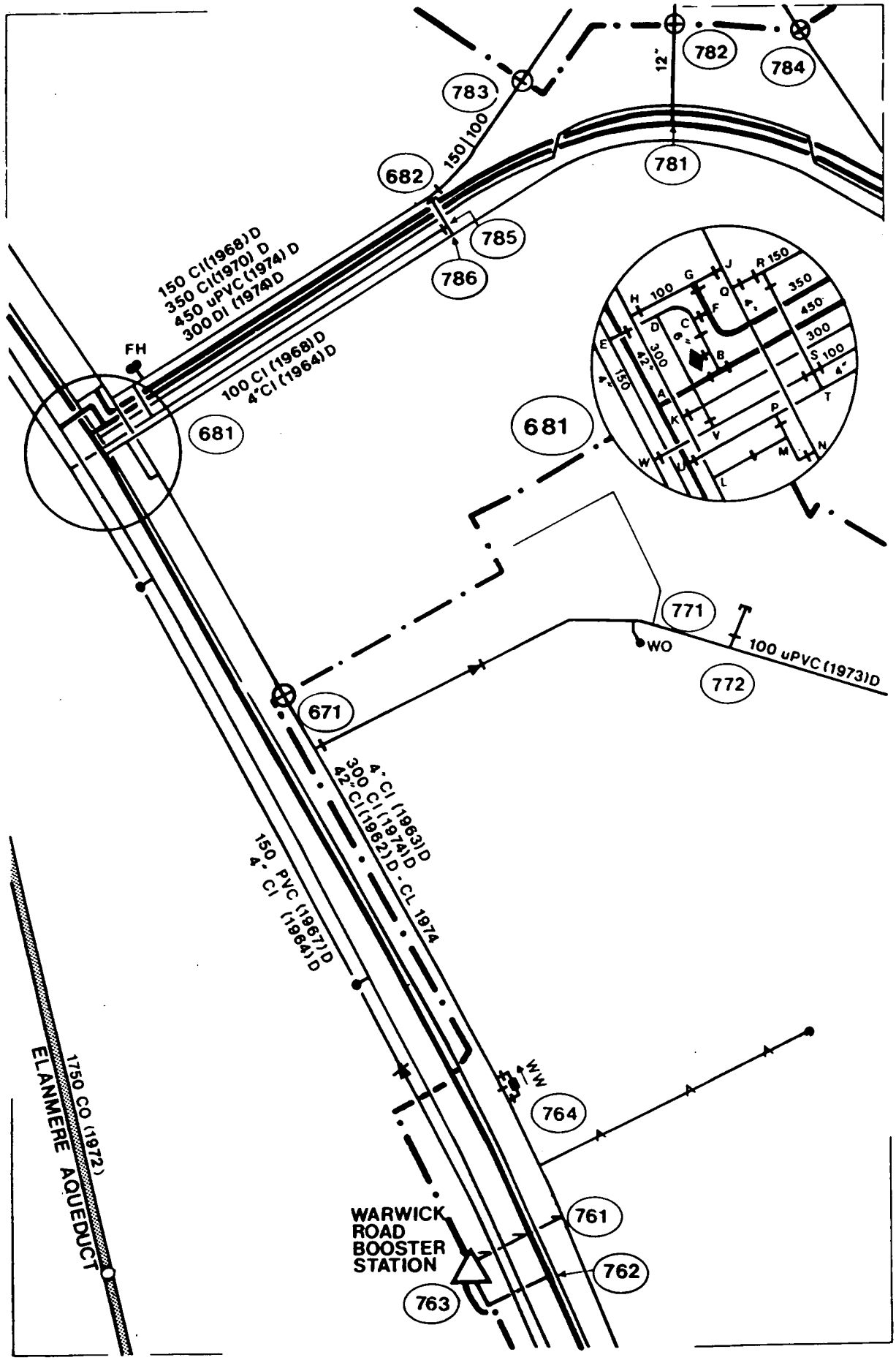
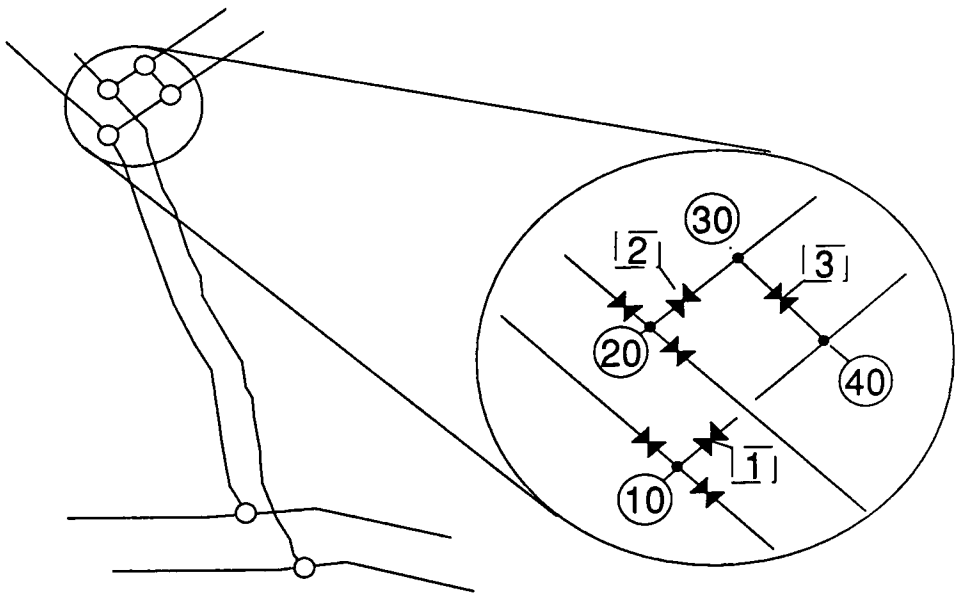

783
782
784
12"
781
682
150 100
785
786
150 CI(1968)D
350 CI(1970) D
450 uPVC(1974) D
300 DI (1974)D
FH
100 CI (1968)D
4"CI (1964)D
681
681
771
WO
100 uPVC (1973)D
772
671
4" CI (1963)D
300 CI (1974)D
42" CI (1962) D - CL 1974
150 PVC (1967)D
4" CI (1984)D
1750 CO (1972)
ELANMERE AQUEDUCT
WW
764
WARWICK
ROAD
BOOSTER
STATION
761
762
763

Figure 4.1

Example of Topology Modification



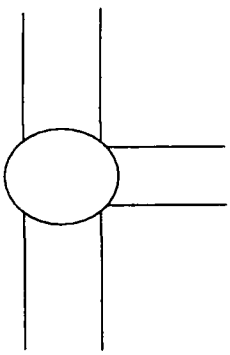
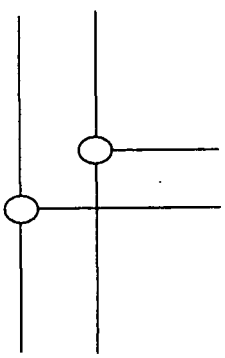
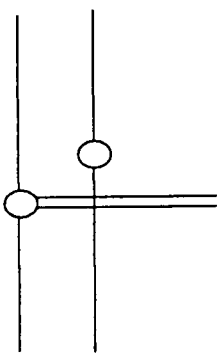
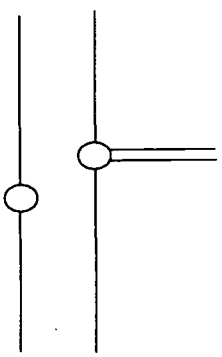
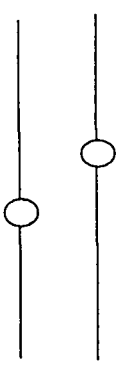
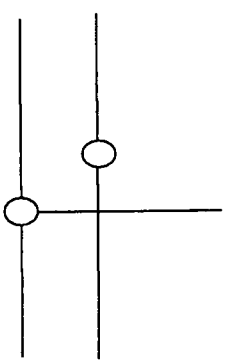
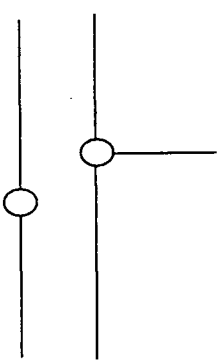
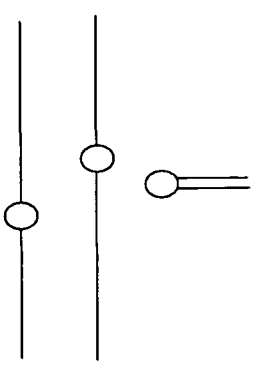
1,2,3 open	1,2 open, 3 closed	1,3 open, 2 closed	2,3 open, 1 closed
			
10 --- 40 20 --- 30 30 --- 40	10 --- 40 20 --- 30	10 --- 40 30 --- 40	20 --- 30 30 --- 40
1,2,3 closed	**1 open, 2,3 closed**	**2 open, 1,3 closed**	**3 open, 1,2 closed**
			
	10 --- 40	20 --- 30	30 --- 40

Figure 4.2

of determining water network topology, a number of papers on the subject of automatic topology determination for electrical power systems are discussed below.

Topology determination algorithms can be categorised depending on the description of the input data. These data can either be branch numbers or node numbers or both. Nagdy and Rumpel [182] further split these classes into sub-classes that range from complete description (node and branch); to branch oriented, node referenced; branch only (directed); branch only (non-directed); node oriented, branch referenced; node and adjacent nodes only. They also introduce the useful concepts of potential topology (PT) and actual topology (AT). Potential topology is effectively the topology of the system with all links intact whereas the actual topology is the topology that must be derived for the model. In other words, AT can be derived from the PT when the breaker states (isolating valve states in the water network analogy) are known.

The problem of general topology determination is in fact composed of two sub-problems, which in this work are called *specific topology determination* and *island determination.* Specific topology refers to determining the list of plant items that are hydraulically connected for a given network whereas island determination refers to the problem of identifying and labeling whole groups of plant that are hydraulically connected within the group but which are not connected to other groups of hydraulically connected plant.

Sasson et. al. [213] present a general topology analysis method based on a branch only description which comprises three algorithms dealing with both off-line and on-line data. Their approach is designed to deal with a number of problems associated with automatically determining the network model:

1.) Substation switch status processing;
2.) Measurement status determination;
3.) Node (substation) splitting;

4.) Islanding;

5.) Automatic model updating.

Their first algorithm deals with basic network information and reduces the physical system into two data files, one of which contains breaker status information where each breaker is identified by the circuit elements it connects to and the second holds data that relates to the availability of each line for inclusion into the model which is dependent on whether or not there exists a suitable measurement for that line. The second algorithm deals with any changes that take place in network operation and therefore adds on-line data to these two tables. The topology determination process is then carried out in a further module that searches through the list of circuits finding closed paths connected via closed breakers. The lists are passed on to a further module which updates the configuration table.

Sullivan et. al. [247] present a simple topology determination technique which was developed specifically with substation processing in mind and consequently is only applicable to power systems. However the simplicity of their approach and the fact that it is based on tables of plant data is such that it is worth considering here. Their method uses tables of data containing configuration details such as node/injection data and circuit breaker data. By processing the status of the circuit breakers in the appropriate tables the algorithm produces intermediate vectors that contain the branch interconnections for each substation and the node interconnections. These are called the BIV and NIV vectors respectively. Lastly, the system as a whole is processed to determine the SIV or system interconnection vector, which holds a set of numbers corresponding to the islands present in the system.

Their method is rather specific to power systems and relies on a substation numbering scheme although it does not impose any restriction on node or substation ordering. In

water systems, where there is no 'substation' number and where the boundaries of node clusters are ill defined, the algorithm would not be applicable without modification.

Dy Liacco et al [102], present a discussion paper on a network status processing system which attempts to determine topology for state estimation in the power industry. The method introduces the concept of 'segments' meaning the length of circuit between two switches. The topology algorithm then consists of a search of all the segments, recording those which are connected, energised or earthed. The exact details of the algorithm are not given in the paper and it is not clear how the input and output data formats interface with other software.

A node number only method proposed in [127] uses the binary connectivity matrix to determine the number of islands in a network. The connectivity matrix is square and symmetrical with 1's on the diagonal and a 1 in position a_{ij} if node i is connected to node j. Zeros are placed in all other positions. The matrix thus formed exhibits first order connectivity in that it determines only the node pairs that are directly connected. If the matrix is multiplied by itself, using Boolean multiplication, the result is a matrix that exhibits second order connectivity. That is, if node i is connected to node j but not to node k, and node j is connected to node k then the elements a_{ij}, a_{jk} and a_{ik} will all be set to one. Multiplying the matrix $N-2$ times for an N node network will guarantee that the resulting matrix is full if and only if there are no islands. However, if after this many multiplications, zeros remain in the matrix then any complete set of linearly dependant rows (or columns) represents an island whose nodes are determined by the row (or column) numbers.

Boolean multiplication is achieved in much the same way as conventional matrix multiplication except that every $\times$ operator is replaced by an $\wedge$ (AND) function and every $+$ is replaced by an $\vee$ (OR). A binary matrix A is multiplied by itself for the p^{th}

time as follows (remembering that $A^T = A$):

$$A^{(p+1)} = A^{(p)} \wedge A^{(p)}$$

$$a_{ij}^{(p+1)} = \bigvee_{k=1}^{N} a_{ik}^{(p)} \wedge a_{kj}^{(p)}$$

$$\text{for } i = 1, \ldots N$$

$$\text{for } j = 1, \ldots N$$

This is best illustrated by a simple example. Consider a sequence of 4 pipes connected in series starting at node 1 and ending at node 5. The binary connectivity matrix would therefore be:

$$A^{(1)} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

and after one operation

$$A^{(p+1)} = A^{(p)} \wedge A^{(p)}$$

we get:

$$A^{(2)} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

and following a further operation the resultant matrix becomes full as expected. The second order connectivity can be seen in the $A^{(2)}$ matrix and when this operation is repeated the next matrix will exhibit third order connectivity which, in this example, would cover the whole system.

The authors of this paper present some computationally efficient schemes for processing the connectivity matrix row or column-wise rather than carrying out full multiplications and schemes which exploit sparsity are also presented. A simple but not

very serious drawback of their method is that it relies on a monotonic node numbering sequence. This means that nodes may have to be re-numbered prior to the connectivity matrix being formed which is a wasted operation since it is likely that further re-ordering will be required when the topology has been fully determined and before the resulting network equations are solved. Efficient ordering schemes for network solutions are given in [249, 250]

4.3 Definition of Problems Associated with Model Derivation

Assuming that the data held in the database is complete in terms of the data requirements for the modelling and analysis functions, it is now necessary to consider what steps need to be taken to transform this data so that the mathematical model can be built automatically. There are two main aspects of automatic model generation:

1. General Topology Determination:
   1.1 Determining node cluster ('substation') connectivity;
   1.2 Determining network wide connectivity and islands.
2. Aggregation of Model Parameters:
   2.1 Correct allocation of lumped consumer demand;
   2.2 Combining series and parallel elements as necessary.

In this section, the above problems will be posed in greater detail and new techniques for their solution will be presented and tested in Section 4.4.

4.3.1 The Network Topology Determination Problem

In practically every pipe there are a number of isolating (or sluice) valves, closure of any one of which can alter the network topology. Unfortunately, the status of these valves is practically never monitored on-line although their state may change reasonably

frequently in the course of network operation. This clearly presents a problem for any method which relies heavily on an accurate description of the network topology and where there is no remote monitoring of status, the only solution is to rely on the operator to enter the valve changes into the computer model as they occur in the physical system. This information must be used to keep the database up to date as the network changes.

Before the solution of the topology determination problem can be found, it is necessary to clarify what is meant by a 'node'. If a graph theoretic approach were taken, and all pipes and pumps etc. were treated as edges of the graph (however short the pipe), then every junction between two or more edges (ie a vertex on the graph) would constitute a node in the model. However, if that approach were taken, the mathematical model would become larger and more detailed than necessary. This degrades the model for three reasons:

1. It increases the complexity of the model without necessarily providing more useful information;
2. It decreases the stability of the analysis due to the increase in the range of sensitivities of flows to changes in pressures thus effectively increasing the ratios of the maximum and minimum singular values of the Jacobian matrix and consequently worsening the condition number;
3. It increases the amount of data that would be presented to the operator, thus adding clutter to the screen or, alternatively, increasing the data reduction task.

There exists, therefore, good reasons for wishing to reduce the complexity of the model somewhat during the topology determination process and to only determine the topology essential to the analytical routines and, ultimately, to the operator.

Due to the formal approach needed for the design of relational calculus expressions, the exact definitions of the terms node, link and node cluster are needed:

1 **Node.** A node is the intersection of two or more links;

2 **Link.** A link is a topological entity that connects exactly two nodes. In the database, they can be designated in two ways, viz:

2.1 Type 'pipe', 'pump' or 'valve' which signifies that the link will form the basis of a unique equation in the mathematical model relating pressure difference to flow;

2.2 Type 'node', which means that the elements belonging to that link are not to be modelled in the way described in 2.1. If a closed valve is present, the nodes at either end of the link are considered not connected. If no valve or an open valve is present it is considered that there exists zero head drop in the link;

3 **Node cluster.** A node cluster is a group of nodes connected by links which are designated to be of type 'node' and which do not possess a closed valve or valves;

4 **Island.** An island is a subnetwork comprising all node clusters (of one or more nodes) which can be connected across links of type $\neq$ 'node' for which all switch valves belonging to those links (if they exist) are open.

Nagdy and Rumpel [182] define potential topology as the set of all links and nodes that can be connected in some way (the topology that would result if all breakers etc were closed), this is written as T_p below. When the status of all the valves in the network is known one can determine T_a, the actual topology.

Let ...

$L \triangleq$ Set of links

$L^{no} \triangleq$ Set of link numbers

$V \triangleq$ Set of valves

$V^{l.no} \triangleq$ Set of link numbers to be found in the set of valves

Each $L_i \in L$ has attributes that define its end nodes, type etc. as explained earlier. Since each L_i refers to the i^{th} link, no information other than that in L is necessary to determine T_p, in fact:

$$T_p \equiv L$$

Since certain links may contain one or more valves which can be either open or shut, the determination of actual topology involves the set of valves V, or more precisely, the set of closed valves CV.

Note that

$$V^{l.no} \subseteq L^{no} \tag{4.1}$$

Let

$$CV^{l.no} = (V^{l.no} \mid V^{status} = \text{'closed'}\ ) \tag{4.2}$$

Then

$$T_a = (L \mid L^{no} \cap (L^{no} - CV^{l.no})) \tag{4.3}$$

This produces the set of links that either do not possess valves or whose valves are open hence, T_a corresponds to the actual topology of the system but it also includes all the detail of the node cluster interconnections. Since it is not desirable to retrieve the node and link numbers that are members of a node cluster, an additional constraint is imposed on the definition. The result is termed the reduced actual topology T_{a_r}:
Let ML represent the set of links that are not in node clusters, ie.:

$$ML = (L \mid L^{type} \neq \text{'node'}\ ) \subseteq L \tag{4.4}$$

then

$$T_{a_r} = ML \cap T_a \tag{4.5}$$

Although T_{a_r} is the set of links whose valves are open and which are not members of a node cluster, this set does not reflect the complete topology of the system since the internal connectivities of the node clusters have been omitted. It does, however, provide useful information for the purpose of building the mathematical model and the missing information is contained in:

$$T_c = ML' \cap T_a \tag{4.6}$$

In Section 4.4 the proposed topology determination method produces a reduced topology by treating nodes that are connected over node clusters as being essentially one node. The combined node is given a unique alias number.

N.B. It is of course necessary to identify those islands connected to pressure generating devices such as reservoirs and pumps. Any islands that are not connected to both sources and sinks (ie demand) will have no flow through them and therefore are not of interest to the model. Although this identification of 'active' islands does not strictly constitute topology determination in the graph theoretical sense, it is considered here to be part of the island determination process.

4.3.2 Aggregation of Model Elements

There are two main aspects to this general problem. The combination of flow carrying, non-zero impedance network elements and the aggregation of distributed consumer demands to the demand nodes in the network.

4.3.2.1 Combination of Links

This problem arises in two cases: Firstly, wherever there is more than one element in a link, the head/flow relationship must be determined such that the individual elements

are combined in series. This does make the assumption that the head/flow relationship is the same whatever the order of the elements in the link, a necessary approximation which is only unacceptable in the case were there are a mixture of element types in the link e.g. a pump and a pipe with a demand distributed along the pipe. For this reason a constraint is imposed on the database which restricts any set of elements that share the same link number to be of the same type.

The second case is where, following a reduced topology determination procedure, two or more links are identified as having the same beginning and end node numbers. In other words the elements are connected in parallel. There are two approaches commonly used by researchers for this problem. The first is to devise an equivalent head/flow formula using an equivalent parameter [146]; the second is to introduce additional node numbers into all the parallel elements bar one thus effectively removing the parallel condition.

4.3.2.2 Aggregation of Demand

Although domestic demand in water systems is usually distributed along the pipes in the network, it is much more convenient computationally to treat demands as lumped parameters allocated to the most appropriate node and of course, not realistic to try to model demands on a per house basis. The allocation of distributed demand to nearby nodes gives rise to the term 'demand nodes'. Unfortunately, the proliferation of isolating valves makes it somewhat difficult to determine the actual demand value that should be lumped at the demand nodes since the valves determine the allocation of consumer demand as seen by each node.

4.4 Solutions to the Topology Determination and Parameter Aggregation Problems

In the previous section the steps involved in determining a mathematical model were analysed from the point of view of the data required for the task. Since the database

has been designed to hold asset data rather than the data specifically required for the mathematical model, it is necessary to devise procedures that can obtain this abstraction of the asset data in the form that the analytical routines can use. The methods presented below make use of the database *view*, a construct which is explained in Section 4.4.1. In Section 4.4.2, a method which solves the topology problem (and the island determination problem) is presented which exploits certain features purposely built into the database schema. Section 4.4.3 presents a method for automatically keeping track of consumer demand as isolating valves in the system change state.

4.4.1 Views

Database views are required when subsets of data from one or a number of tables provide one overall piece of information. If tables were used to store this data it would result in duplication and the possibility of inconsistency. Furthermore, since views are constructed out of tables, when data in the tables change, the data in the views change immediately without the need to construct triggers. Views greatly facilitate the determination of network topology as will be explained below, however, they do have certain draw backs, not least of which is the fact that obtaining data from views with a complicated construction can be time inefficient. A further problem is that complicated views often involve many tables in their construction and hence impose structural constraints on those tables.

An example of a subset of data that might be used to form a view is the set of all pipes that are members of links whose isolating valves are open. This subset can be used to form the basis of the mathematical model of the system. The views presented below are given as SQL instructions since SQL is designed as both a relational algebra and calculus and therefore conveys the formal meaning of the operation.

4.4.2 Topology from Database Views

In order to illustrate the special features of the proposed method more clearly, a test network has been devised that exhibits certain modelling difficulties considered to be representative of those encountered in actual distribution systems and which this approach is designed to overcome.

Consider the top portion only of the divided network shown in Fig 4.3. (See Fig 3.3 for the numbering symbology). Note that if isolating valves 3 and 4 are closed then the network is effectively split into two halves (or islands). One island has a source, the other does not. Clearly, it would be appropriate to model the island containing the source (called an active island) and to ignore the other (inactive) island since with no source in the inactive island there is no pressure to supply any demand. Furthermore, this split occurs in the middle of what would normally be considered to be two nodes (clusters 2;3;4 and 6;7) in a schematic network diagram for a typical network analysis program. Altering the status of these isolating valves therefore creates a substantially different system as far as the model is concerned. A further feature of this network is that if valve 5 is shut then the demand in that pipe must be assigned to the nodes connected to the pipe in which the valve is situated according to the division of the demand on both sides of the valve. However, if valve 6 is shut as well, then any consumers in between the two valves will no longer be supplied and the demand at node 4 must be re-adjusted to take into account of this.

The topology problem, as mentioned above, comprises two subproblems: the determination of node cluster connectivity; and the determination of system connectivity. A node cluster is any collection of nodes connected by links which have insignificant impedance. These are called zero-impedance links to emphasise the fact that the head drop across them is likely to be too small for any local pressure transducers to resolve correctly. Links satisfying this description are given the classification 'node' in the

Dual Test Network for Topology Validation

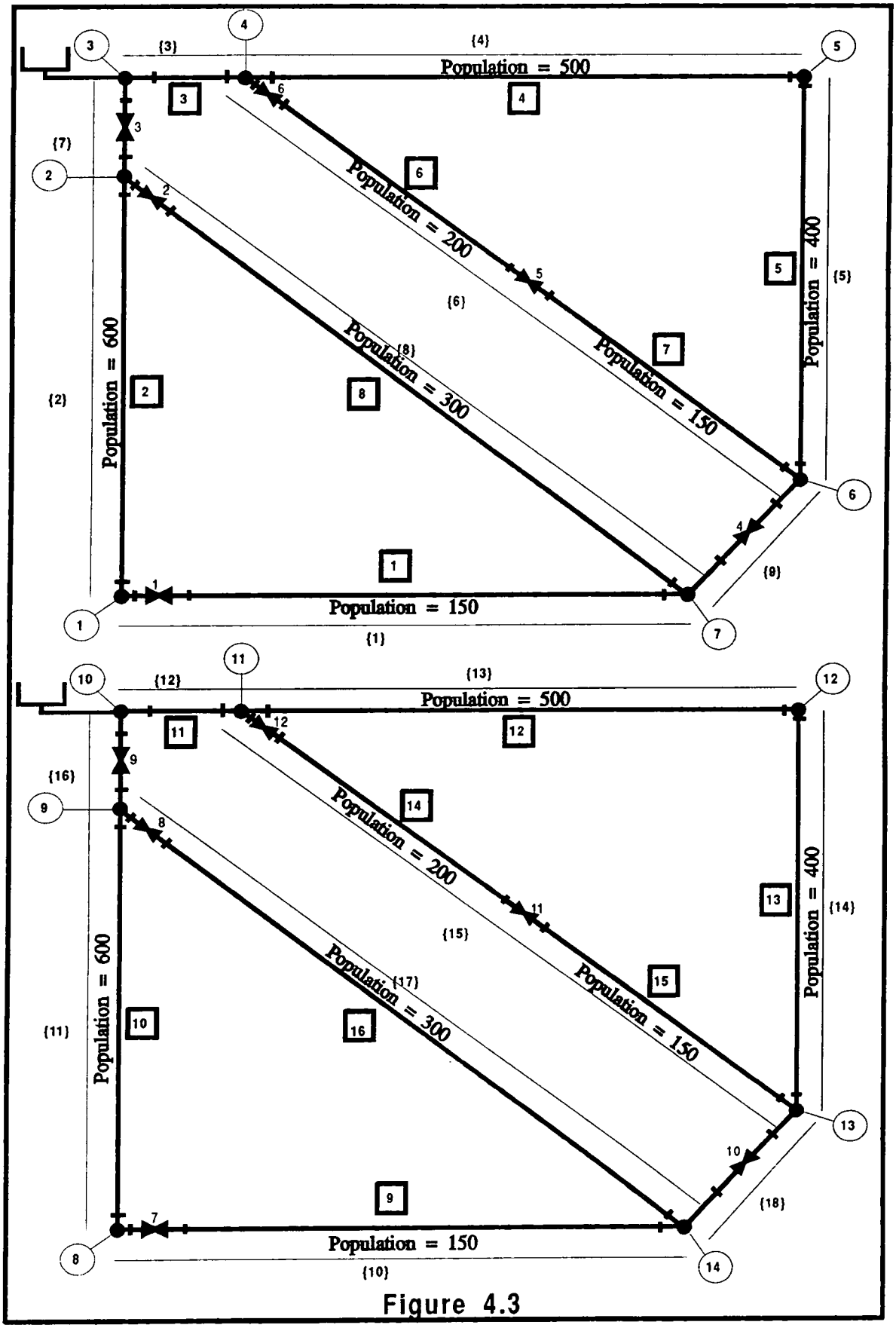


Figure 4.3

`Links.type` column. The user of the system can determine which links are to be considered zero-impedance links and hence can determine the boundaries of each node cluster without having to allocate a node cluster number.

Given that every link in a node cluster is classified as being of type 'node' and that the status of the valves that are members of any of these links is assumed known, then a set of node pairs joined by links of type 'node' whose valves are open can be created directly from the database. These node pairs can be treated as single nodes in the mathematical model and hence each pair can be combined into one equivalent node. To retrieve this list of node pairs a view can be constructed containing the following query:

```
select node_i, node_f
from Links
where
        type = 'node'
and not exists
        (select * from Valves
        where
            Valves.status = 'closed'
        and
            Valves.link_no = Links.link_no);
```

Using the complete network shown in Fig. 4.3 (ie nodes 1 – 14) and assuming valves 3, 4, 9, & 10 are open, the above query produces the results shown in Table 4.1.

Nodepairs	
node_i	node_f
3	2
3	4
7	6
10	9
10	11
14	13

Table 4.1

These node pairs are stored in an arbitrary order and can be thought of as aliases of each other. The alias concept is used to create a vector whose index represents the actual node number but whose value represents the alias by which that node is known. This algorithm is explained as follows for a system with N nodes (where '$\leftarrow$' represents assigment):

1. Initiallise the alias vector to 0;

2. Retrieve new node pair p, q from the node pair list.

if $alias(p) = 0$ and $alias(q) = 0$ then
$alias(p) \leftarrow p$
$alias(q) \leftarrow p$
if $alias(p) = 0$ and $alias(q) \neq 0$ then
$alias(p) \leftarrow alias(q)$
if $alias(q) = 0$ and $alias(p) \neq 0$ then
$alias(q) \leftarrow alias(p)$
if $alias(p) \neq 0$ and $alias(q) \neq 0$
and $alias(p) \neq alias(q)$then
for $i = 1, \ldots N$ do
if $alias(i) = alias(q)$ then
$alias(i) \leftarrow alias(p)$

Repeat until no more node pairs.

3. Set all nodes with no alias to their original number:

```
for i = 1,...N do
    if alias(i) = 0 then
        alias(i) ← i
```

This algorithm thus sets up a pointer array which maps the actual node numbers into their alias numbers. The set of links that will comprise the mathematical model can now be retrieved from a view containing the following select operation:

```
select link_no, node_i, node_f
from Links
where
    type ≠ 'node'
    and not exists
        (select * from Valves
        where
            Valves.status = 'closed'
        and
            Valves.link_no = Links.link_no);
```

As the node values are obtained, they are mapped into their alias node numbers, thus constituting a set of links connected at common node numbers. The following table is the result of this operation for the same network assuming, this time, that all the valves are open:

Topology		
link_no	node_i	node_f
1	1	7
2	1	3
4	3	5
5	5	7
6	3	7
8	3	7
10	8	14
11	8	10
13	10	12
14	12	14
15	10	14
17	10	14

Table 4.2

Although the correct topology has now been established, the full system connectivity is unknown, ie it is not known whether there is more than one island in the system. Application of the algorithm for determining the node alias numbers to the above set of node pairs shown in Table 4.2 immediately reveals the island numbers. In this case there are two islands and the algorithm selects the numbers 1 and 8 which are derived from the node numbers encountered in the set of node pairs selected from the view.

For the same system, if all the valves are shut, a quite different situation results. This is shown in the following table:

Topology		
link_no	node_i	node_f
2	1	2
4	3	5
5	5	6
11	10	12
13	10	12
14	12	13

Table 4.3

Islands	
original node	island number
1	1
2	1
3	3
4	3
5	3
6	3
7	7
8	8
9	8
10	10
11	10
12	10
13	10
14	14

Table 4.4

The final stage is to determine which of the islands are active ie which islands have sinks and sources. This is a trivial final step and entails selecting all the data from the pump tables and reservoir tables for the case where those pumps or reservoirs have `status = 'on'` and are connected to the network at a given node. The node alias number can be used to retrieve the island number for each source in the system. Any island number not selected is not active. A similar determination of demand location is carried out. However, since demand is not fixed at one point in the network it is necessary to consider first the method for automatic demand allocation presented below in Section 4.4.3.

4.4.3 Automatic Allocation of Demand to Demand Nodes

In performing this type of network simplification, care must be taken not to 'loose' demand along the way. It is relatively straight forward to keep track of demand provided the DBMS supports the necessary mathematical functions that are part of the SQL standard† [8].

The method presented here is given in two stages. For each stage a view is used that holds the data necessary to allocate the demand for those pipes that are members of links with shut valves. Both English and SQL versions are given:

1 For those links where the valves are shut, find the population on each side of every shut valve. The minimum of this set of values for each link will then be assigned to each end of the relevant links.

```
create view Demand1(link_no, load_i)
as
```

† A well known DBMS that the author was using did not support these functions necessitating a switch to another DBMS product which has only recently, at the time of writing, been updated to support them.

```
select link_no, min(load_i)
    from Valves
    where
        status = 'closed'
group by link_no;
```

2 For each node, sum the total amount of load coming from all links incident to that node.

```
select Links.node_i, sum(Demand1.load_i)
    from Links, Demand1
    where
        Links.link_no = Demand1.link_no
group by node_i;
```

The two operations together calculate the demand at the sending end of each link in the database, a further two identical operations are required to calculate the demands at the receiving ends using `load_f`, `node_f` and `Demand2` where `load_i`, `node_i` and `Demand1` respectively appear above.

Both operations above use the standard SQL statistical functions `min` and `sum` (meaning 'minimum of' and 'sum of' respectively) which are provided by the DBMS. The second operation is called a *join*. In this case the operation joins the table `Links` with the view `Demand1` on the shared attribute `link_no`. That is, the specified column of the table is joined to the specified column of the view wherever the link numbers are the same. The `group by` clause eliminates duplication employing the appropriate statistical function wherever a combination of two tuples occurs.

In retrieving the demands from the final two views, each demand must be added to any demand currently assigned to the end node of each link involved. Furthermore,

the node numbers used should be the alias numbers rather than the original numbers. This can be achieved straight forwardly using the alias vector as a pointer and summing the demands as they are retrieved from the view. The final result is a set of demands that have been automatically allocated by summation to the appropriate aliased nodes. This completes the task of demand aggregation. A complete set of demand values and topological information is given in a series of tables (4.5) — (4.11) for various valve states in the example network:

Valve Status	
No.	Status
1	open
2	open
3	open
4	open
5	open
6	open

Islands	
Node No.	Island No.
1	1
2	1
3	1
4	1
5	1
6	1
7	1

Topology		
Link No.	Initial Node	Final Node
1	1	7
2	1	3
4	3	5
5	5	7
6	3	7
8	3	7

Tables 4.5 a,b,c

Valve Status	
No.	Status
1	open
2	open
3	open
4	open
5	closed
6	open

Islands	
Node No.	Island No.
1	1
2	1
3	1
4	1
5	1
6	1
7	1

Topology		
Link No.	Initial Node	Final Node
1	1	7
2	1	3
4	3	5
5	5	7
8	3	7

Tables 4.6 a,b,c

Demand Allocation	
Node	Added Demand
3	200
7	150

Table 4.6 d

Valve Status	
No.	Status
1	open
2	open
3	open
4	open
5	closed
6	closed

Islands	
Node No.	Island No.
1	1
2	1
3	1
4	1
5	1
6	1
7	1

Topology		
Link No.	Initial Node	Final Node
1	1	7
2	1	3
4	3	5
5	5	7
8	3	7

Tables 4.7 a,b,c

Demand Allocation	
Node	Added Demand
7	150

Table 4.7 d

Valve Status	
No.	Status
1	open
2	open
3	closed
4	closed
5	closed
6	closed

Islands	
Node No.	Island No.
1	1
2	3
3	3
4	3
5	3
6	3
7	1

Topology		
Link No.	Initial Node	Final Node
1	1	7
2	1	2
4	3	5
5	5	6
8	2	7

Tables 4.8 a,b,c

Demand Allocation	
Node	Added Demand
6	150

Table 4.8 d

Valve Status	
No.	Status
1	closed
2	closed
3	closed
4	closed
5	closed
6	closed

Islands	
Node No.	Island No.
1	1
2	1
3	3
4	3
5	3
6	3
7	7

Topology		
Link No.	Initial Node	Final Node
2	1	2
4	3	5
5	5	6

Tables 4.9 a,b,c

Demand Allocation	
Node	Added Demand
6	150
7	450

Table 4.9 d

Valve Status	
No.	Status
1	closed
2	closed
3	open
4	open
5	closed
6	closed

Islands	
Node No.	Island No.
1	1
2	1
3	1
4	1
5	1
6	1
7	1

Topology		
Link No.	Initial Node	Final Node
2	1	3
4	3	5
5	5	7

Tables 4.10 a,b,c

Demand Allocation	
Node	Added Demand
7	600

Table 4.10 d

Valve Status	
No.	Status
1	closed
2	closed
3	closed
4	open
5	closed
6	closed

Islands	
Node No.	Island No.
1	1
2	1
3	3
4	3
5	3
6	3
7	3

Topology		
Link No.	Initial Node	Final Node
2	1	2
4	3	5
5	5	7

Tables 4.11 a,b,c

Demand Allocation	
Node	Added Demand
7	600

Table 4.11 d

Note that in these tables the demand has been summed at the correct point and that the island number appears alongside the list of demands. Note also that the island number can now be used to determine whether there exists any islands that have no consumers attached to them.

While discussing demand, it is perhaps appropriate to consider how to deal with leakage. Although leakage is often quoted as a value per property per hour, it depends more closely upon the number pipes currently in service, their length and the operating pressure. This is discussed in two reports [128, 228] and practical examples can be found in [190, 226]. In his Ph.D. thesis, Germanopoulos, [121], made an important step in simulating leakage as a pressure dependant variable. In accordance with [121] and

work by Jowitt, Germanopoulos and Xu, [122, 147], leakage is considered in this work as a variable derived from measurements rather than a value to be stored in a relation.

The final problem to consider in this section is that of series and parallel pipe combination. This arises because of the possibility of there being two or more elements comprising one link or when two or more pairs of nodes are combined thus forming a parallel set of links. The serial combination is carried out first and can be done by recognising the occurrence of more than one element with identical link numbers. Finally, parallel combinations are calculated to eliminate repeated pairs of node numbers, the exact algorithms for which are presented in the next chapter.

4.4.4 Keeping Track of Changes in the Database

Once a change has been detected in a given table, two courses of action are open. Firstly, the entire table can be re-read and a new model built. Secondly, the `select` operation can request all the tuples after a specified time thus reducing the number of records retrieved in the process. Both methods are proposed in this work for different situations:

1 Changes likely to result in new topology. The addition of new links or isolating valve changes will result in a new system topology and therefore the entire model must be rebuilt. (Changes of this nature would require a manual edit and re-balance sequence for a standard analysis package but are taken care of automatically in the proposed system.)

2 Parametric changes. Changes to parametric values do not change the system topology and therefore a system re-build is unnecessary. Instead, only the rows pertinent to the change are accessed from the database.

3 Benign data changes. Such changes do not affect the mathematical model at all and therefore are ignored by the database revision control modules.

4.5 Performance Tests

A number of performance tests have been undertaken in a similar fashion to those in the previous chapter. The tests were carried out after creating networks of sizes ranging from 7 nodes to 7000 nodes, using, as before, an original seed network, two copies of which are shown in Fig. 4.3.

The tests were undertaken using the same program as described in Chapter 3, which loaded data into the database, retrieved it and then deleted it ready for the next test. In this particular case, the program called the subroutines to build the alias information, the island information and to determine the correct allocation of consumer demands. The total elapsed times (CPU + IO) are shown in Fig. 4.4. In Fig. 4.5, the efficiency of the system is shown in terms of records processed per second.

The times obtained show that there is a non-linear relationship between number of records processed and elapsed time which is almost certainly due to the way the operating system was allocating resources to the tasks. In these tests, unlike those in Chapter 3, there is a certain amount of floating point arithmetic going on, handled within the DBMS task and a small amount of additional processing within the subroutines themselves (calculation of alias numbering, for instance). Fig. 4.6 highlights the CPU and IO activities for the task of calculating nodal demand allocations. It is to be noted that one should add the three sets of timings of Fig 4.4 to the total time of Fig. 3.4 to get the total elapsed time for a new model to be available.

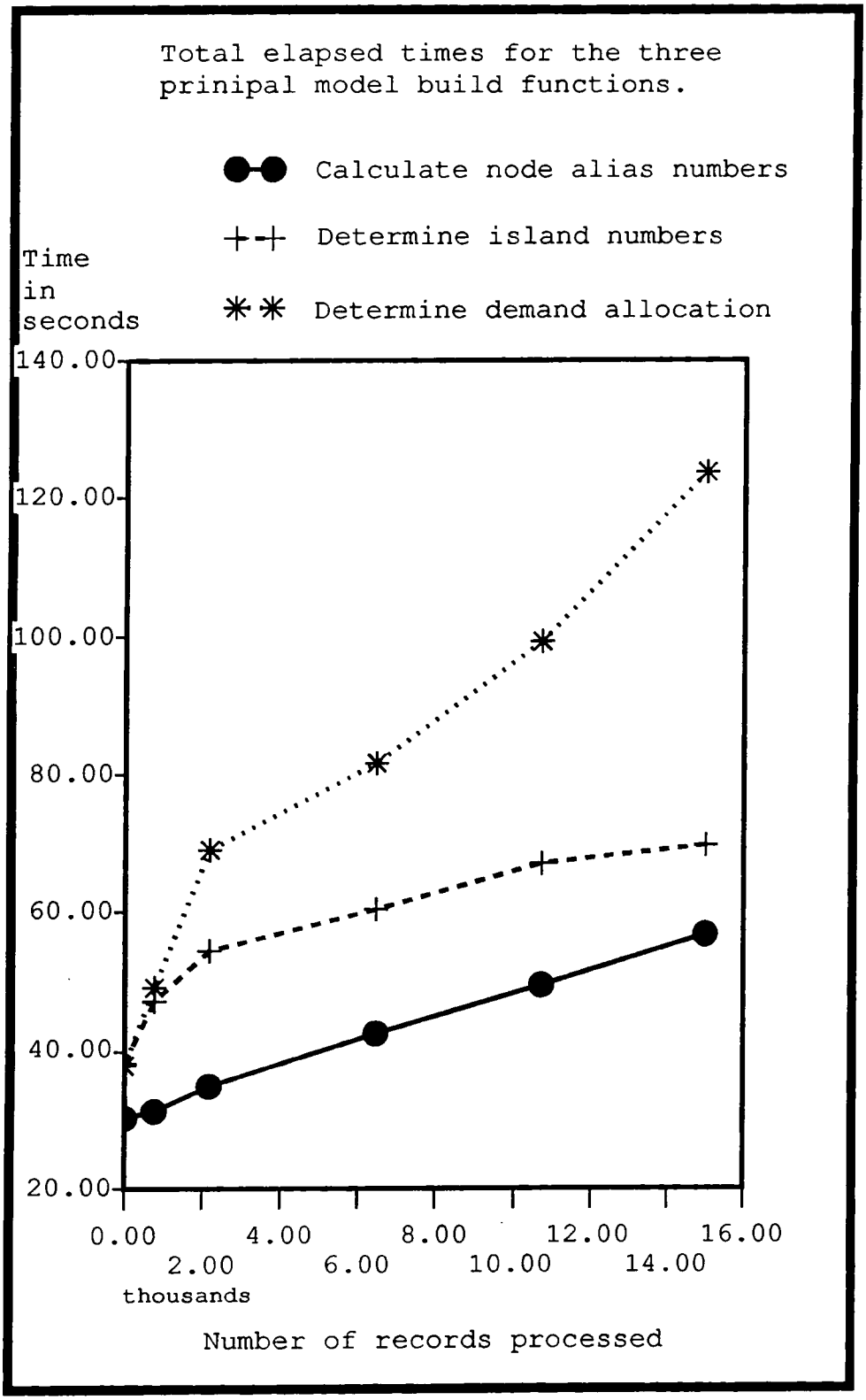

Total elapsed times for the three prinipal model build functions.
Calculate node alias numbers
Determine island numbers
Determine demand allocation
Time in seconds
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
thousands
Number of records processed

Figure 4.4

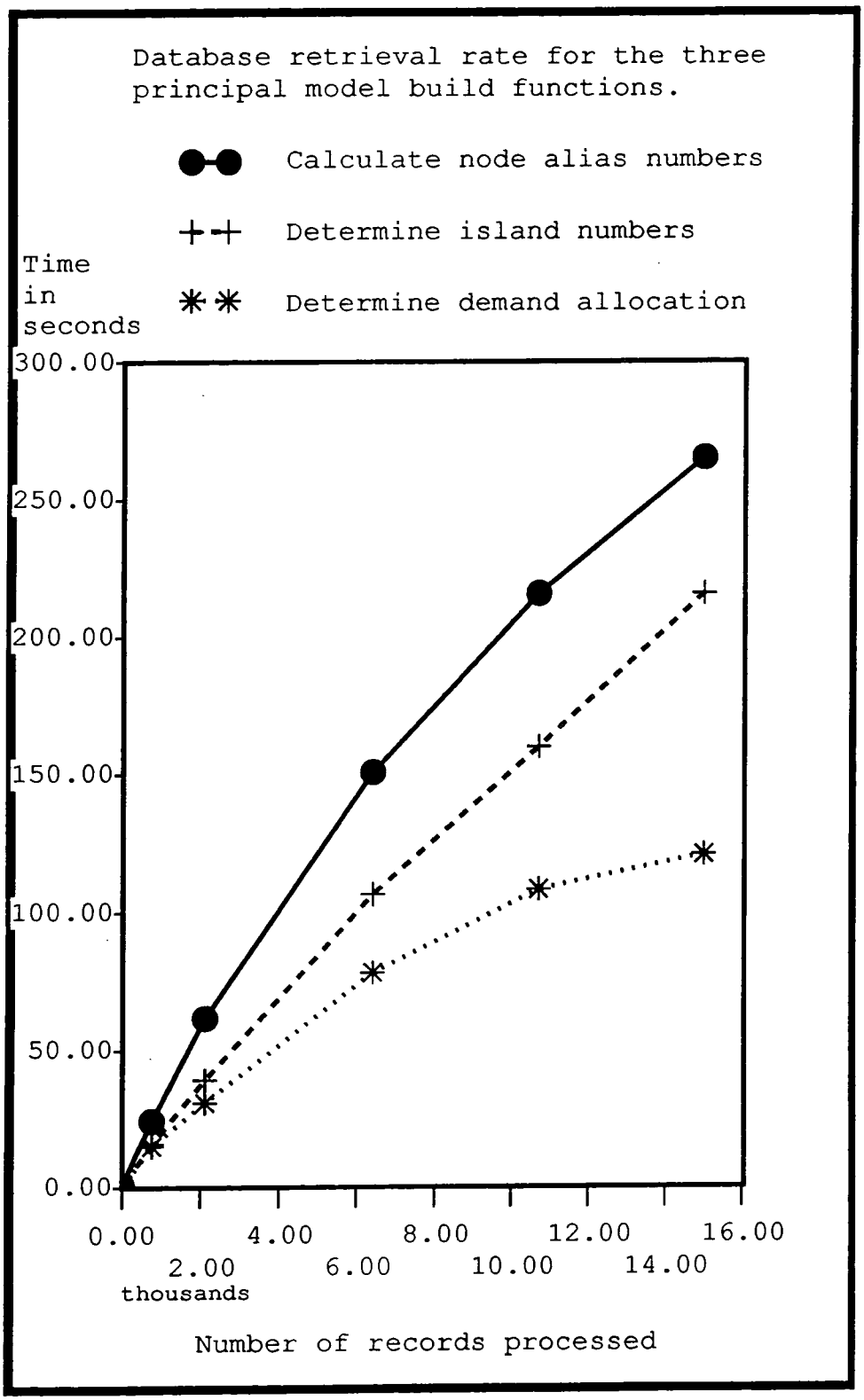

Database retrieval rate for the three principal model build functions.
Calculate node alias numbers
Determine island numbers
Determine demand allocation
Time in seconds
300.00
250.00
200.00
150.00
100.00
50.00
0.00
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
thousands
Number of records processed

Figure 4.5

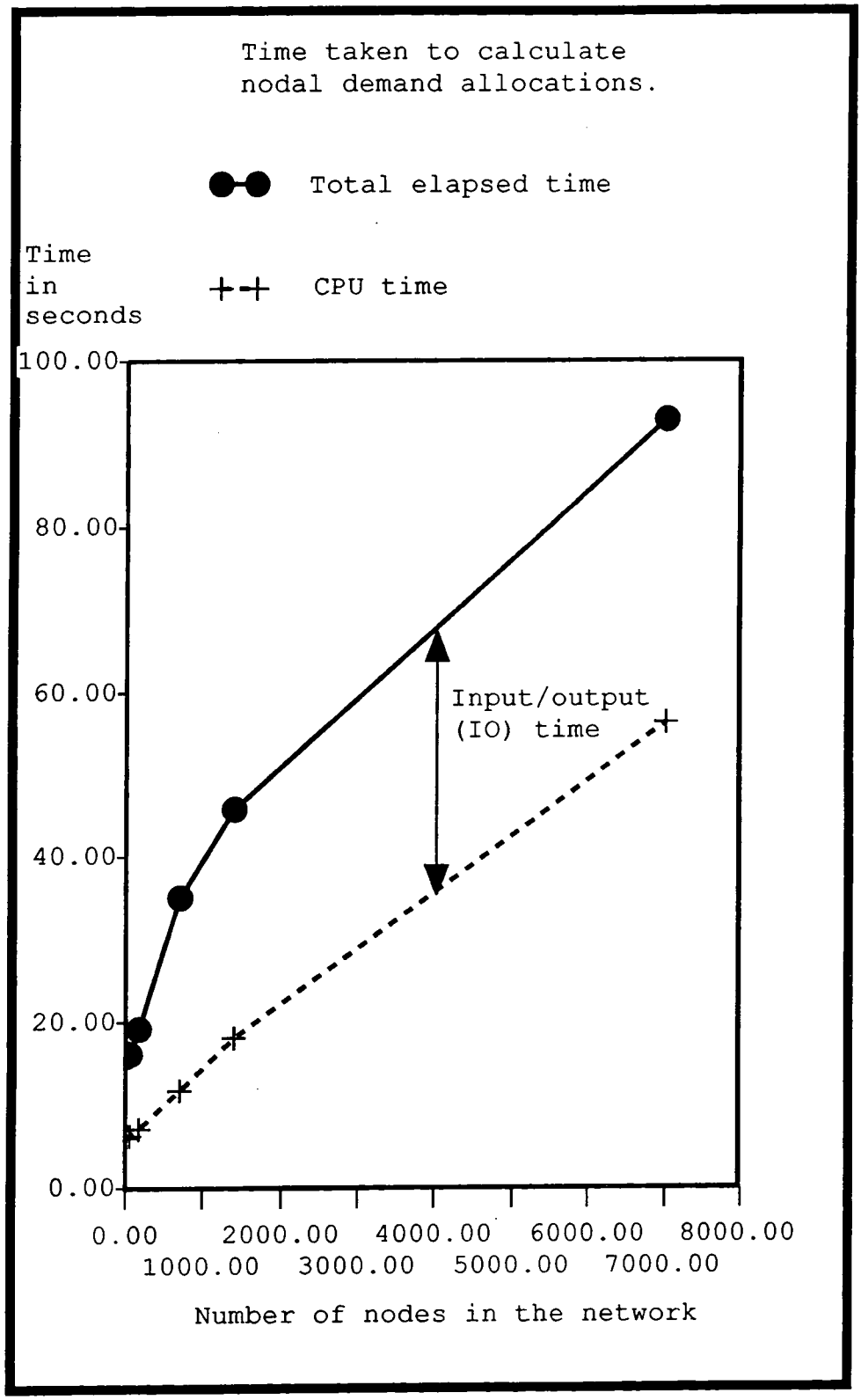

Time taken to calculate
nodal demand allocations.
Total elapsed time
CPU time
Time
in
seconds
100.00
80.00
60.00
40.00
20.00
0.00
Input/output
(IO) time
0.00
1000.00
2000.00
3000.00
4000.00
5000.00
6000.00
7000.00
8000.00
Number of nodes in the network

Figure 4.6

Model build time	
Activity	Elapsed time
get parameters	45 seconds
get alias numbers	57 seconds
get islands	68 seconds
allocate demands	125 seconds
TOTAL	295 seconds

Table 4.12

Thus for a 7000 node system, this amounts to an elapsed time of about 5 minutes and represents the additional overhead in using the database. As was stated before, the comparison with this elapsed time and the time taken to read directly from a file is completely irrelevant. The comparison that should be made is that of manually determining a new topology, manually determining the unique node numbers for combined nodes and manually determining the demand allocations by observing valve statii and node combinations. This series of operations would then be followed by a data re-entering exercise and only after this had been successfully completed would the new file be read in. Obviously, the time taken to read in the new file will be insignificant when compared to the other operations for a 7000 node system.

In considering these results, as in Chapter 3, it should be realised that the computer's system parameters can have a marked affect on performance. The problem of demand allocation for instance causes the DBMS to construct temporary tables of one view and to build the data for the second view out of this. Therefore the processing requirements of these transactions are far greater than the simpler transactions presented in the previous chapter and are even more sensitive to the correct tuning of the computer operating system. Secondly, since the computer workstation used in these experiments was a minimal system (about 1MIP with 8Mbytes of memory, amounting to about 2–3Mbytes

usable), it is certain that the timings represent worst cases as it is unlikely that a smaller or less powerful computer would be used in practice.

4.6 Conclusions

Considerable time and effort has been and still is being devoted to the building of large asset databases by water companies. In this chapter we have seen that this type of data can be used directly by the analysis code. Great care has been taken in the design of the database schema to store data in a way that most aptly describes the individual items concerned rather than make concessions to the way the data might subsequently be used. The work shows that by careful design of queries, by building views and by making some small additions to the database scheme, the basic asset data can be used directly and that there is no need to hold additional data separately for the modelling task (with the exception of task meta data such as convergence control parameters etc.).

This chapter has additionally shown that even if it is not desirable or possible to connect directly to a real asset database, storing data in the way described has distinct advantages over a file based approach in that it gives the user much more flexibility to modify the network data in a way that closely resembles the operation of the plant itself.

The present chapter has proposed new algorithms for logical network simplification (as opposed to simplification based on flow sensitivities). It has presented a new approach to handling demand values which allows demand to be entered into the model as it occurs in reality rather than forcing the operator of the system to determine the equivalent demand allocations by hand. This method has the further advantage in that the operator of the system can interact with the system model as if it was the physical system.

The speed of the data retrieval is acceptable but of course slower than a file based approach. However, the more significant comparison is that made between a system

where the operator has to edit a file, allocate new demands to demand nodes, determine network topology, calculate equivalent parameter values in the case of parallel or series elements and determine the effect of valve changes on the system islands. In this regard it is likely that the proposed system amounts to a considerable increase in performance over a file based approach.

CHAPTER 5

FLOW FORMULÆ FOR MODELLING WATER NETWORKS

5.1 Introduction

Before a full on-line monitoring scheme for a water supply network can be implemented, it is necessary to consider how to model the system concerned from both the descriptive data available and from the measurements of the time varying states obtained via the telemetry system. So far in this work, the storage of these two classes of data has been considered together with the way in which the data can be transformed into a format suitable for mathematical modelling, however, the modelling function itself has thus far been ignored.

Programs for network analysis using a variety of different flow models have been in existence for many years and are widely used throughout the water industry. However, use of a mathematical model as the core of an on-line monitoring scheme requires some special consideration. In particular, it is necessary to ensure that the model is representative of the system for as wide a range of conditions as possible and is not chosen purely for its computational simplicity. It is also necessary to consider the effects of lumping parameters together in the course of building the model and what errors may result in so doing.

The aims of this chapter are to present the head/flow formulæ of various network elements, to consider their inter-relationship and to determine the effects of consumer demand which is distributed along pipes. The combination of network elements is also presented as this compliments the methodology outlined in Chapter 4. Section 5.2 reviews the previous work in this area, Section 5.3 briefly introduces the concepts of water network simulation; Sections 5.4 through 5.6 present the equations typically used in network simulation; Section 5.7 shows the inter-relationship between two flow models and presents a simple way for converting data between the two. Section 5.8 discusses methods for combining parallel and series elements and Section 5.9 discusses the possible modelling errors that might arise from inexact modelling of distributed consumer demand. In Section 5.10 the threads of the chapter are brought together to show how the network model can be derived from the data held in the database. Section 5.11 summarises the findings of the chapter and draws several conclusions from the work.

5.2 Review of Previous Work

Computational methods to aid water network analysis first became available with the Hardy-Cross relaxation method [76] in 1936. This method permits calculation of flows and pressures in networks using a simple loop-by-loop scheme. The procedure requires an initial estimate of flows in each pipe in the network and also requires that the node continuity equations are satisfied. A node flow in-balance is permitted but this is treated as an invariant nodal load throughout the calculation. The algorithm then solves for the head in-balance in a loop-by-loop sequence. Despite occasional poor convergence properties [145], the Hardy-Cross method found favour in the water industry early on because it is relatively easy to implement using a pocket calculator [6].

When digital computers became more widely available, more sophisticated linear algebraic methods were adopted. The solution of large sets of linearised network equations by the Newton-Raphson method was proposed in [164] but convergence was not

guaranteed unless the starting guess was close to the final solution [167], however, a later modification given by Lemieux [156] provided a method which converged from a very poor start. A large number of publications appeared during the 1970's reporting on implementations of the Newton-Raphson method e.g. [97, 104, 120, 132, 185].

More recently, Wood and Charles, [260], proposed the 'Linear Theory Method' of solution which avoids forming the Jacobian terms necessary in the Newton-Raphson method. This method uses the following re-formulation of the Hazen-Williams flow model discussed later:

$$q_{ij}|q_{ij}|^{0.852} = U_{ij}\Delta h_{ij} \tag{5.1}$$

where

$$U_{ij} = \frac{10.743 C H W_{ij}^{1.852} D_{ij}^{4.87}}{L_{ij}} \tag{5.2}$$

which moving from the $(p)^{th}$ iteration to the next, can be written as:

$$q_{ij}^{(p+1)} = \frac{U_{ij}\Delta h_{ij}}{|q_{ij}^{(p)}|^{0.85}} \tag{5.3}$$

which was shown to be convergent provided a reasonable starting guess was found. Germanopoulos [121] and later, Jowitt and Xu [147] have used the linear theory method in leakage control applications.

A graph-theoretical procedure proposed by Kesavan and Chandrashekar [148] enables the head in-balance loop equations to be built using the fundamental circuit matrix with a suitable choice of spanning tree. The node flow continuity equations are similarly assembled using the cut-set matrix applicable to the original spanning tree. A method of choosing the best spanning tree is presented which is dependent on the position of the constant pressure devices in the network and on the location of any fixed flow devices such as fixed flow pumps. Cohen and Carpentier use a very similar argument in their formulation of the state estimation problem [66].

An approach used by DeMoyer and Horwitz [87–89] employs a Kalman filtering algorithm in an extended period network simulation. More recently Coulbeck et al. [72] have used regression analysis in order to achieve a tracking simulation model of the supply system on which to build optimised control strategies.

Germanopoulos and Jowitt [121, 122] have shown that leakage can be incorporated quite simply into the network model mass balance equations. This enables the pressure dependent leakage term to be included as a hydraulic constraint in a linear program to minimise distribution system leakage using the valves in the network as the actuators for the pressure reduction. The leakage control problem was formulated earlier by Sterling and Bargiela [238] without the pressure dependent leakage term but in which the Newton-Raphson method was used rather than the linear theory method. An extension to Bargiela's method has now been derived which permits the pumps and the valves to be used simultaneously in the optimal pressure reduction algorithm [266].

Following the design of network analysis algorithms, Rahal, Sterling and Coulbeck designed a system that enabled the network analysis function itself to be carried out more simply [201–203]. Their method was to identify the parameters in a network model as likely candidates for 'tuning' and then to alter these values (typically pump parameters or CHW values) such that an error function derived from the differences between measured and calculated network flows and pressures was minimised.

Non-linear optimisation methods for network analysis have also been proposed using gradient methods [210, 252] but while more capable to converge from a distant starting point in the solution space, converge can be slow due mainly to the line search part of the algorithm.

Pipe resistance formulæ are discussed in [16, 53, 167, 180, 204, 254, 261]. Much of the work done by these researchers concerns the reliability of various pipe flow formulæ

and their origins. Pipe combinations are also covered in some of these references e.g. [145, 146] whereas more sophisticated network simplification has been undertaken by Chen et al [47, 48].

5.3 Water Network Simulation

In this section a general formulation of the water network simulation problem is presented for the sake of completeness, since the principal idea behind this work is to propose a move away from simulation towards on-line state estimation.

The flow along the k^{th} link is a non-linear function of the head drop (or rise in the case of a booster pump):

$$q_k = f_k(\Delta h_k) \tag{5.4}$$

where $f(\cdot)$ is typically of the form $f = K\Delta h^{\gamma}$ and γ is normally non-integer.

The proposed method is, in common with most other methods, based on linearisation about a current point followed by linear solution exploiting linear algebraic methods.

Given a network of N variable head nodes and F fixed head nodes, the continuity equation for all the nodes states that the sum of flows entering and leaving each node must equate to zero. Although slightly unconventional, we will assume that flows into a node are $-ve$.

$$\sum_{\substack{j=1 \\ j \neq i}}^{N+F} q_{ij} + C_i = 0 \qquad i = 1, \ldots, n \tag{5.5}$$

where q_{ij} is the flow leaving node i and $n = N + F$. N.B. $q_{ij} = 0$ for nodes not connected to node i. C_i are the $(-ve)$ inflows from sources, or the $(+ve)$ loads but are actually made up of a number of components due to various types of demand and inflows from reservoirs or pressure reducing valves connected to other networks etc. Representing

these two broad categories as d_i and u_i respectively, (where u_i are $-ve$), we have, using Eqn. 5.4 (the unbalanced case, ie prior to convergence):

$$g_i = \sum_{\substack{j=1 \\ j \neq i}}^{N+F} f_{ij}(\Delta h_{ij}) + d_i + u_i \neq 0 \qquad i = 1, \ldots, n \tag{5.6}$$

The demand vector, **d**, is considered known *a priori* in the simulator but is treated as a noisy measurement in state estimation. For the purposes of this derivation, we will consider only the simulation case. The inflow vector **u**, on the other hand, is an unknown and must be calculated. There are, therefore, $N + F$ unknowns corresponding to the N heads and F inflows.

We can easily linearise Eqn. 5.6 using a Taylor expansion in two variables where we allow an approximation to occur by taking just the first order terms:

$$\sum_{j=1}^{N} \frac{\partial g_i}{\partial h_j} \Delta h_j + \sum_{j=N+1}^{N+F} \frac{\partial g_i}{\partial u_j} \Delta u_j = -g_i \qquad i = 1, \ldots, n \tag{5.7}$$

Renaming the vectors as follows:

$$\begin{pmatrix} \Delta h_1 \\ \Delta h_2 \\ \vdots \\ \Delta h_N \\ \Delta u_{N+1} \\ \vdots \\ \Delta u_{N+F} \end{pmatrix} = \begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_N \\ \Delta x_{N+1} \\ \vdots \\ \Delta x_n \end{pmatrix} \tag{5.8}$$

which, in matrix notation gives:

$$\mathbf{J}\Delta\mathbf{x} = -\bar{\mathbf{g}} \tag{5.9}$$

where the matrix **J** is termed the Jacobian matrix. $\Delta\mathbf{x}$ is sometimes termed the vector of corrections due its role in the Newton-Raphson sequence, where, on the k^{th} iteration:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha\Delta\mathbf{x}^{(k)} \tag{5.10}$$

The quantity α is used to control the Newton-Raphson steps. This could be found using a line search, e.g. Fibonacci as in [156] but here, after reasonably extensive testing, [254], we use $\alpha = 0.6$.

An alternative formulation, given in [246], elegantly demonstrates how the arc-node incidence matrix for the network imparts its structure into the above equations. The sparsity structure of water network equations is also discussed in [44].

Eqn. 5.9 is solved in this work at each iteration using Guassian elimination via Harwell subroutines LA05 [206]. Iterations are halted at a point when the value of $\mathbf{\Delta x}$ has reached a predefined small value.

5.4 Hazen-Williams Formula

The Hazen-Williams formula owes its popularity to its simplicity in form and accuracy over the range of flows frequently encountered in water supply networks. It is normally presented in its S.I. units format:

$$h_i - h_j = 10.743 CHW_{ij}^{-1.85} L_{ij} D_{ij}^{-4.87} q_{ij}^{1.85} \tag{5.11}$$

or

$$q_{ij} = 0.27746 CHW_{ij} D_{ij}^{2.63} \left(\frac{|h_i - h_j|}{L_{ij}} \right)^{0.54} \tag{5.12}$$

or, for computer program implementation:

$$q_{ij} = K_{ij}^{-0.54} (h_i - h_j) |h_i - h_j|^{-0.46} \tag{5.13}$$

which ensures the correct sign for the flow in the link and where:

$$K_{ij} = 10.742 CHW_{ij}^{-1.85} L_{ij} D_{ij}^{-4.87} \tag{5.14}$$

It is assumed that the resistance measurement is obtained for pipes not containing control valves such as PRV's although they may contain isolating valves. This being

so, pipes and isolating valves are modeled as single elements and control valves are modeled separately. For control valves, the same flow formula can be used as for pipes, suitably modified to take into account of the pressure controlling action of the valve. This is an approximation since it assumes that the controlling set pressure, H_{PRV} is fixed and known. In fact, as shown in [91, 198, 248], valves impart their own dynamics as they attempt to control the pressure and hence H_{PRV} may differ from the value applicable to the valve. However, a dynamic model of a control valve would depend on the physical construction of the valve and therefore it is proposed to continue with the more conventional control valve model in this work for the sake of generality.

Conventionally, the valve equations are written as follows (assuming ideal, steady-state conditions):

1) Pressure reducing valve, Fig 5.1:

$$q_{ij} = \begin{cases} K_{ij}^{-0.54} |H_{PRV} - h_j|^{0.54} & h_i > H_{PRV} > h_j \\ 0 & H_{PRV} \leq h_j \\ K_{ij}^{-0.54} |h_i - h_j|^{0.54} & H_{PRV} \geq h_i > h_j \end{cases} \tag{5.15}$$

2) Pressure sustaining valve, Fig 5.2:

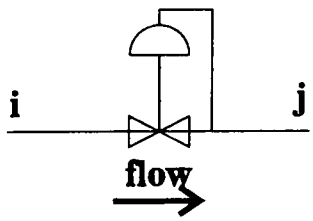
$$q_{ij} = \begin{cases} K_{ij}^{-0.54} |H_{PSV} - h_j|^{0.54} & h_i > H_{PSV} > h_j \\ 0 & h_i < H_{PSV} \\ K_{ij}^{-0.54} |h_i - h_j| & h_i \geq H_{PSV} \end{cases} \tag{5.16}$$

3) Non-return valve, Fig 5.3:

$$q_{ij} = \begin{cases} K_{ij}^{-0.54} |h_i - h_j|^{0.54} & h_i > h_j \\ 0 & h_i \leq h_j \end{cases} \tag{5.17}$$

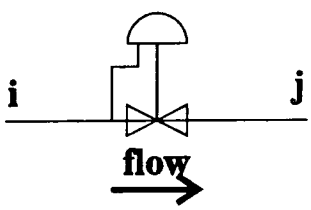
4) Remote control valve (eg level control or remote nodal pressure), Fig 5.4:

$$q_{ij} = \varphi K_{ij}^{-0.54} |h_i - h_j|^{0.54} \tag{5.18}$$



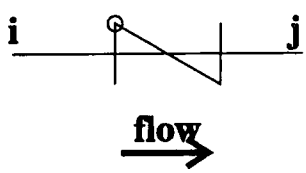
P.R.V.

Figure 5.1



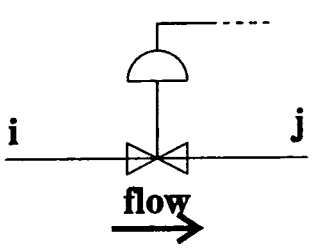
P.S.V.

Figure 5.2



Check Valve

Figure 5.3



Remote Control Valve

Figure 5.4

where $0 \leq \varphi \leq 1$.

5.5 An Explicit Form of the Colebrook-White Equation

The motivation for an accurate flow model, via the Colebrook-White formulation lies in the fact that the model derived is to be used for measurement validation purposes in the state estimator. Quite frequently it is tolerable to have a partly inaccurate model of a network where small discrepancies between information gathered from data loggers and the analysis model itself are insufficient to affect later decisions resulting from such an exercise. However, when designing a model of a system for later use in a state estimator for on-line data validation and leak detection, it is necessary to have as accurate a model as possible.

A large number of authors have discussed the Colebrook-White equation and its use in network analysis is not new. The original paper by Colebrook [67] derived the equation from a study of the smooth and rough flow regimes and a proposed a linear combination of the equations that describe these regimes. Matthew [166] provides an up-to-date exposition of the original derivation. Various papers, e.g. [16, 167, 204] show how to obtain a non-explicit form of this equation and this derivation is presented below:

The Colebrook-White equation, is usually quoted as:

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left\{\frac{e}{3.71D} + \frac{2.51}{Re\sqrt{f}}\right\} \tag{5.19}$$

where D is the pipe diameter, Re is the Reynolds number, f is the friction factor and e is the pipe surface roughness (sand particle size). Due to the appearance of the friction term, f, on both sides of this equation, it is necessary to derive an explicit version, this is achieved using the Darcy-Wiesbach formula Eqn. 5.20 and Reynold's formula Eqn. 5.21:

$$h = \frac{fLv^2}{2gD} \tag{5.20}$$

where L is the pipe length, v is the average velocity of flow and g is the acceleration due to gravity;

$$Re = \frac{vD}{\nu} = \frac{4q}{\pi D\nu} \tag{5.21}$$

where q is the volume flow rate and ν is the kinematic viscosity. We thus obtain

$$q = -\pi\sqrt{\frac{gD^5}{2L}}\sqrt{h}\log_{10}\left\{\frac{e}{3.71D} + \frac{2.51\nu}{\sqrt{\frac{2gD^3}{L}}\sqrt{h}}\right\} \tag{5.22}$$

which can be written as:

$$q_{ij} = -\pi\sqrt{\frac{gD_{ij}^5}{2L_{ij}}}\frac{h_i - h_j}{\sqrt{|h_i - h_j|}}\log_{10}\left\{\frac{e_{ij}}{3.71D_{ij}} + \frac{2.51\nu}{\sqrt{\frac{2gD_{ij}^3}{L_{ij}}}\sqrt{|h_i - h_j|}}\right\} \tag{5.23}$$

which guarantees a consistent sign convention where $q_{ij} > 0$ signifies flow from node i to node j. In this work Eqn. 5.23 is termed the 'explicit Colebrook-White equation'.

For the sake of brevity, Eqn. 5.23 can be written as:

$$q_{ij} = A_{ij}\frac{h_i - h_j}{\sqrt{|h_i - h_j|}}\log_{10}\left\{B_{ij} + \frac{C_{ij}}{\sqrt{|h_i - h_j|}}\right\} \tag{5.24}$$

where A_{ij}, B_{ij} and C_{ij} are obvious by inspection.

The valve equations can also be written in terms of the explicit Colebrook-White formulation as follows:

1) Pressure reducing valve, Fig 5.1:

$$q_{ij} = \begin{cases} A_{ij}\frac{H_{PRV} - h_j}{\sqrt{|H_{PRV} - h_j|}}\log_{10}\{B_{ij} + \frac{C_{ij}}{\sqrt{|H_{PRV} - h_j|}}\} & h_i > H_{PRV} > h_j \\ 0 & H_{PRV} \leq h_j \\ A_{ij}\frac{h_i - h_j}{\sqrt{|h_i - h_j|}}\log_{10}\{B_{ij} + \frac{C_{ij}}{\sqrt{|h_i - h_j|}}\} & H_{PRV} \geq h_i > h_j \end{cases} \tag{5.25}$$

2) Pressure sustaining valve, Fig 5.2:

$$q_{ij} = \begin{cases} A_{ij}\frac{H_{PSV} - h_j}{\sqrt{|H_{PSV} - h_j|}}\log_{10}\{B_{ij} + \frac{C_{ij}}{\sqrt{|H_{PSV} - h_j|}}\} & H_{PSV} > h_i > h_j \\ 0 & h_i \leq H_{PSV} \\ A_{ij}\frac{h_i - h_j}{\sqrt{|h_i - h_j|}}\log_{10}\{B_{ij} + \frac{C_{ij}}{\sqrt{|h_i - h_j|}}\} & h_i \geq H_{PSV} \end{cases} \tag{5.26}$$

3) Non-return valve, Fig 5.3:

$$q_{ij} = \begin{cases} A_{ij}\frac{h_i - h_j}{\sqrt{|h_i - h_j|}} \log_{10}\{B_{ij} + \frac{C_{ij}}{\sqrt{|h_i - h_j|}}\} & h_i > h_j \\ 0 & h_i \leq h_j \end{cases} \tag{5.27}$$

4) Remote control valve, Fig 5.4:

$$q_{ij} = \varphi\left(A_{ij}\frac{h_i - h_j}{\sqrt{|h_i - h_j|}} \log_{10}\left\{B_{ij} + \frac{C_{ij}}{\sqrt{|h_i - h_j|}}\right\}\right) \tag{5.28}$$

where $0 \leq \varphi \leq 1$.

5.6 Head/Flow Models for Other Elements

Pumps are normally modeled by a quadratic law:

$$\Delta h_{ij} = a_{ij}q_{ij}^2 + b_{ij}q_{ij} + c_{ij} \tag{5.29}$$

where a_{ij}, b_{ij} and c_{ij} can be obtained by pump testing although this is not always practical. If the pump manufacturer supplies pump curve data then the above parameters can be found by least squares fitting and then tuned, if necessary, using dynamic parameter tuning [203].

Hence,

$$q_{ij} = -b_{ij} \pm \sqrt{\frac{b_{ij}^2 - 4a_{ij}(c_{ij} - |\Delta h_{ij}|)}{2a_{ij}}} \tag{5.30}$$

A typical pump curve is shown in Fig. 5.5 and the two roots of Eqn. 5.30 are shown on the curve. Since a_{ij} is always $-ve$, the curve is an inverted parabola and hence the $-ve$ root causes unstable operation. Therefore, the $+ve$ root is chosen from the above relationship.

General form of the pump and system curves

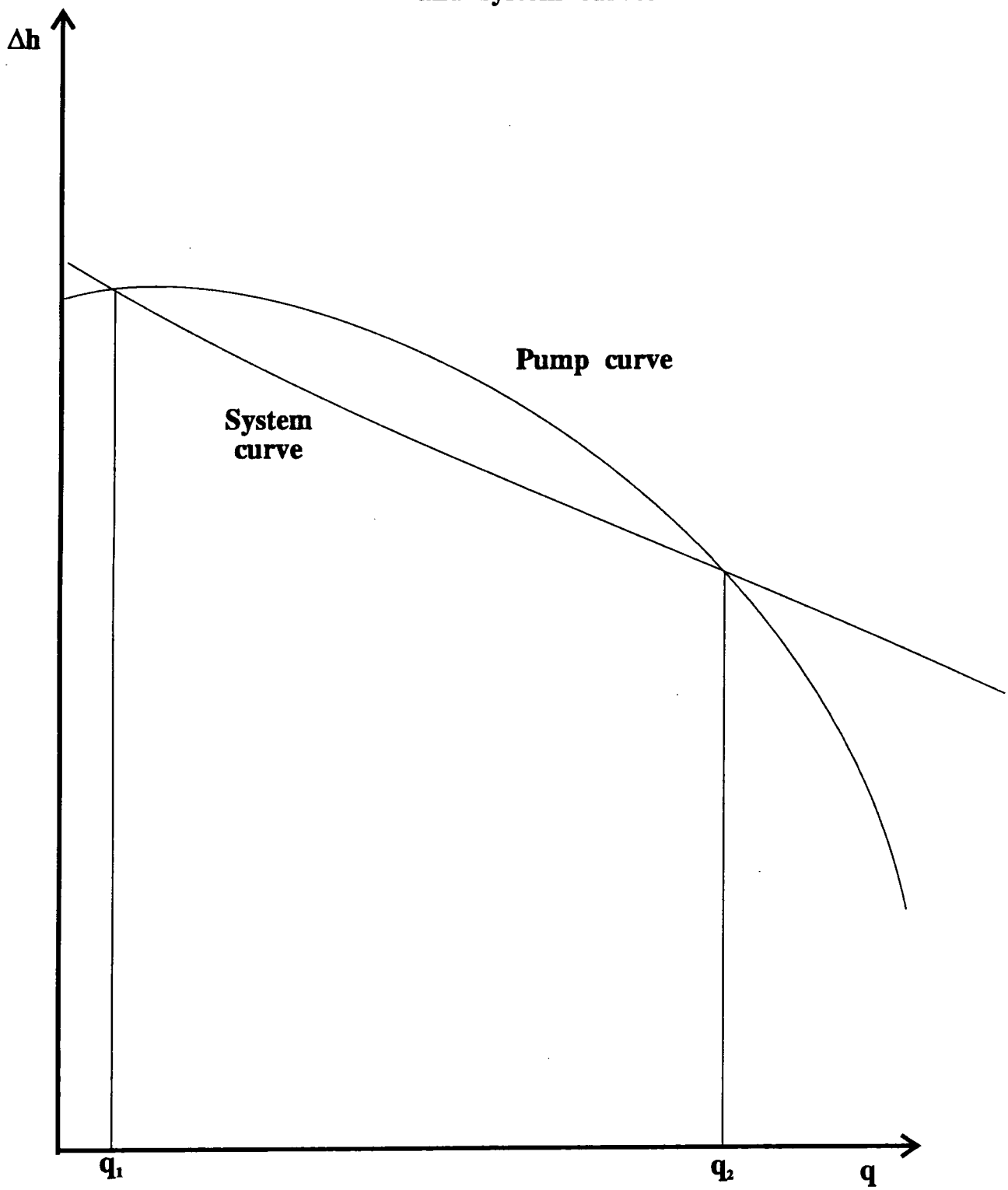


Figure 5.5

5.7 The Interrelationship between the formulæ of Hazen and Williams, Miller, Massey, Colebrook and White

The available flow formulæ for full pipe flow (as opposed to open channel flow) fall into one of two categories. Either they use Darcy's equation:

$$h_f = \frac{\Delta p}{\rho g} = \frac{flv^2}{2dg} \tag{5.31}$$

or, as in the Hazen-Williams equation they are direct ie $q = g(\Delta h)$, where $g(\cdot)$ is some non-linear function. Equations that rely on f are sometimes confusing since f can take on one of two meanings depending on the original derivation. In this work, except where stated in one case below, f is defined by its use in Eqn. 5.31 above.

Moore [180], presents an analysis of the interrelationships of certain equations commonly used in pipe flow calculations and also provides a discussion on the use of indices in the direct formulæ.

In order to understand the range of applicability of the two principal flow formulæ used in this work, a number of tests were run to show qualitatively the relationship between them. These results are shown in Figs. 5.6 & 5.7. The plots are effectively Moody diagrams (ie. Re vs. f plotted on log-log axes) and show that the Hazen-Williams equation is just a straight line of slope proportional to the index. Interestingly, Moore [180] showed that for very rough pipes, which perhaps have been in service a long time, an index nearer 2.0 (rather than 1.85) would be more suitable. Since the age distribution of pipes in many networks is approximately flat over the last 100 years, a significant portion of most distribution networks will have pipes that have been in service a long time and this observation demonstrates the dangers of using the Hazen-Williams formulation in cases outside the conditions for which it was designed. Furthermore, there is a range of Reynolds numbers for which the Hazen-Williams equation is not really applicable as can be seen for the different cases shown in Figs. 5.6 & 5.7. It is

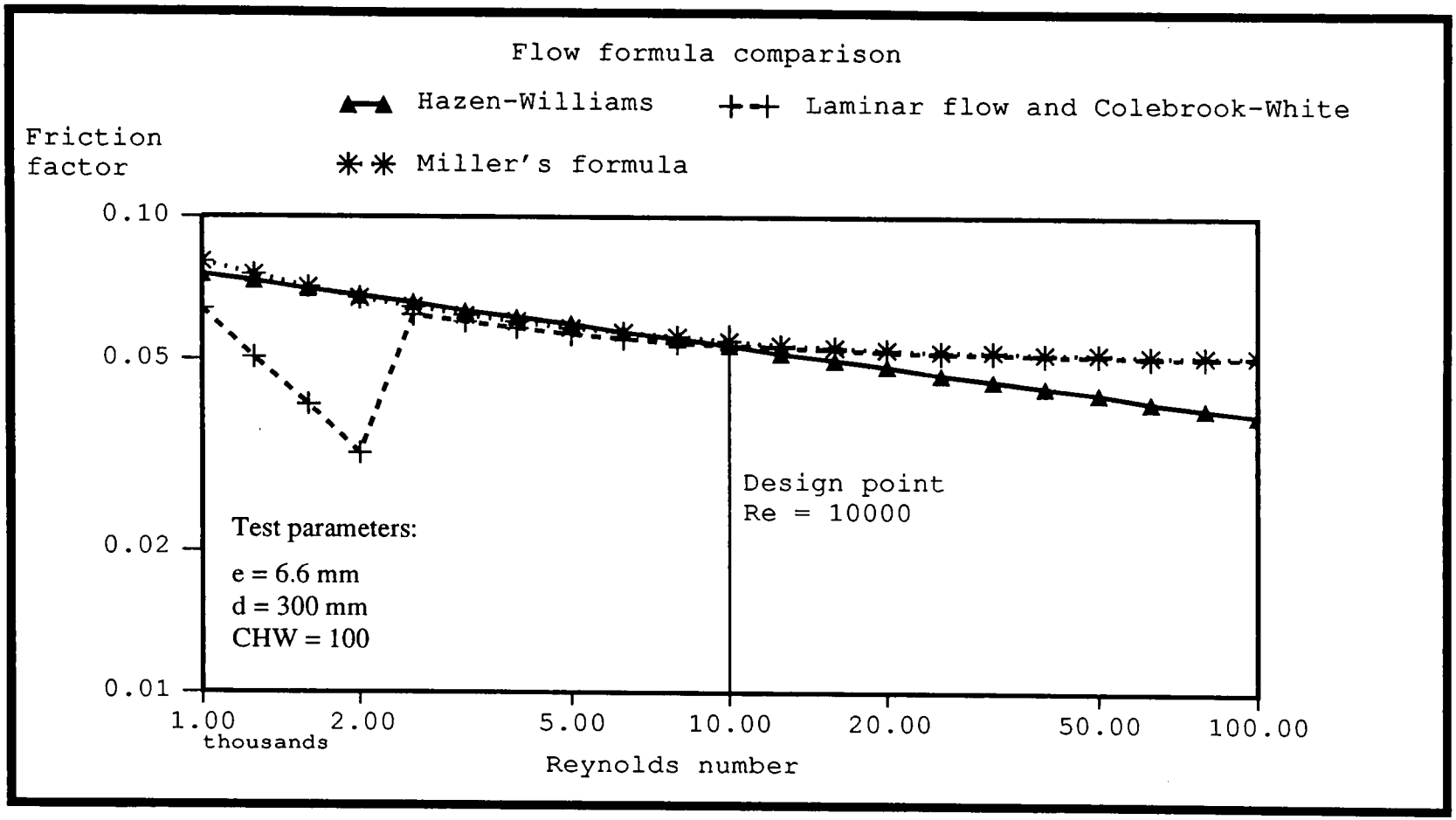

Flow formula comparison
Hazen-Williams
Laminar flow and Colebrook-White
Miller's formula
Friction factor
0.10
0.05
0.02
0.01
Test parameters:
e = 6.6 mm
d = 300 mm
CHW = 100
Design point
Re = 10000
1.00
thousands
2.00
5.00
10.00
20.00
50.00
100.00
Reynolds number

Figure 5.6

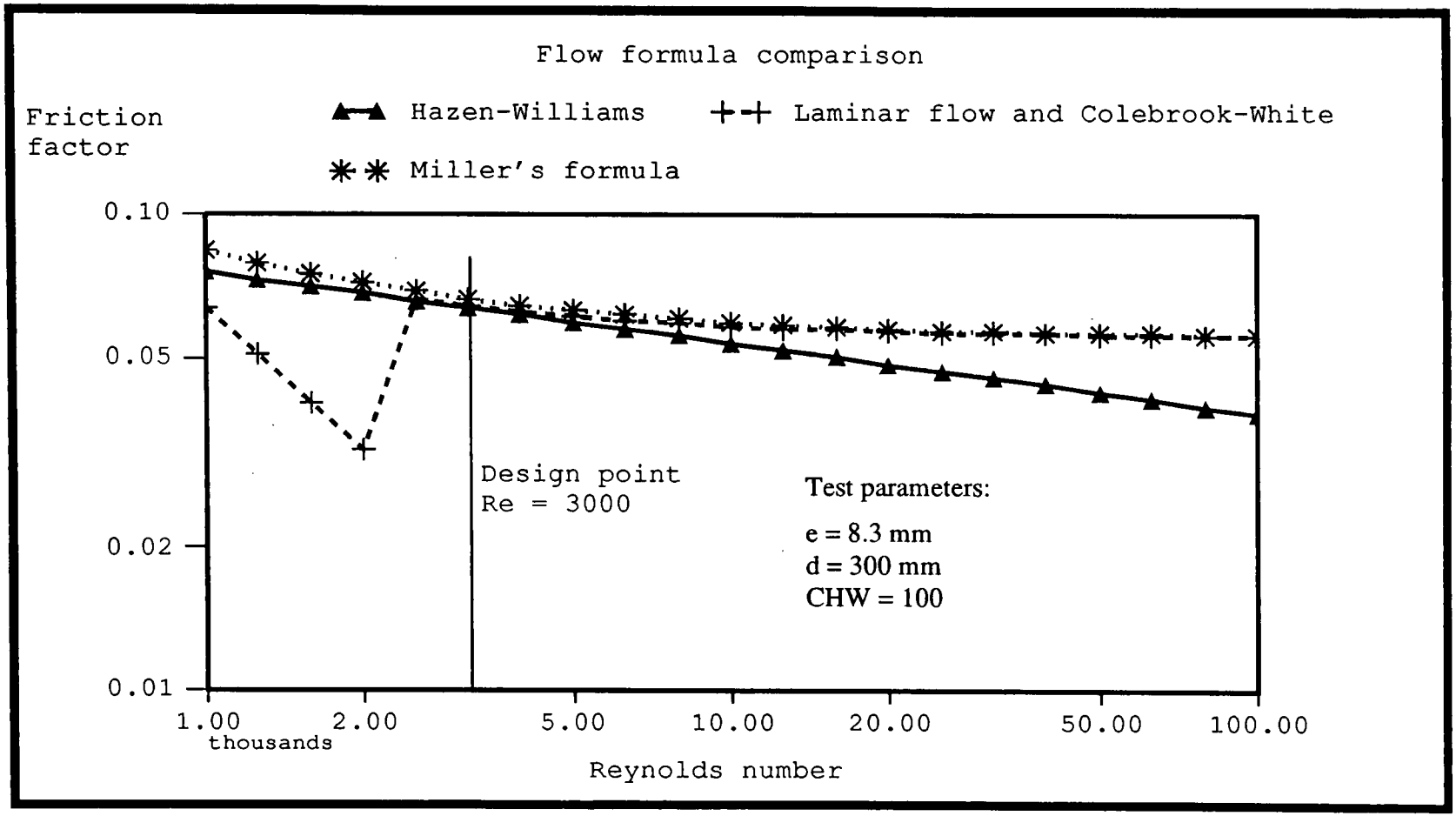

Flow formula comparison
Friction factor
Hazen-Williams
Laminar flow and Colebrook-White
Miller's formula
0.10
0.05
0.02
0.01
Design point
Re = 3000
Test parameters:
e = 8.3 mm
d = 300 mm
CHW = 100
1.00
thousands
2.00
5.00
10.00
20.00
50.00
100.00
Reynolds number

Figure 5.7

conceivable that unexpected errors might arise when using the Hazen-Williams formula for state estimation purposes for certain flow conditions.

Figs. 5.6 & 5.7 also show the curves for the equation known as Miller's equation which follow the Colebrook-White equation curves very closely and might therefore be considered a suitable alternative to either of the other two equations under discussion. Miller [174] introduces his formula in the following form:

$$f = \frac{0.25}{\left[\log_{10}\left(\frac{e}{3.7D} + \frac{5.74}{Re^{0.9}}\right)\right]^2} \tag{5.32}$$

alternatively, the equation can be rearranged into a more familiar form:

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left(\frac{e}{3.7D} + \frac{5.74}{Re^{0.9}}\right) \tag{5.33}$$

unfortunately, the substitution made in Section 5.5 to provide the explicit version of the Colebrook-White equation, viz:

$$Re\sqrt{f} = 2\sqrt{\frac{gD^3}{L}}\sqrt{h} \tag{5.34}$$

cannot be made in the right hand side of Eqn. 5.33 and hence we are left with either an f term or a q term neither of which is desirable. Since an explicit version of the original Colebrook-White equation is available and because of the difficulty with the right hand side of Miller's equation, this equation has not been adopted for this work.

Massey [165] also quotes an equation that can be used in the same way as Miller's equation:

$$\frac{1}{\sqrt{f}} = -7.2\log_{10}\left\{\frac{6.9}{Re} + \left(\frac{e}{3.71D}\right)^{1.11}\right\} \tag{5.35}$$

but the same arguments prevail as for Miller's equation.

One very good reason why the Hazen-Willams equation is so popular is that the method of obtaining the Hazen-Williams coefficient, CHW, in field tests is straight

forward. The value is derived by injecting a known flow along a pipe and measuring the resulting head drop. The head and flow values are then used in the Hazen-Williams equation to obtain the CHW value. Strictly speaking however, the CHW value thus obtained is only valid for the conditions for which the test was conducted. The CHW value is, in fact, marginally flow dependent as can be seen from the following equations.

The roughness values necessary for the Colebrook-White formulation are more difficult to obtain and hence there is a motivation for determining estimates of roughness values directly from the CHW values. This can be done by combining the Hazen-Williams, Darcy-Weisbach and Reynold's equations, ie Eqns. 5.11, 5.20 and 5.21 respectively giving:

$$f = \frac{134.4663}{CHW^{1.852} Re^{0.148} D^{0.0187} \nu^{0.148}} \tag{5.36}$$

Rearranging the Colebrook-White equation, Eqn. 5.19, an estimate of the effective value for e can be determined which is expressed below as $\hat{e}$:

$$\hat{e} = \left(10^{-1/2\sqrt{f}} - \frac{2.51}{Re\sqrt{f}}\right) 3.71D \tag{5.37}$$

Hence, from a knowledge of the parameters of the pipe and the conditions of the field test (ie ν and Re), it is possible to convert directly to an equivalent roughness value $\hat{e}$ or $\hat{e}/D$. It is therefore proposed that the Reynolds number for the field test which yielded CHW be held in the database and used to find $\hat{e}$ whenever e is not known in advance. This is the purpose of the 'design point' shown in Figs. 5.6 & 5.7. This very simple relationship provides more utility in Section 5.8 when making serial and parallel pipe combinations. Once the CHW and Re numbers are known, the parameters for the Colebrook-White equation can be obtained and used for a range of flows that might be inappropriate for the Hazen-Williams formula.

5.8 The Combination of Parallel and Series Elements

The combination of various network elements is necessary when exploiting the relational database approach proposed in Chapter 4 of this thesis since each element of the distribution network is stored in the database as a separate entity whereas, in reality, the elements in question may be connected together between nodes in parallel or in series. Therefore, to build the model of the distribution network from the data held in the database without further human intervention, a direct and automatic method for entity combination is required. In future work it might be desirable to allow for any combination of network elements but in this thesis only combinations of like entities are permitted. Pipes are the most common element for parallel or series combination since an operator is not normally interested in flows down individual pipe sections but rather across areas of the network as a whole. In some cases, pumps will be connected in parallel and therefore it is necessary to consider how the coefficients of parallel pumps combine to make a new pump curve. When pumps are connected in series, they are separated by large distances (often with supply points in between) when it would be inappropriate to combine their characteristics from the point of view of the analysis. The only other in-line elements are a variety of valves whose characteristics are normally modelled in the same way as pipes (e.g. Eqn. 5.25) and their combination therefore falls into the same class as pipes. Furthermore, it is more unlikely that parallel or series valve combinations exist in water networks. For these reasons, only pipe and pump combinations are considered in this work and for pumps, only parallel combinations are dealt with.

5.8.1 Pipe Combinations

A number of authors have discussed the problems of combining equivalent pipes, e.g. Moore [180] and Jeppson [146] and the procedure in the case of the Hazen-Williams formula is straight forward since it permits the adjustment of the hydraulic resistance parameter K_{ij} as the most convenient method for the determination of an equivalent

formula for two, or more, pipes combined in series or parallel. This type of combination is extended from the Hazen-Williams equation to the Colebrook-White equation in this section for both the parallel and the series case. The present method differs from that of Jeppson in that it is not iterative.

5.8.1.1 Series Combination

(N.B. In what follows, the subscript $_{e_s}$ refers to the equivalent parameter for a series combination whereas $_{e_p}$ refers to the equivalent parameter for a parallel combination.)

The Hazen-Williams equation can be written in the form:

$$\Delta h = Kq^{1.85} \qquad \text{where} \quad K = \frac{10.743L}{CHW^{1.85}D^{4.87}} \tag{5.38}$$

For a series of pipes connecting two nodes that are to be modelled as one link, it must be assumed that any consumer demand can be lumped at the end nodes of that link. Therefore, if there are p pipes in the link:

$$\Delta h_{e_s} = \sum_{i=1}^{p} \Delta h_i \tag{5.39}$$

and

$$q_{e_s} = q_i \quad \forall i = 1, \ldots p \tag{5.40}$$

therefore,

$$\Delta h_{e_s} = \sum_{i=1}^{p} K_i q^{1.85} = K_{e_s} q^{1.85} \tag{5.41}$$

hence,

$$K_{e_s} = \sum_{i=1}^{p} K_i \tag{5.42}$$

We are free to choose arbitrary values for the set of equivalent pipe parameters, $\{L_{e_s}, D_{e_s}$, and $CHW_{e_s}\}$, provided they satisfy Eqn. 5.42. However, when determining

the equivalent result for the more complicated explicit Colebrook-White equation, non-arbitrary values for these constants will have to be chosen. Intuitively, the length of the equivalent pipe should be equal to the sum of the lengths of the component pipes. Furthermore, if it can be assumed that

$$f_{e_s} = \sum f_i \tag{5.43}$$

then a sequence of relations can be used that in turn provide all the remaining unknown parameters for the equivalent pipe. It will be shown below that these assumptions hold reasonably well when compared to a simulation using the exact set of pipe sections and that the Colebrook-White version is more accurate than the Hazen-Williams representation even in the equivalent pipe form.

The total length of the equivalent pipe is therefore:

$$L_{e_s} = \sum L_i \tag{5.44}$$

hence

$$D_{e_s} = \left(\frac{8 f_{e_s} L_{e_s} q_{e_s}^2}{\Delta h_{e_s} \pi^2 g}\right)^{0.2} \tag{5.45}$$

$$Re_{e_s} = \frac{4 q_{e_s}}{\pi D_{e_s} \nu} \tag{5.46}$$

$$e_{e_s} = \left(10^{-1/2\sqrt{f_{e_s}}} - \frac{2.51}{Re_{e_s}\sqrt{f_{e_s}}}\right) 3.71 D_{e_s} \tag{5.47}$$

This procedure assumes that q_{e_s} was used as the injected flow for each CHW test carried out in each pipe section in the chain. It is possible that the CHW-values for the different pipe sections could be obtained using different injected flow rates. This means that $\exists i : q_i \neq q_{e_s}$. Since the CHW value is only marginally dependent on Re this effect could be ignored without too much loss in accuracy, particularly as it is unlikely that the injected flows used will be vastly different in each case. Nevertheless, a simple modification is now presented to further minimise the errors emanating from this eventuality.

Assume that each q_j, $(j = 1, \ldots p)$ used in the CHW tests were different. Since it is possible to convert to Colebrook-White form for each of these cases via Eqn. 5.36 and 5.37, for the j^{th} pipe we can obtain a set of independent parameters $\{D_j, \quad L_j, \quad e_j\}$ but f_j computed from Eqn. 5.36 relates to the test flow rather than q_{e_p}. Using a q_{e_p}, arbitrarily chosen from the range of test flows, q_j, $(j = 1, \ldots p)$, the new f_j can be computed directly from Miller's equation (5.31) and Δh_{e_p} can be found using the Darcy-Weisbach equation. If the same q_{e_p} is used for each pipe in the series combination we can compute the f_j's accordingly and use Eqn. 5.43 to provide f_{e_p}. Now the descriptions are compatible and the above procedure, Eqns. 5.44 to 5.47, can be used.

5.8.1.2 Parallel Combination

The procedure for parallel combination follows similar lines and will therefore be dealt with more succinctly than the case for series combinations.

The new hydraulic resistance value K_{e_p} for the Hazen-Williams equation is given by:

$$\frac{1}{K_{e_p}} = \sum_{i=1}^{p} \frac{1}{K_i} \tag{5.48}$$

When finding equivalent pipe parameters for the Colebrook-White equation, a reasonable assumption might be that $L_{e_p} = L_i \quad \forall \quad i$. However, it could easily be the case that this does not hold since two or more pipes may take different routes between the same two nodes. Since this introduces an additional unknown it is necessary to make two (rather than one) *ad hoc* simplifications, namely:

$$L_{e_p} = \frac{1}{p} \sum_{i=1}^{p} L_i \tag{5.49}$$

$$f_{e_p} = \frac{1}{p} \sum_{i=1}^{p} f_i \tag{5.50}$$

The other parameters $\{CHW_{e_p},\ D_{e_p},\ e_{e_p}\}$ follow from these two assumptions as described above. The accuracy of these assumptions is verified in the graphs of Figs. 5.8 & 5.9, the derivation of which is described in the following sub-section.

5.8.1.3 Simulation of Series and Parallel Pipe Combinations

The algorithm used to carry out the comparison for both series and parallel pipe combinations is described as follows:

1 Read the parameters for each individual pipe: $\{L_i,\ D_i,\ e_i\}$

2 Read each injected flow, q_i, assumed used in each CHW_i test;

3 Read the head drop, Δh_i, for each CHW_i test;

4 Calculate each CHW_i, f_i;

5 Calculate the set of equivalent parameters as described above;

6 Test the calculated flows in each equivalent pipe representation against the actual situation using the original parameters;

7 Repeat (6) for a series of Reynold's numbers without varying the equivalent parameters.

There were slight differences in approach in the two cases. In the serial case, the same injected flow was used (on the assumption that all the pipes in series would be tested under the same volumetric flow rate) and hence the design Reynolds numbers followed. In the parallel case, different flow rates were used to ensure the same design point of $Re = 10000$ in each case. However, to keep the flows in the same region for the two tests, similar flow rates were used in the series test to those derived from the parallel test. The kinematic viscosity was assumed to be $\nu = 1.3 \times 10^{-6}\ \ m^2/s$. The pipe parameters were the same in each test and are given in the following table:

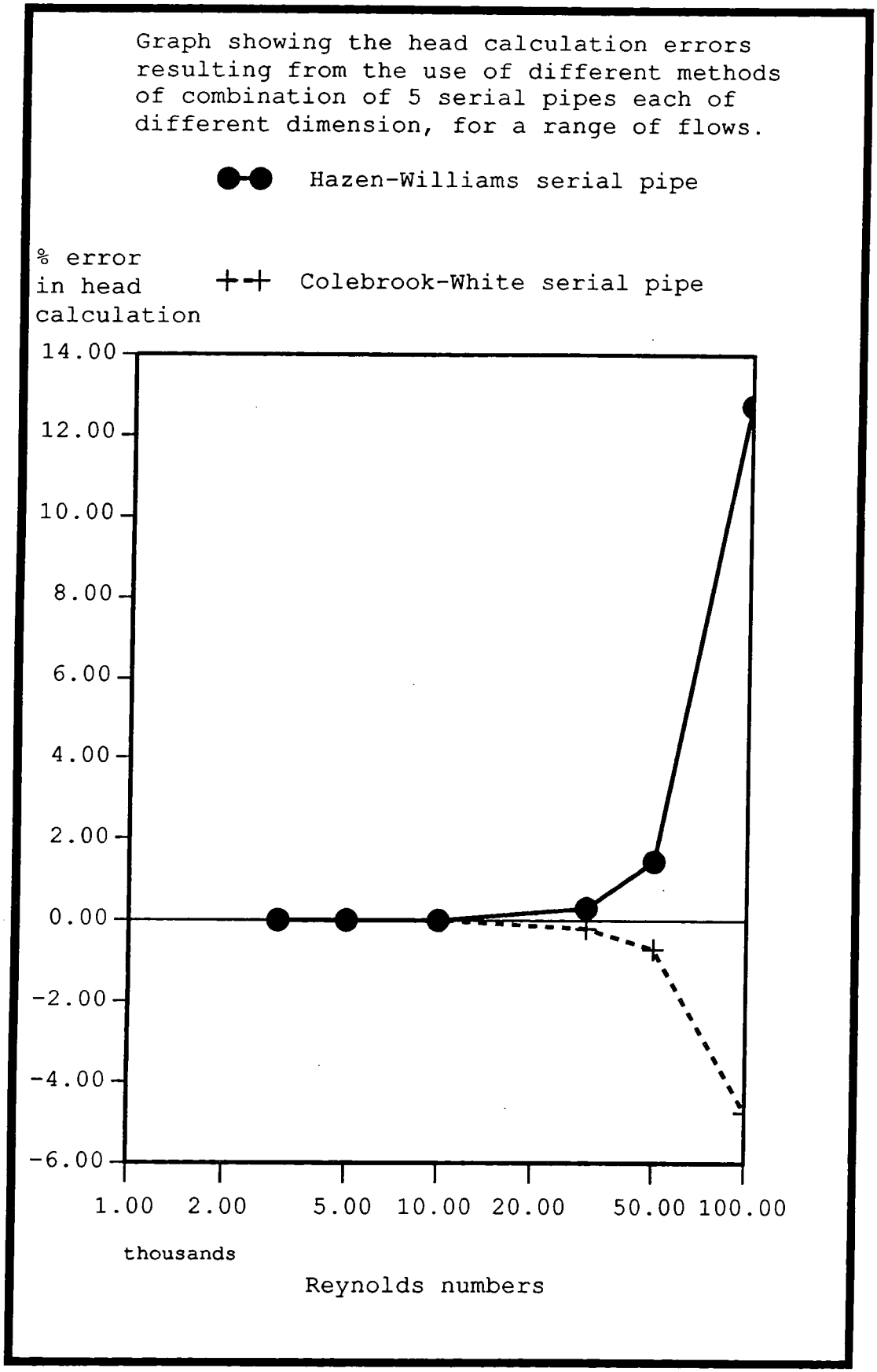

Graph showing the head calculation errors resulting from the use of different methods of combination of 5 serial pipes each of different dimension, for a range of flows.
Hazen-Williams serial pipe
Colebrook-White serial pipe
% error in head calculation
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
-2.00
-4.00
-6.00
1.00
2.00
5.00
10.00
20.00
50.00
100.00
thousands
Reynolds numbers

Figure 5.8

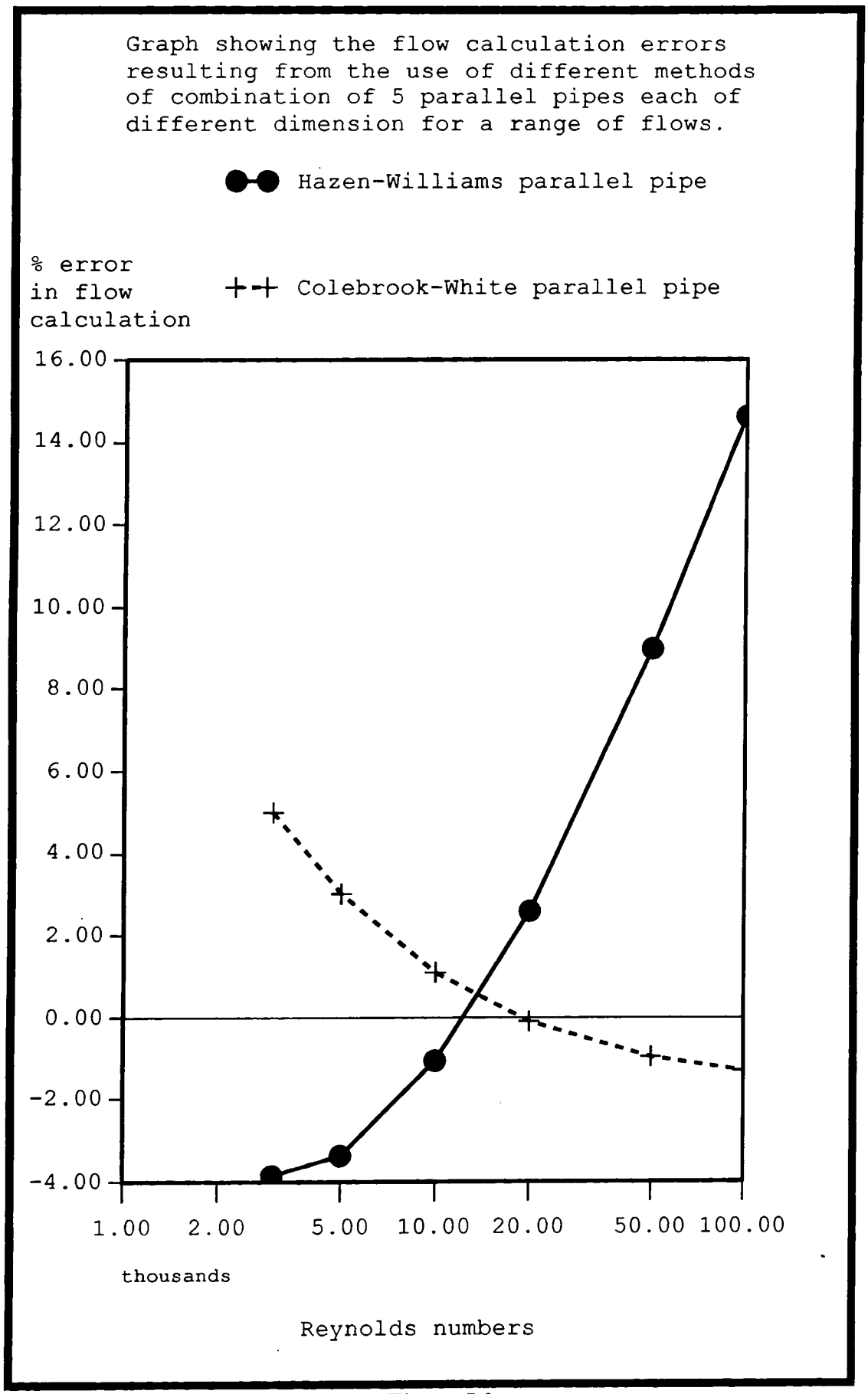

Graph showing the flow calculation errors resulting from the use of different methods of combination of 5 parallel pipes each of different dimension for a range of flows.
Hazen-Williams parallel pipe
Colebrook-White parallel pipe
% error in flow calculation
16.00
14.00
12.00
10.00
8.00
6.00
4.00
2.00
0.00
-2.00
-4.00
1.00
2.00
5.00
10.00
20.00
50.00
100.00
thousands
Reynolds numbers

Figure 5.9

Pipe Parameters				
Pipe No.	Diameter (mm)	Length (m)	Roughness (mm)	CHW –
1	100	500	0.5	120
2	200	500	4.0	100
3	300	500	14.0	84
4	150	500	8.0	82
5	225	500	5.0	98

Table 5.1

In the serial case, the flow was fixed at 2.3 l/s. This produced a new set of CHW values due to the new design Re numbers. The conditions specfic to each test are as follows (the subscript $_s$ for the series test and $_p$ for the parallel test.

Specific Parameters			
Pipe No.	Re_s	CHW_s	q_p (l/s)
1	22526	120	1.02
2	11263	100	2.04
3	7509	86	3.06
4	15018	80	1.53
5	10012	99	2.3

Table 5.2

As can be seen the CHW number is relatively insensitive to the quite wide range of Re numbers. The set of pipe parameters was chosen quite arbitrarily to give a wider range of conditions than would be likely in practice.

Notice that the Hazen-Williams formulation is once again prone to unacceptably large errors. Unfortunately, in the parallel case, errors are present in the Colebrook-White equivalent pipe calculation but these are smaller in general than for the Hazen-Williams equation. In fact, the choice of design point (Figs. 5.6 & 5.7) is important here. If the operating Reynold's number is close to the test Reynold's number (or design point) the errors are small. Hence the combination of pipes that have a large difference in Reynold's number for the same head drop result in large errors due to the implicit assumption in Eqn. 5.50, ie. that the velocities are similar. Although only one such test is presented for each case, it is felt that this is representative and the reader can observe that it is in no way contrived.

5.8.2 Pump Combinations

Consider a set of N fixed speed pumps connected on a manifold as shown in Fig. 5.10 where all the pumps have different but known coefficients. Let us assume that the effect of the combination of pumps on a common manifold is to leave the coefficients of each individual pump curve unchanged. It is desired to treat this set of pumps as one unit and hence to determine one set of coefficients from the available data of the form:

$$H = a' q_T^2 + b' q_T + c' \tag{5.51}$$

where

$$H = h_1 = h_2 = \cdots = h_k = \cdots = h_N \tag{5.52}$$

and

$$q_T = \sum_{k=1}^{N} q_k \tag{5.53}$$

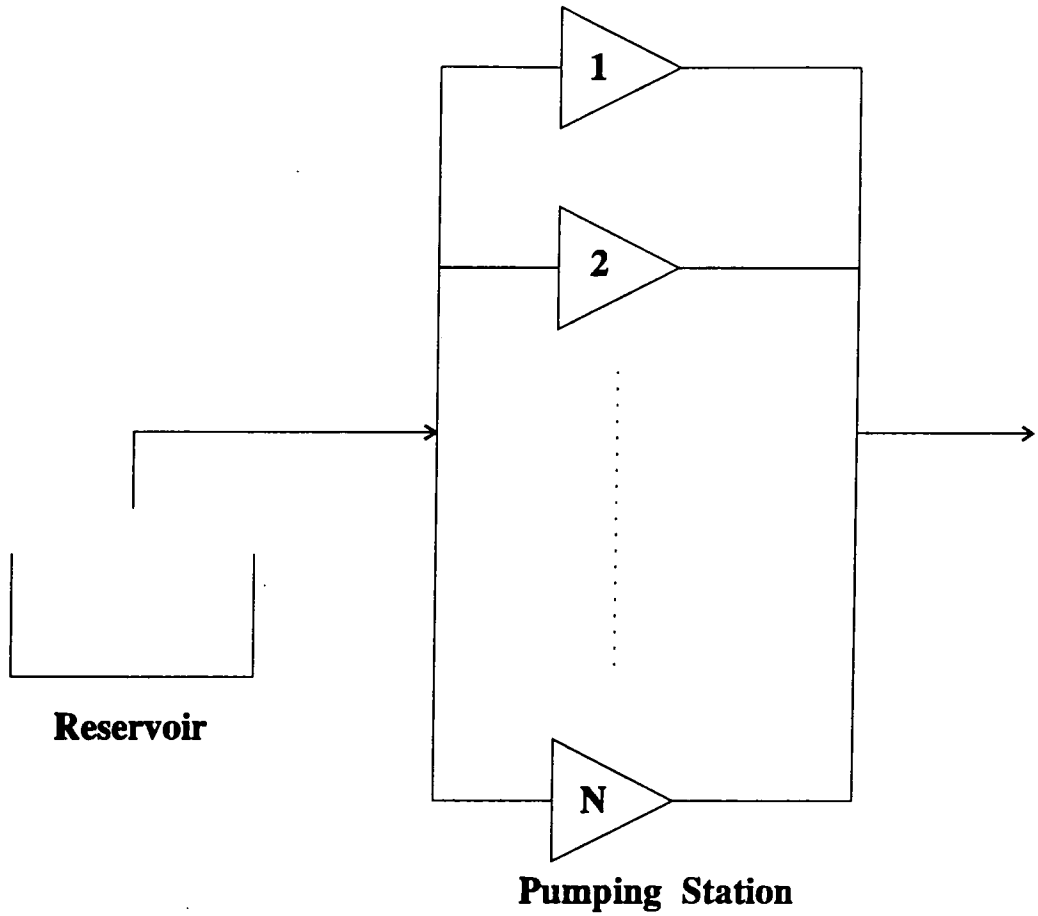


Figure 5.10

Distributed Demands

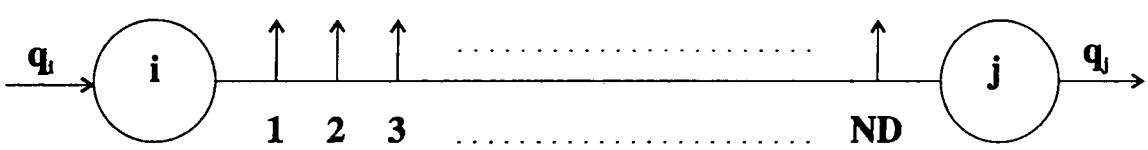


Figure 5.11

The set of pump equations are therefore:

$$h_1 = a_1 q_1^2 + b_1 q_1 + c_1$$

$$\vdots \quad \vdots$$

$$h_k = a_k q_k^2 + b_k q_k + c_k$$

$$\vdots \quad \vdots$$

$$h_N = a_N q_N^2 + b_N q_N + c_N$$

Summing these equations together, we have:

$$\sum_{k=1}^{N} (a_k q_k^2 + b_k q_k + c_k) = N \times H$$

therefore

$$H = \frac{1}{N} \sum_{k=1}^{N} (a_k q_k^2) + \frac{1}{N} \sum_{k=1}^{N} (b_k q_k) + \frac{1}{N} \sum_{k=1}^{N} (c_k) \tag{5.54}$$

which is in the same form as Eqn. 5.51. By comparing coefficients and using Eqn. 5.53, we have the following relations:

$$a' = \frac{\sum(a_k q_k^2)}{N\{\sum(q_k)\}^2} = \frac{\sum(a_k q_k^2)}{N q_T^2} \tag{5.55}$$

$$b' = \frac{\sum(b_k q_k)}{N \sum(q_k)} = \frac{\sum(b_k q_k)}{N q_T} \tag{5.56}$$

$$c' = \frac{\sum(c_k)}{N} \tag{5.57}$$

For each q_k, we can write:

$$q_k = \frac{-b_k \pm \sqrt{b_k^2 - 4a_k(c_k - H)}}{2a_k} \tag{5.58}$$

and hence each new coefficient can be determined in terms of a combination of all the other coefficients and the head rise H.

If $a_1 = \cdots = a_N = a$ and $b_1 = \cdots = b_N = b$ and $c_1 = \cdots = c_N = c$, then from Eqn. 5.58, $q_1 = \cdots = q_N$ and hence we have, as in [47]:

$$a' = \frac{aq^2}{q_T^2}$$

$$b' = \frac{bq}{q_T}$$

$$c' = c$$

The Jacobian terms can easily be determined as follows:

$$q_k = \frac{-b \pm \sqrt{b_k^2 - 4a_k(c_k - H)}}{2a_k}$$

$$\frac{\partial q_k}{\partial H} = \pm\{b_k^2 - 4a_k(c_k - H)\}^{-0.5}$$

The sensitivities of the parallel elements can be written as:

$$\frac{\partial q_T}{\partial H} = \frac{1}{\frac{\partial q_1}{\partial H}^{-1} + \cdots + \frac{\partial q_k}{\partial H}^{-1} + \cdots + \frac{\partial q_N}{\partial H}^{-1}} \tag{5.59}$$

$$= \frac{N}{\sum_{k=1}^{N}(b_k^2 - 4a_k(c_k - H))^{0.5}} \tag{5.60}$$

Unfortunately, the above analysis is not exact due to the inaccuracy of the stated assumption concerning the individual pump coefficients. However, to a first approximation, this pump combination method for similar types of pumps would produce coefficient values that could be used as the starting point for a parameter tuning exercise as described in [201–203] whence more accurate coefficients would result.

In summary, this section has presented methods for combining network elements automatically. The methods are not exact but the approximations are reasonable. The proposed algorithms are designed to allow automatic combination of data held in a database where more than one element is a member of a link. (Where 'link' is defined in Chapter 4).

5.9 Modelling Distributed Consumer Demands

An assumption that is almost always made in network analysis is that consumer demand, which is often distributed along each pipe (e.g. Fig. 5.11), can be represented by lumped demands divided equally between the two end nodes of each pipe link. Whilst this makes sense intuitively and is computationally convenient, it is not clear for what circumstances this assumption holds good and where, if at all, it leads to unacceptable modelling errors. In this section, modelling errors are investigated for several cases where distributed demand is treated as a lumped parameter. The ensuing analysis is by necessity crude, since it is of course impossible to determine what instantaneous demand each consumer will make and in any event improbable that there will be a high correlation between one user's demand pattern and the next. However, on average, the demand from large numbers of consumers will tend to follow a steady diurnal pattern and it is this, rather ill defined behavioural trend, that to some extent justifies the following discussion.

5.9.1 Analysis of Distributed Demand

Consider a pipe element denoted i,j that links nodes i and j in a network system (Fig 5.11) having a length L_{ij}, diameter D_{ij}, inlet flow q_i and outlet flow q_j. The consumption along the pipe, Q_L will be assumed to be distributed in ND equal amounts along the length of the pipe.

This demand could in fact be modelled in a number of ways:

a) uniformly distributed along the pipe;

b) as ND discrete demands;

c) lumped at node i;

d) lumped at node j;

e) lumped at nodes i and j using some allocation function.

In order to understand the errors involved the following analysis will investigate case a) above and then using the notation for discrete analysis, derive a simple result for case b). These two approaches are then compared graphically with cases c), d) and e).

Case (a):

If the load were uniformly distributed along the pipe, the volume flow at any point x along the pipe would be given by:

$$q(x) = Q_{in} - \frac{Q_L}{L}x \tag{5.61}$$

In order to calculate the head drop across an element Δx at some point x in the pipe, we need the average flow in Δx which is given by:

$$\begin{aligned} q_{av}(x, \Delta x) &= \frac{q(x) + q(x + \Delta x)}{2} \\ &= \frac{(Q_{in} - \frac{Q_L}{L}x) + (Q_{in} - \frac{Q_L}{L}x - \frac{Q_L}{L}\Delta x)}{2} \\ &= Q_{in} - \frac{Q_L}{L}x - \frac{1}{2}\frac{Q_L}{L}\Delta x \end{aligned} \tag{5.62}$$

Having obtained the average volume flow it is now possible to determine the average head drop from the Hazen-Williams flow formula:

$$\frac{\Delta H}{\Delta x} = \left(\frac{Q}{0.27746 C H W D^{2.63}}\right)^{1.85} \tag{5.63}$$

For the element under consideration, $L = \Delta x$ and $Q = q_{av}(x, \Delta x)$ and since this is the idealised, uniformly distributed demand case, we allow the approximation $\Delta x \rightarrow 0$, giving:

$$\frac{dh}{dx} = \left(\frac{Q_{in} - \frac{Q_L}{L}x}{0.27746 C H W D^{2.63}}\right)^{1.85} \tag{5.64}$$

Integrating this along the pipe gives:

$$\begin{aligned} h &= \int_0^L \left(\frac{Q_{in} - \frac{Q_L}{L}x}{0.27746 C H W D^{2.63}}\right)^{1.85} dx \\ &= \frac{(-(Q_{in} - Q_L)^{2.85} + Q_{in}^{2.85})L}{2.85 Q_L (0.27746 C H W D^{2.63})^{1.85}} \end{aligned} \tag{5.65}$$

The appearance of Q_L in the denominator of Eqn. 5.65 does not introduce a problem since by application of L'Hopital's rule it can be shown that as $Q_L \rightarrow 0$ Eqn. 5.65 reduces to Eqn. 5.11. A solution of the form $q = f(\Delta h)^{1/\alpha}$ is not obvious from the above due to the non-integer power. However, Moore [180] noted that for old pipes an index of 2 was more appropriate than 1.85. This allows us to write:

$$\frac{\Delta H}{\Delta x} \approx \frac{Q^2}{(0.27746 C H W D^{2.63})^{1.85}} \tag{5.66}$$

(a slight improvement over using the index 2 for the denominator) with solution:

$$h = \frac{10.743 L}{C H W^{1.85} D^{4.87}} (Q_{in}^2 - Q_{in} Q_L + \frac{Q_L^2}{3}) \tag{5.67}$$

A similar but more accurate result can be obtained from the Darcy-Weisbach equation:

$$h = \frac{8 f L^2 q^2}{\pi^2 D^5 g} \tag{5.68}$$

viz:

$$h = \frac{8 f L}{\pi^2 D^5 g} (Q_{in}^2 - Q_{in} Q_L + \frac{Q_L^2}{3}) \tag{5.69}$$

This assumes that f is not a function of the distributed demand in the pipe which of course is untrue. However, provided the Reynolds number stays reasonably high, say, above 4000, then f varies only slightly with Re.

Case (b):

In the case of discrete consumer demands (the actual situation), consider the pipe as shown in Fig 5.11 where the flow continuity at the k^{th} consumer is given by:

$$q_k = q_{k+1} + c_{k+1} \tag{5.70}$$

and for the last consumer in the pipe we have,

$$q_{ND-1} = q_j + c_{ND} \tag{5.71}$$

summing these over the number of consumers in the pipe produces:

$$q_i - q_j = \sum_{1}^{ND} c_k \tag{5.72}$$

Using the Hazen-Williams formula, the head drop for the k^{th} element is given by:

$$h_{(k-1)} - h_k = \frac{10.743 L_k q_k^{1.852}}{CHW_{ij}^{1.852} D_{ij}^{4.87}} \tag{5.73}$$

The head drop in the last link is given by:

$$h_{ND} - h_j = \frac{10.743 L_{ND} q_j^{1.852}}{CHW_{ij}^{1.852} D_{ij}^{4.87}} \tag{5.74}$$

In many streets, houses are spaced equally along the pipe and so we might make the further approximation that:

$$L_1 = L_2 = \cdots = L_{ND+1} = \frac{L_{ij}}{ND+1}$$

Therefore we have,

$$h_i - h_j = \frac{10.743 L_{ij}}{CHW_{ij}^{1.852} D_{ij}^{4.87}} \sum_{k=1}^{ND+1} q_k^{1.852} \tag{5.75}$$

where

$$q_k = q_i - \sum_{p=1}^{k-1} c_p \tag{5.76}$$

hence

$$h_i - h_j = \frac{10.743 L_{ij}}{CHW_{ij}^{1.852} D_{ij}^{4.87}} \sum_{k=1}^{ND+1} (q_{in} - \sum_{p=1}^{k-1} c_p)^{1.852} \tag{5.77}$$

Furthermore, if, $\forall p \; c_p = c$ then:

$$h_i - h_j = \frac{10.743 L_{ij}}{CHW_{ij}^{1.852} D_{ij}^{4.87}} \sum_{k=1}^{ND+1} (q_{in} - (k-1)c)^{1.852} \tag{5.78}$$

5.9.2 Schemes for Considering Consumer Demand

Let K be defined as in Eqn. 5.14, then the various methods for lumping demand can be described as follows:

1) Demand lumped at node i:

$$h_i - h_j = K_{ij}(q_{in} - Q_L)^{1.852} \tag{5.79}$$

2) Demand lumped at node j:

$$h_i - h_j = K_{ij} q_{in}^{1.852} \tag{5.80}$$

3) Demand lumped equally at nodes i and j:

$$h_i - h_j = K_{ij}(q_{in} - \frac{Q_L}{2})^{1.852} \tag{5.81}$$

4) Demand modelled as in Eqns. 5.67 and 5.69.

5.9.3 Results of Different Distributed Demand Methods

Using these results, an analysis of the sensitivity of pressure calculation can be made by considering the various assumptions that are commonly made in pipe flow calculation. The first such analysis considers the pressure error variation with respect to a variation in the proportion of demand in the pipe compared to the total inlet flow:

To make a comparison of the above formulæ with Eqn. 5.65, let:

$$H_0 = K_{ij} \sum_{k=1}^{ND+1} (q_{in} - (k-1)c)^{1.85} \tag{5.82}$$

$$H_1 = K_{ij}(q_{in} - Q_L)^{1.852} \tag{5.83}$$

$$H_2 = K_{ij} q_{in}^{1.852} \tag{5.84}$$

$$H_3 = K_{ij}(q_{in} - \frac{Q_L}{2})^{1.852} \tag{5.85}$$

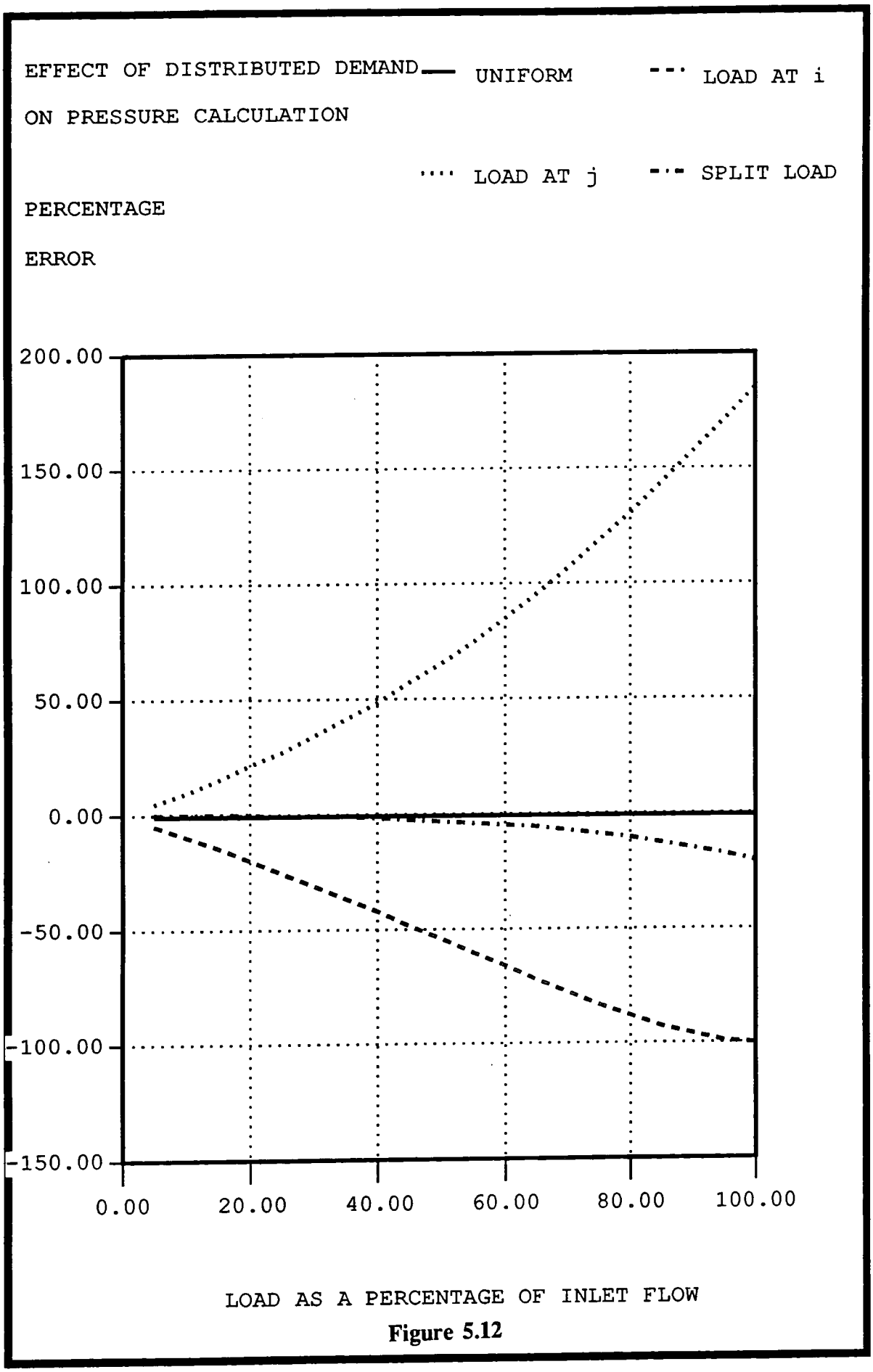

EFFECT OF DISTRIBUTED DEMAND
ON PRESSURE CALCULATION
UNIFORM
LOAD AT i
LOAD AT j
SPLIT LOAD
PERCENTAGE
ERROR
200.00
150.00
100.00
50.00
0.00
-50.00
-100.00
-150.00
0.00
20.00
40.00
60.00
80.00
100.00
LOAD AS A PERCENTAGE OF INLET FLOW

Figure 5.12

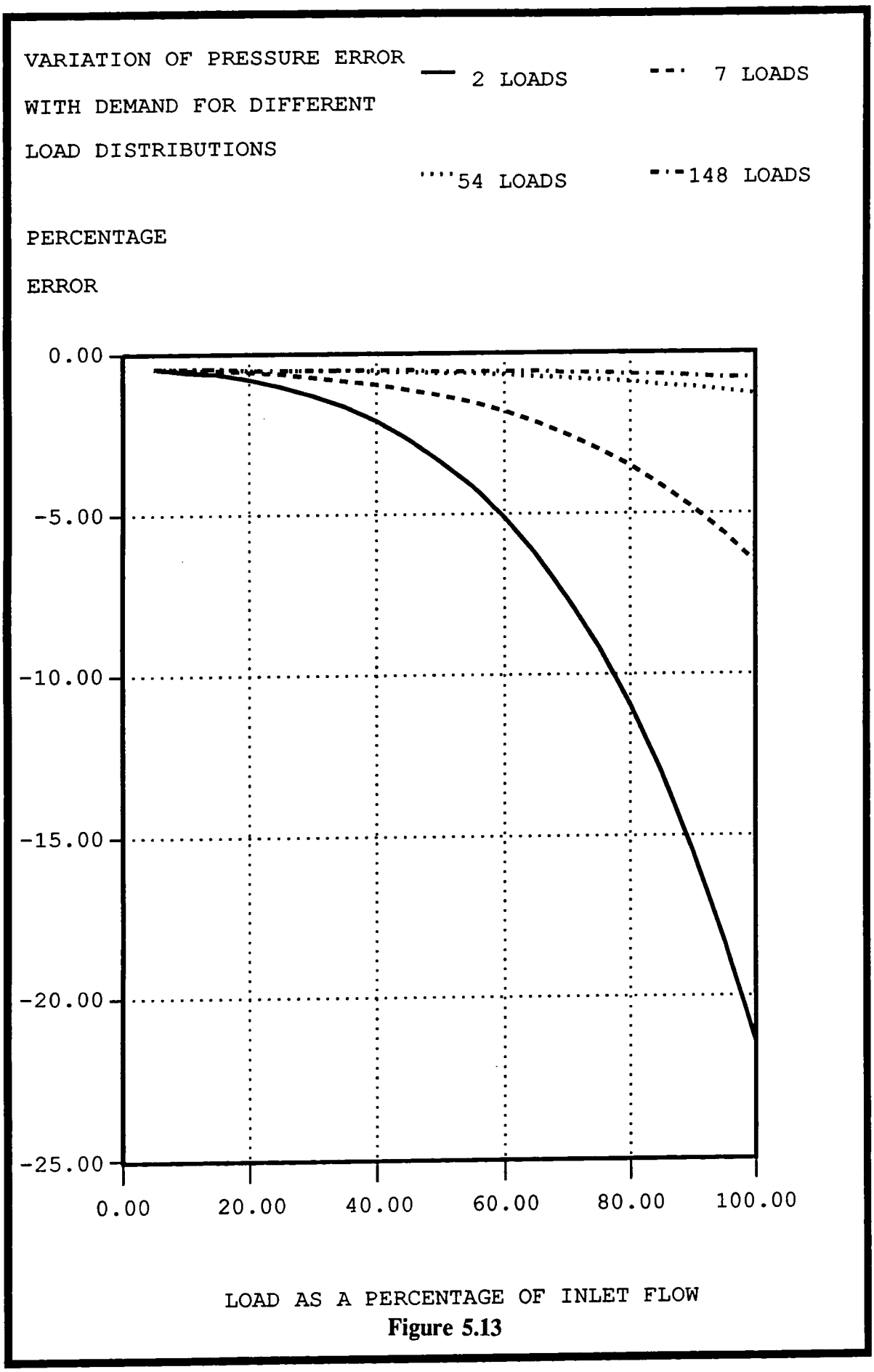

VARIATION OF PRESSURE ERROR
WITH DEMAND FOR DIFFERENT
LOAD DISTRIBUTIONS
2 LOADS
7 LOADS
54 LOADS
148 LOADS
PERCENTAGE
ERROR
0.00
-5.00
-10.00
-15.00
-20.00
-25.00
0.00
20.00
40.00
60.00
80.00
100.00
LOAD AS A PERCENTAGE OF INLET FLOW

Figure 5.13

EFFECT OF VARYING THE RATIO OF LOADS LUMPED AT NODE i AND j ON THE ERROR IN THE PRESSURE CALCULATION

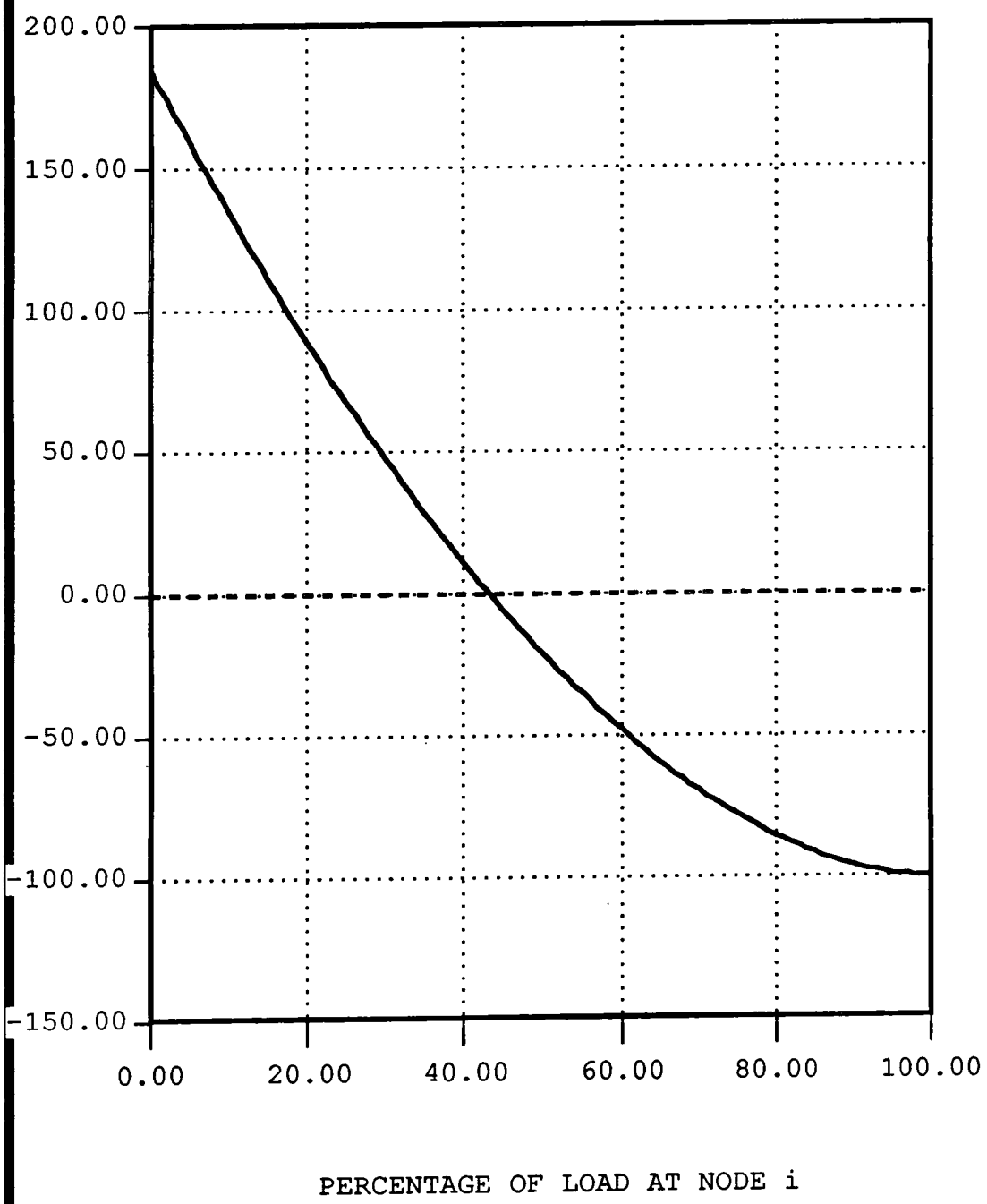


Figure 5.14

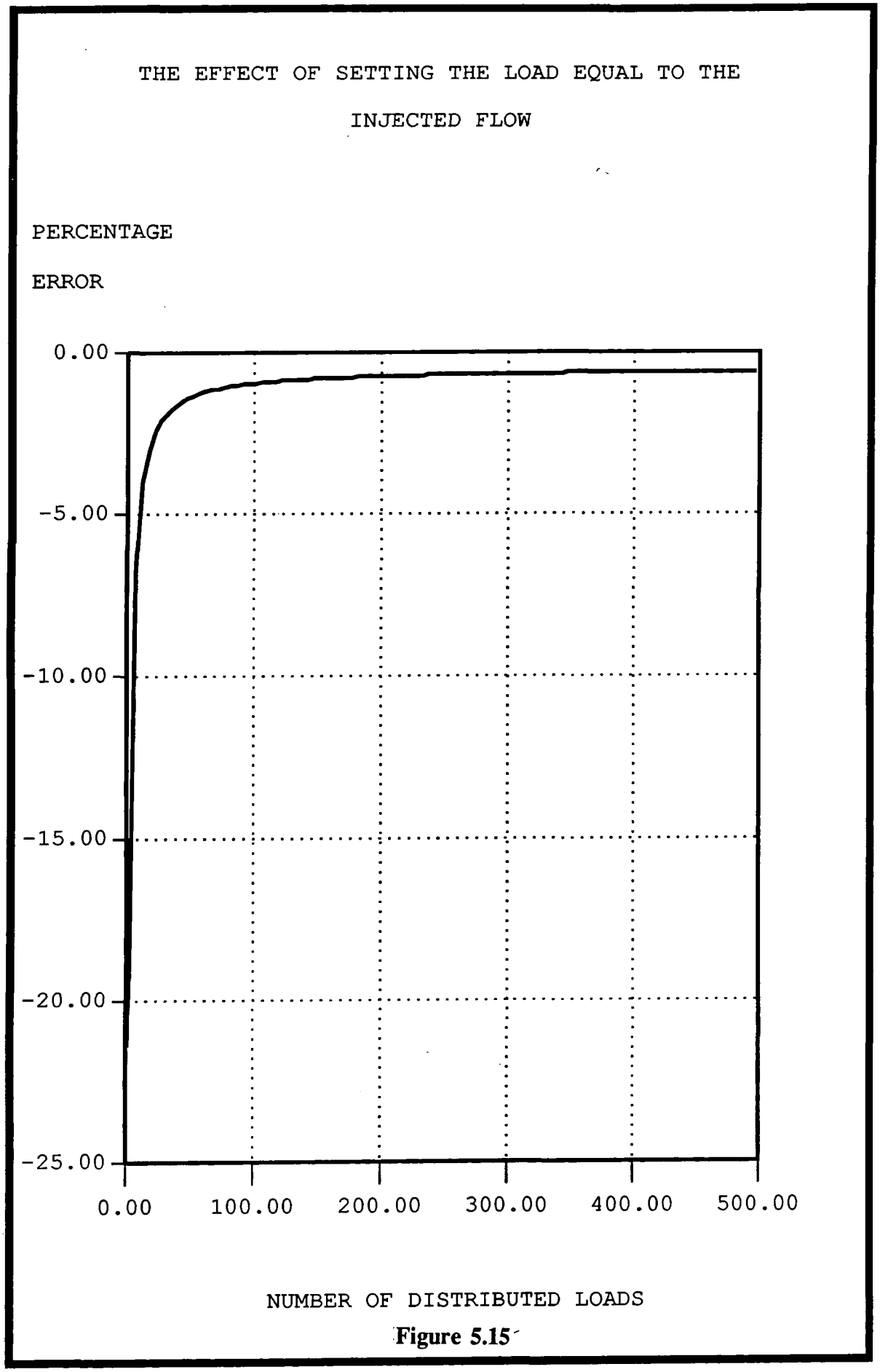

THE EFFECT OF SETTING THE LOAD EQUAL TO THE
INJECTED FLOW
PERCENTAGE
ERROR
0.00
-5.00
-10.00
-15.00
-20.00
-25.00
0.00
100.00
200.00
300.00
400.00
500.00
NUMBER OF DISTRIBUTED LOADS

Figure 5.15

1) The variation of error in the pressure calculation as the ratio of demand to inlet flow is increased for each of the different modelling methods is shown in Fig. 5.12.

2) Fig. 5.13 shows the converse situation when the physical system has a known (small) number of nodes but the model is assuming an evenly distributed load pattern.

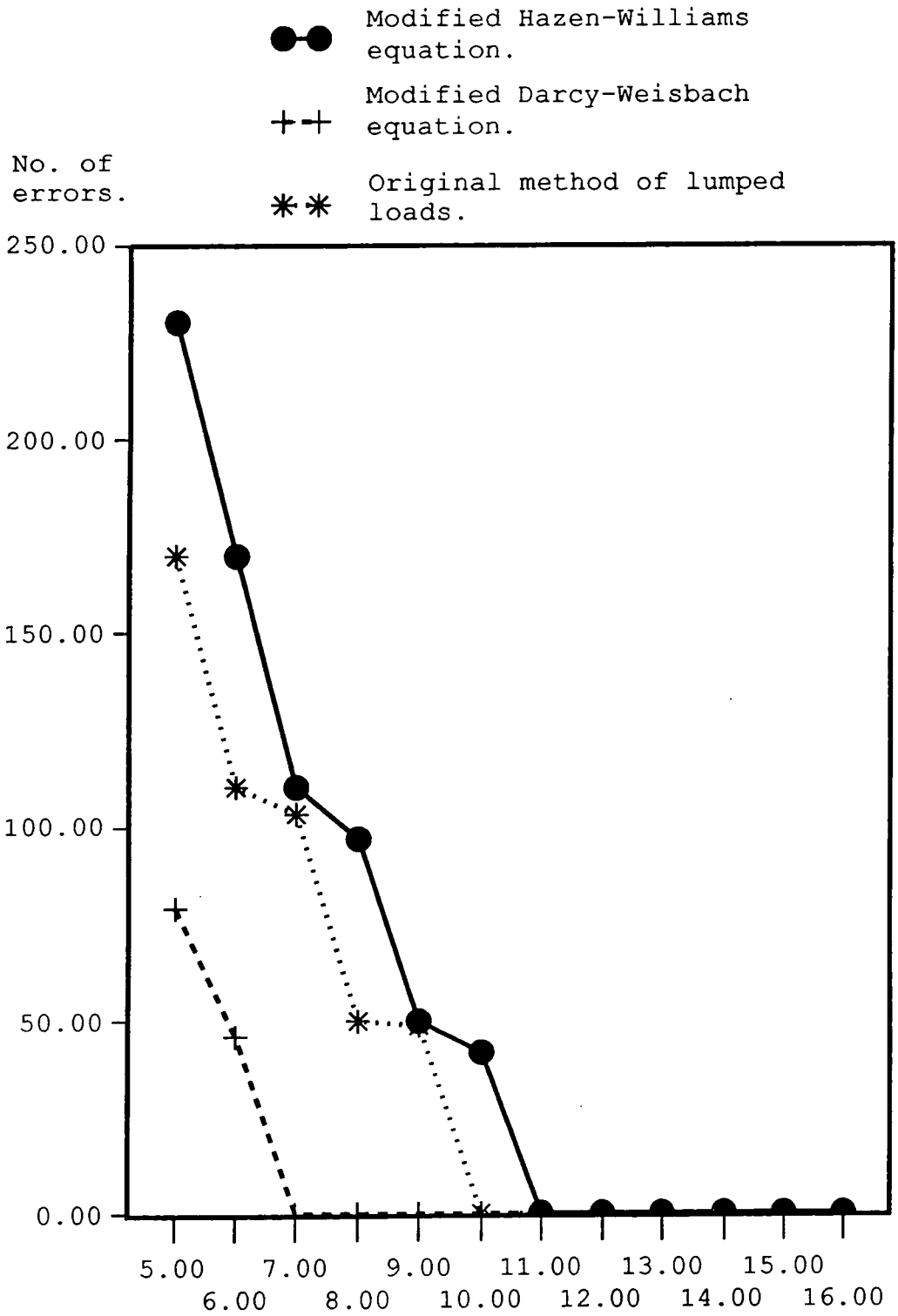
3) The results in Fig. 5.14 show the effect of continuously varying the load ratio (ie the ratio of two loads lumped at nodes i and j) across a pipe with an evenly distributed load. The zero case is achieved when node i has 42% of the load and node j 58%.

4) The largest errors come about when the injected flow is equal to the total demand. Fig. 5.15 shows how this effect is diminished when the actual system has a distributed load and the modelled system has an increasing number of loads.

The models of distributed demand, Eqns. 5.67 & 5.69, have been tested against the method of lumping demand described in Eqn. 5.81. The method used was to divide the pipe into a series of short pipes in cascade and model the flow using the Colebrook-White formula with demands from each junction. A wide variety of conditions were imposed by varying the total number of consumers ND in the range $1 \leq ND \leq 1000$; the demand magnitude in the range $0 \leq Q_L/Q_{in} \leq 0.5$; the diameter in the range 0.1m $\leq$ D $\leq$ 1.0m; and the roughness in the range 0.001m $\leq$ e $\leq$ 0.01m. The length was fixed at 1000m. With Q_{in} in the range $0.001\text{m}^3/\text{s} \leq \text{Q}_{\text{in}} \leq 0.05\text{m}^3/\text{s}$, this produced a wide range of Reynolds numbers over 3000.

The results are shown in Figs. 5.16 & 5.17. The tests outlined above produced the error magnitudes as the difference between the pipe modelled with discrete demands along its length and the three simplifications already discussed. In Fig 5.16 there is a

Graph showing the relative accuracy of three methods of calculating the head loss in a pipe with a distributed demand. (All demands are assumed equal.)



Error in head calculation as a percentage difference with a known reference value.

Figure 5.16

Graph showing the relative accuracy of three methods of calculating the head loss in a pipe with a distributed demand. (Individual demands have a Poisson distribution.)

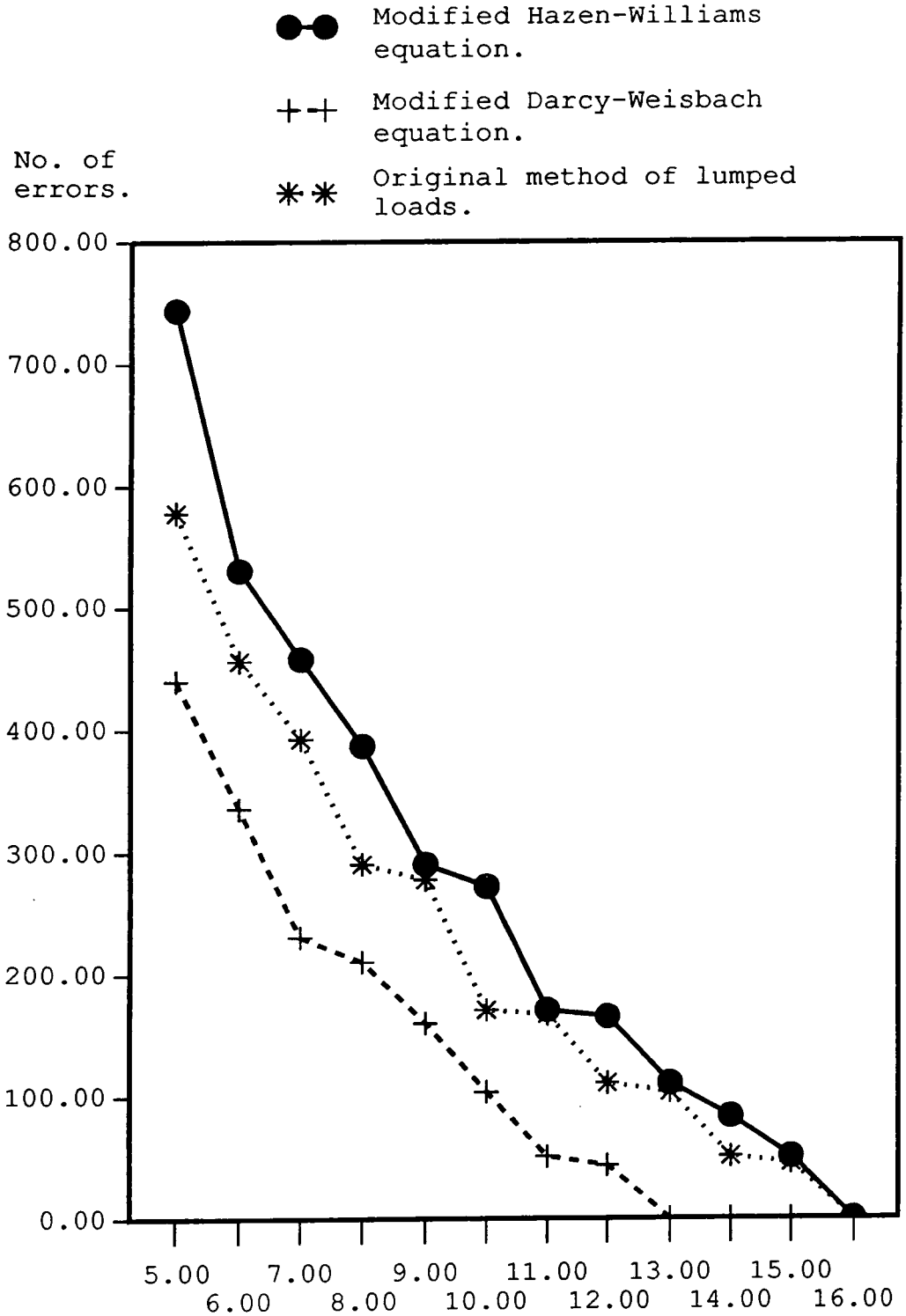


Figure 5.17

uniform demand along the pipe, whereas Fig. 5.17 is for the case where the demand is modelled as a random variable, as descibed below.

Each consumer was modelled according to the following formula:

$$c_i = \beta_i \times \frac{Q_L}{ND} \tag{5.86}$$

where β is a random variable belonging to a Poisson distribution with a mean of 1.0. This was chosen so as to simulate the behaviour of consumers drawing water from their water main, such that the sum of their average demand corresponded with the total load, Q_L, but which permitted a variation ranging from no demand at all to a larger than average demand. This test produced too large a set of results to conveniently present in this thesis but it is hoped that the underlying conditions were varied enough to give significant insight to the full range of accuracies likely to result if any of the methods proposed here were adopted.

The biggest improvements were gained by the use of the Darcy-Weisbach distributed demand model, Eqn. 5.69 instead of the Hazen-Williams lumped demand representation. However, only small improvements are available by switching from the standard lumped Darcy-Weisbach equation to the new form and so there remains a good case for continuing with the lumped demand method, Eqn. 5.81. Nevertheless, since improvements in accuracy are possible with a simple modification to the Darcy-Weisbach formula, this might just as well be used. It is can be seen from the results that single point demands or small numbers of demands produce the biggest deviation from the proposed model, whereas when the number of demands rises to 100 or more the accuracy improvement becomes more marked.

5.10 Building a Network Model

One of the main objectives of this chapter is to show that it is possible to model distribution networks from data stored in a form that can easily be maintained. The reason that this is proposed here is that, as has been pointed out earlier, the thesis aims to show that it is possible to design a system that can be used for on-line monitoring in a control room environment where the model used for measurement validation purposes must replicate the real system in the ground and, furthermore, that it is possible to manually change the model, on-line, without the need for an understanding of network analysis procedures. In this chapter and in the previous two, the necessary tools for deriving a network model from asset data have been described. All that is left to cover is some of the details of the order in which the data is accessed and how it is used in the simulation process.

5.10.1 Outline of algorithm

1) Retrieve the current topology. This method has been fully explained in the previous chapter.

2) Combine all serial elements. By building the model using the link number as the loop counter, pipes with duplicated link numbers can be detected. Then, using an algorithm similar to that given in Section 4.4.2 which assigned alias numbers to nodes which were part of the same node cluster, the new parameters are calculated.

3) Combine all parallel elements (including those resulting from 2). Similarly, it is easy to find shared node number pairs to detect parallel pipes. The combination algorithm is again similar to the one outlined in 2.

4) Check for existence of islands without sinks or sources. This is done by assigning the island number to each element in the model. Any island without a source pump or a reservoir is not considered to have a supply. (An exception to this is sometimes encountered in network analysis when a network is considered that receives its supply from 'let down' valves from a higher pressure zone.)

5) Spawn each island simulation (or state estimation) as separate process. This can easily be done if the network build process is capable of initiating batch sub-processes. (Such features are computer operating system specific.)

Note: From the Chapter 3, links were defined as unique paths joining two nodes. A pipe cannot have two link numbers although two (or more) pipes can share a link. Hence links containing sets of pipes in series can be in parallel but not the converse. A set of pipes is only considered in series for combination if they all share the same link and a set of links must share the same end nodes if they are to be combined. All that is needed if there exists between two nodes a topology that does not comply with the imposed restrictions is to include the ignored nodes.

5.10.2 Existence of Solution to the Network Equations

The methods presented thus far do not guarantee that a particular mathematical model automatically built actually has a solution. Conversely, we can say that undetectable (in the model construction sense) pathological cases could exist. Whilst some 'shallow' attempt has been made to check for illogical model formulations, such as networks with no sources or no demands, no 'deep' method has been incorporated into the retrieval and build procedures. However, it should be remembered that the techniques proposed here are for use in an on-line monitoring application for a network which has been recently analysed. It is tacitly assumed that prior to configuring an on-line monitoring scheme, extensive network analyses would be carried out and that any topological

change of a network currently being monitored, in the way proposed here, would simply imply a change from one feasible solution to another. It is therefore the case that network analysis is a prerequisite to on-line monitoring using the proposed method.

The main difference between simulation and state estimation, algebraically speaking, is that simulation involves the solution of a square, consistent set of n linearised equations of the form:

$$\mathbf{Ax} = \mathbf{b}$$

for which a unique solution, $\mathbf{x}$, exists provided $\mathbf{b} \in \mathcal{R}\{\mathbf{A}\}$ (i.e. $\mathbf{b}$ is in the column space of $\mathbf{A}$). Whereas, in the state estimation problem, the solution involves an overdetermined set of linear equations ($m \times n$, $m \geq n$) which are permitted to be inconsistent, ie $\mathbf{b} \notin \mathcal{R}\{\mathbf{A}\}$. Strictly speaking, no solution exists for this case but our intuition tells us that a 'best fit' solution is probably more reliable than an 'exact' solution based on physical measurements as in the simulation case. State estimation uses the projection of $\mathbf{b}$ onto $\mathcal{R}\{\mathbf{A}\}$. The exact form of this projection is the subject of the next chapter but, even at this stage, it is possible to see that state estimation will tolerate inconsistency which is not possible in simulation. Formal proofs of existence and uniqueness of the solution to the state estimation problem (for power systems, in fact, although the constructions are quite general) are given in [43] but a better introduction can be found in [246].

Notwithstanding this, the introduction of new network elements can cause convergence difficulties. For instance, discontinuous variables such as level, flow or pressure switched valves or pumps, although piecewise differentiable, can introduce a non-smooth transition between linearisation points resulting in convergence problems.

Another cause of computational problems is ill-conditioning. For example, very low head drops can result from oversized pipes even carrying significant quantities of water. A small error in the calculation of the head drop can mean that the flow calculation

might carry a very large absolute error. Such systems could be said to be *physically* ill-conditioned hence increasing the likelihood that the model will be *numerically* ill-conditioned. Monticelli et al [177] discuss some of the causes of the ill conditioning problem for a power system from the point of view of setting up the state estimation equations using different types of measurement.

This subject will be discussed a little further in the next chapter. The problem of providing diagnostic information is of importance to practical implementations of on-line model construction and is left for further work.

5.11 Conclusion

This chapter has concentrated on the techniques for mathematically modelling the elements in distribution systems and how the collection of the elements can be built into one integrated system of equations suitable for real-time on-line monitoring of water supply networks.

Pipe and pump combination is further justified by the likelihood that, on average, the effect of measurement errors in the determination of the original element's parameters will be canceled out. The more elements that are combined in the way outlined in this chapter, the less the errors in the original parameters will effect the flow model. This procedure could therefore produce a more accurate model which should not seem surprising since a larger amount of information is being processed than with more conventional methods of network analysis.

The general assumptions concerning modelling distribution networks have been explored in considerable depth which has enabled an assessment to be made about the implications in terms of the accuracy of some of these assumptions.

The modelling function is the interface between the physical data on the one hand and the monitoring system on the other. A central theme of this thesis is the idea of providing a simple transition from a network change to a new network model. This is achieved by automating as much of the model building function as possible by exploiting the flexibility of the relational database and using algebraic methods for element combination where necessary.

CHAPTER 6

STATE ESTIMATION OF WATER DISTRIBUTION SYSTEMS

6.1 Introduction

In order to control a time varying system, a set of measurements relating to the system's current state is needed. The effectiveness of the actuating signals derived from the control objective function will depend upon the validity of the measurements received in the measurement process. If those measurements are inaccurate or, in the case of large scale systems where local loop control is inappropriate, if the system model does not reflect the physical system, then it is impossible to guarantee that the control action will be optimal even if an 'optimal' control algorithm had been implemented.

The complexity and size of water distribution systems coupled with the statutory levels of service and the cost of pumping and treatment, leads to a requirement for at least some type of control. In certain cases, designated areas can be controlled by the local loop method [198] but there is an increasing interest in implementing full scale optimised control of complete supply systems. It is possible to control a large scale water distribution system in a quasi-closed loop scheme using network simulation (rather than state estimation), [71], but since, without the use of a state observer, data validation is not guaranteed to be optimal under any circumstances, it is impossible to determine the real difference between the simulated state and the actual state of the system. Lack

of validated observation data restricts the use of an optimal control algorithm since the system's state is strictly unknown and therefore optimal control can never be guaranteed. Furthermore, it is not even possible for an operator to be certain of the effectiveness of any control actions carried out, optimal or otherwise.

The purpose of this chapter is to continue the theme of on-line monitoring of water distribution systems by introducing the techniques of state estimation as a method for observing the state of a water network on-line and in real time. State estimation is defined here as the computation of the minimum set of variables, $\hat{\mathbf{x}}(\mathbf{t}_k)$, necessary to completely describe all other pertinent variables at the k^{th} time interval for a given system from some measurement data $\mathbf{z}(\mathbf{t}_k)$. The method relies heavily on having an accurate parametric model of the system in question and, for on-line applications, the model is required to be constantly up-to-date. Since the problems posed by this need to maintain a complicated parametric model at all times have been, in the most part, solved by the methods presented earlier, it is now relevant to consider the design of a robust scheme for state estimation such as might actually be used in water network on-line monitoring.

The chapter is organised as follows: Section 6.2 reviews the current research into algorithms for state estimation for utility networks; Section 6.3 presents a new state estimation algorithm designed specifically for this application which is shown to be more suitable than the currently available techniques. The results of the investigations are presented in Section 6.4 and the conclusions are drawn in Section 6.5.

6.2 Review of Recent Work

The use of state estimation techniques covers many application areas and as a consequence, different algorithms abound. All such methods fall into one of two categories, either static (including tracking) or dynamic state estimation. The related areas of parameter estimation and statistical estimation are broader still and provide the foundation

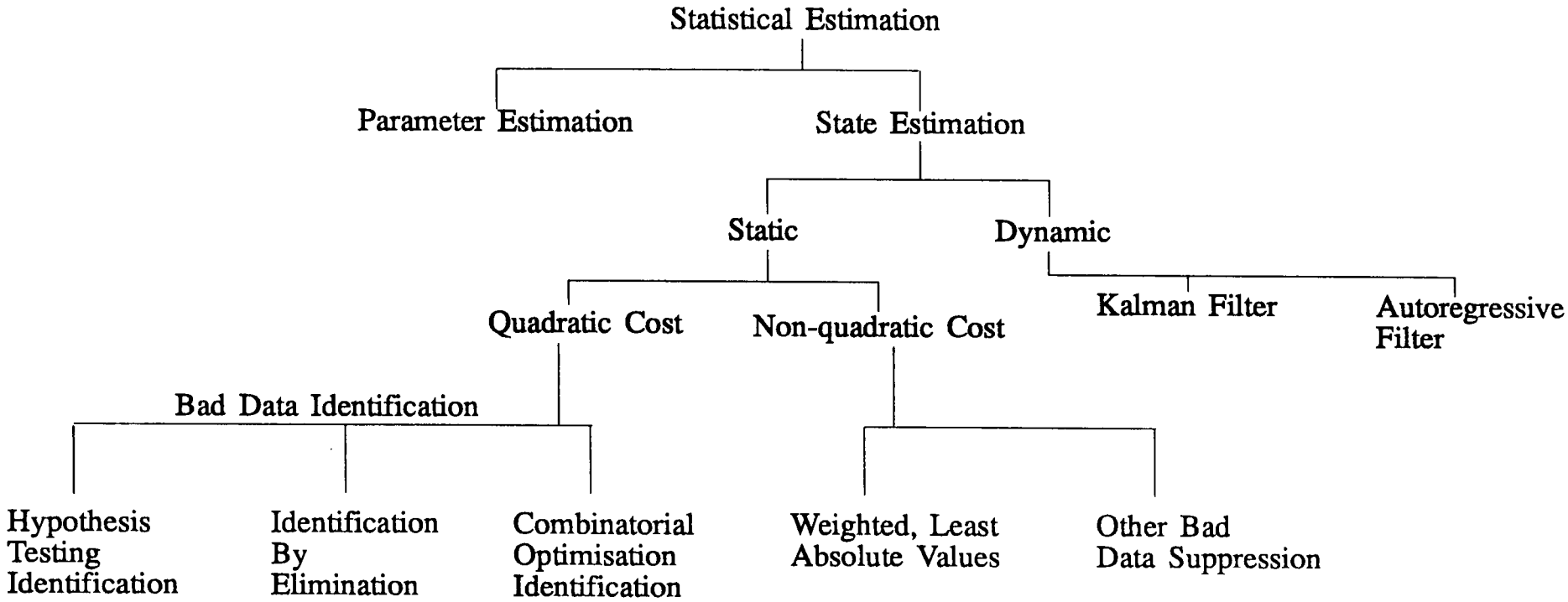

Statistical Estimation
Parameter Estimation
State Estimation
Static
Dynamic
Quadratic Cost
Non-quadratic Cost
Kalman Filter
Autoregressive Filter
Bad Data Identification
Hypothesis Testing Identification
Identification By Elimination
Combinatorial Optimisation Identification
Weighted, Least Absolute Values
Other Bad Data Suppression

Fig. 6.1

upon which the techniques reviewed below are based. The diagram of Fig. 6.1 shows the hierarchy of different types of state estimation algorithms and their relationship to each other. In the review that follows, most of the nodes in this conceptual tree will be visited.

6.2.1 Static State Estimation

The application of any form of on-line state estimation in water distribution systems has so far to happen, although at least two off-line studies have been carried out [38, 196]. A directly analogous area is that of power system state estimation where there has been considerable interest over the last two decades.

In power system state estimation by far the most attention has been paid to the class of state estimator known as the static (or sometimes 'tracking') state estimator. In this class the algorithms broadly sub-divide depending on the type of generic formulation of the cost function, either non-quadratic, e.g. weighted least absolute values (WLAV) or quadratic, e.g. weighted least squares (WLS). Within these divisions, different algorithms such as linear programming, non-linear programming and unconstrained optimisation are used. Presented below is a review of the currently used techniques that fall under these headings.

6.2.1.1 WLS Static State Estimation

The statistical properties of estimators have been the subject of much investigation [129, 140, 170, 227] and of central interest is the proof that they provide unbiassed state estimates. Early papers by Dopazo, Klitin and Sasson [98], and Aboytes and Cory [3] present an introduction to WLS static state estimation by investigating the bad data rejection properties of the weighted least squares technique. A similar approach to that taken in [98] is presented in Appendix 2 which shows that the WLS estimate is unbiassed.

Also in Appendix 2 are some of the basic statistical properties of WLS estimation found throughout the literature on the subject.

Carpentier and Cohen, [38, 66], advocate a pure WLS technique for state estimation in water distribution systems but report significant problems with the detection and identification of errors that may lead to leak detection capabilities. However, a purely least squares solution is only optimal if the errors in the measurement vector are $N(0, \sigma^2)$ ie that

$$\mathrm{E}\{\epsilon\} = \underline{0} \tag{6.1}$$

$$\mathrm{E}\{\epsilon\epsilon^{\mathrm{T}}\} = \mathbf{R} \tag{6.2}$$

where E is the expectation operator, ϵ represents the measurement noise vector and $\mathbf{R}$ is the (diagonal) noise covariance matrix with diagonal components:

$$r_{ii} = \sigma_{ii}^2 \tag{6.3}$$

However, these assumptions are normally not realistic as will be discussed below in Section 6.3 and in Chapter 7. The reason is of course that Eqns. 6.1 and 6.2 do not allow for the existence of gross errors in the data. Gross measurement errors, in keeping with the published literature, are called 'bad data', despite the fact that 'measurement outliers' would seem a more apt term. The main problem posed by the existence of bad data is to design an estimator that continues to produce unbiassed estimates from the point of view of the 'good data' ie. estimates which are not effected by the existence of bad data. Unfortunately, in most cases, before the bad data can be detected, the estimates have to be calculated and therefore are biassed by the bad data.

Aside from crude logical filtering of data to remove errors $> 50\sigma$, one of the most straight forward methods of bad data elimination is that of 'pre-cleaning' [108]. This is a pre-estimation technique designed to identify and remove large errors, e.g. $5\sigma - 10\sigma$ [115]. One method of pre-cleaning is to assume that the value of a measurement will

not change dramatically if the measurements in the neighbouring nodes do not change [162]. Doraiswami [115], on the other hand uses a base case where there are no errors to calculate the Jacobian and compares a function of the solution vector thus obtained with the current measurements. A difference higher than a specified tolerance indicates the presence of new bad data. A disadvantage with this method is that it is only valid where there has been no topological change since the base case was recorded and setting up a new base case for a changed system would be non-trivial. The advantage is that the bad data become known prior to executing the estimation algorithm, an idea which has been used in a different way in other forms of bad data detection, as will be explained below. A network search technique that also avoids the state estimation step is provided by Bonami and Gramberg [27]. This has the advantage of calculating the state using a linked list approach and is therefore computationally simple. Their method is better suited to the detection of topological changes (ie changes of circuit breaker status) than bad data detection.

Time series analysis [116] and exponential smoothing techniques [107] have also been investigated and these properly belong to the pre-estimation class of methods. Both techniques rely on the fact that the network states are not changing rapidly in time and any large instantaneous excursions are probably due to bad data. In a similar way, dynamic state estimation has also been investigated using Kalman filtering [155, 186]. Abur et al [4] use autoregressive filters coupled with WLS to detect and identify bad data in a pre-estimation scheme.

Algorithms (other than those described above) capable of detecting and identifying bad data can be categorize into four types: Hypothesis Testing Identification (HTI); Identification By Elimination (IBE); Combinatoric Optimisation Identification (COI); and Non-Quadratic Criteria (NQC). HTI, IBE and COI are procedures carried out after the WLS algorithm has converged and are therefore classed as post-estimation methods,

whereas NQC is part of the estimation process itself. Nonetheless, some researchers use a combination of these techniques, e.g. [217, 221, 270, 271].

Dopazo et al [98] use the fact that if there is no bad data, $C(\hat{\mathbf{x}})$ (the sum of the squared residuals) is a χ^2 random variable with $m-n$ degrees of freedom and show that bad data can be detected by observing when this cost function minimum is greater than some predefined quantity, β with certainty α. This is called the $C(\hat{\mathbf{x}})$ test†.

A conceptually simple but computationally inefficient method for bad data identification following a positive detection is to perform sufficient re-estimations, systematically eliminating each measurement or measurement pair or triple etc., until the $C(\hat{\mathbf{x}})$ test is negative. Schweppe et al [216] use a more efficient version of this (IBE) technique for bad data identification by using the list of largest weighted or normalized residuals (r^W and r^N respectively) as a guide to the location of the bad data. Handschin et al [134] also note that the $C(\hat{\mathbf{x}})$ test could be replaced with the r^W or the r^N tests as the basis for an NQC estimator. In the r^W test, a set of weighted residuals:

$$r_i^W = \left| \frac{r_i}{\sigma_i} \right| \tag{6.4}$$

are examined for the presence of abnormally high values and if any are detected the list is ordered and used to identify bad data on the assumption that the largest $|r_i^W|$ corresponds to the largest measurement error. The normalized residual test, is identical in form except that normalized residuals are used instead of weighted residuals, each normalized residual being calculated as follows:

$$r_i^N = \frac{r_i}{\sqrt{p_{ii}}} \tag{6.5}$$

where p_{ii} is the i^{th} diagonal element of $\mathbf{P}$,

$$\mathbf{P} = \mathbf{R} - \mathbf{J}\Sigma\mathbf{J}^T \tag{6.6}$$

† In power systems literature it is conventional to call this the $J(\hat{\mathbf{x}})$ test, whereas here, we reserve $\mathbf{J}(\hat{\mathbf{x}})$ for the Jacobian calculated from the set of state estimates, $\hat{\mathbf{x}}$.

and

$$\mathbf{\Sigma} = (\mathbf{J}^T \mathbf{R}^{-1} \mathbf{J})^{-1} \tag{6.7}$$

($\mathbf{P}$ is the covariance matrix of the residuals and $\mathbf{\Sigma}$ is the covariance matrix of estimation errors, as derived in Appendix 2.) The r^N test is always more reliable than the $C(\hat{\mathbf{x}})$ test in the single bad data case but only generally more reliable in the multiple (interacting) bad data case as several authors have shown [54, 58, 172, 173].

Clearly there is a computational overhead in computing the normalized residuals that is not manifest in the calculation of the weighted residuals. However, it is widely accepted that bad data identification on the basis of the r^W test leads to unreliable results in all but the single bad data case [172, 173]. Bargiela [12] overcomes the computational problem of the calculation of the residual sensitivity matrix by computing only the diagonal components in a series of back substitutions of the augmented equations derived from Hachtel's method (explained later). Broussoulle [32] has shown that the expensive computation of the covariance matrix can be avoided in all but the essential diagonal terms and that these can be computed quickly using a triangular decomposition technique. Broussoulle's method employs a technique known as the Z-sparse inverse. Interestingly, in a much later paper by Slutsker [221], it is shown that this technique can be extended to determine the off-diagonal elements of $\mathbf{P}$ as well. Slutsker and other authors [172, 173, 263, 264] have noted that once the $\mathbf{P}$ matrix is formed there is little need to re-calculate it based on bad data rejections as it is relatively insensitive to changes in the values of the states.

A slightly different test to the r^N test is introduced by Monticelli and Garcia [176] called the $\hat{b}$ test. It is explained as follows:

Let $\tilde{z}_i$ be the estimated value of some i^{th} variable considering all the existing

measurements except z_i. Then the estimate $\tilde{z}_i$ can be determined from:

$$\tilde{z}_i = z_i - \frac{\sigma_i^2}{p_{ii}^2} r_i \tag{6.8}$$

where σ_i is the presumed standard deviation, p_{ii}^2 is the i^{th} diagonal element of the covariance matrix $\mathbf{P}$ and r_i is the residual. Providing there is sufficient redundancy (or better still, that z_i is the only bad data point) then the *coherency* of the measurement z_i with the rest of the system is given by:

$$\hat{b}_i = \frac{z_i - \tilde{z}_i}{\sigma_i} = \frac{\sigma_i}{p_{ii}^2} r_i \tag{6.9}$$

Note that this does not obviate the need for the calculation of the normalized residuals but that it paves the way for bad data detection by direct measurement replacement, also known as IBE, as explained below.

The more recent *identification by elimination* (IBE) approaches [219, 220, 263, 264] have been developed out of the above techniques and are predominantly based on the r^N test. A typical IBE procedure is as follows:

1 Detect and identify the bad data using the r^N test;

2 Replace the bad data using the available measurement errors;

3 Re-estimate.

Note: If it were not for the fact that the relative size of the normalized residuals approximately correlate with the measurement errors in order of their relative size, the IBE technique would be a trial and error approach. In the case of single or multiple non-interacting bad data, the correlation is highly reliable [3].

One of the main goals of the HTI (and IBE) methods is to determine a maximum set of t true measurements and not use the remaining s selected measurements [264]. Of interest here is the nature of the residual sensitivity matrix $\mathbf{S}$. This is singular but with

a structure such that if the error locations are known in advance and the error vector can be partitioned thus

$$\mathbf{e} = [\mathbf{e}_s^T \quad \mathbf{e}_t^T]^T \tag{6.10}$$

where $\mathbf{e}$ is the measurement error vector, $\mathbf{e}_s$ are those values selected as errors and $\mathbf{e}_t$ are those values deemed to indicate correct measurements, then the sensitivity matrix can also be partitioned with the result that the residuals can be derived as

$$\mathbf{r} = \mathbf{S}_s\mathbf{e}_s + \mathbf{S}_t\mathbf{e}_t \tag{6.11}$$

whence, the estimate of the errors can be obtained from

$$\hat{\mathbf{e}}_s = \mathbf{S}_s^{-1}\mathbf{r}_s$$

thus the error estimates can be used to correct states since $\mathbf{S}_s$ is non-singular, although $\mathbf{S}$ *is* singular.

Hypothesis testing identification (HTI) [172, 173, 264, 270] is basically a four step method: detect; identify; remove; and re-estimate and therefore essentially includes IBE. The main difference between HTI and IBE is in the use of statistical tests used in each r^N identification step. In perhaps the definitive paper on HTI for power systems applications, Mili et al [172] extend the earlier ideas of Xiang et al [264] by proving several of the statistical relationships of the decomposed state estimator equations where the decomposition is carried out by considering the measurements as partitioned into a set of all good data and a set of all bad as in Eqn. 6.10. The authors also derive results for a mistaken partitioning in which some bad measurements are treated as good. Their method of HTI uses these statistical measures to guide the decision as to whether to accept the null hypothesis that measurement z_i is good or to reject it. Furthermore, two methods are presented based on each side of the false alarm paradox, ie that if false alarms are to be avoided then the probability of not selecting some bad data is increased or, conversely, if non-selection is important then the probability of false alarms is increased. The authors make the important step of recognising that it is more important to draw up an initial

list of candidate measurements that include all bad data and possibly some good data and then to trim the list iteratively according to the known statistics rather than the reverse procedure of trying to grow the list with newly identified bad data. The false alarm probability is therefore set high at the start of their proposed process and reduced iteratively.

Rather than deal with the statistics of the estimator equations and variables, it is possible to take a completely fresh view of the bad data identification problem based on failure probabilities. This method allows for maximum likelihood notions to aid in the determination of the list of suspect measurements and is called *combinatoric optimisation identification* (COI). Thus the list of suspect measurements is formed from the conventional least squares method followed by an r_N test but the list is processed by considering the probabilities and joint probabilities of failure of each particular instrument or pair of instruments [178, 221].

In 1971, Merril & Schweppe [170], introduced a method of state estimation that uses a variable *non-quadratic cost* (NQC) function which approximates to a WLS criteria when all the data is good. This particular bad data suppression method is often termed a 'direct approach' since it does not require pre-estimation processing or post-estimation detection and can therefore be considered a one step procedure as compared to the estimate, then post-process techniques of IBE, HTI and COI. The cost function can have one of two forms, viz:

$$C(\hat{x}) = \sum_{i=1}^{m} T_i \tag{6.12}$$

where

$$T_i = \begin{cases} \frac{r_i^2(\hat{x})}{\sigma_i^2} & \text{if } \left|\frac{r_i(\hat{x})}{a\sigma_i}\right| \leq 1 \\ a^2\left(4\left|\frac{r_i(\hat{x})}{a\sigma_i}\right|^{0.5} - 3\right) & \text{if } \left|\frac{r_i(\hat{x})}{a\sigma_i}\right| > 1 \end{cases} \tag{6.13}$$

The value a controls the detection threshold ($\geq 3\sigma$) and the residual is given by

$$\mathbf{r}(\hat{\mathbf{x}}) = \mathbf{z} - \mathbf{h}(\hat{\mathbf{x}}) \tag{6.14}$$

where the set of functions, $\mathbf{h}(\cdot)$, represent the nonlinear equations governing the behaviour of the electrical system. This version of NQC is also reported in [108]. A number of other non-quadratic cost functions are suggested by Handschin et al, [134], including quadratic-straight, multiple segment, quadratic-square root (as above) and quadratic-constant (see also [158, 159]). The authors of [134] make use of the normalized residuals rather than the weighted residuals (as above) and report on the trade-off between effectiveness and reliability of the various NQC options tested. One important but not surprising finding is that reliability deteriorates where there are abrupt changes in cost function slope. This deterioration usually arises out of numerical convergence difficulties, on the other hand, abrupt changes in slope can result in excellent bad data rejection. Other alternatives to the bad data suppression theme are given in [121, 270]. Falcao et al in [108] use a pre-estimation exponential filter combined with NQC to achieve slightly better discrimination against bad data. They also report on a subset of NQC, that of WLAV estimation.

6.2.1.2 WLAV Static State Estimation

An alternative formulation, known as the weighted least absolute values method (WLAV), has been proposed by a number of authors [12, 108, 119, 121, 142, 143, 152, 235] for water and power systems applications. The WLAV formulation:

$$\min_{\hat{\mathbf{x}}_k} |\mathbf{z}_k - \mathbf{g}(\hat{\mathbf{x}}_k)| \tag{6.15}$$

is more simple in form than the WLS cost function but tends to require a more sophisticated solution technique. There are three principal ways of deriving the WLAV function: Linear programming [12, 19, 142], non-linear programming [2, 20, 22] and approximate methods [50, 51, 144, 193, 223]. Dropping the k subscript, the cost function is normally written:

$$\min_{\hat{\mathbf{x}}} \mathbf{C} = \mathbf{w}^{\mathbf{T}} |\mathbf{z} - \mathbf{g}(\hat{\mathbf{x}})| \tag{6.16}$$

where $\mathbf{w}^T$ is an $m \times 1$ measurement weight vector. The linearisation step is performed as shown in Appendix 2. This gives:

$$\min_{\Delta\hat{\mathbf{x}}} \mathbf{C} = \mathbf{w}^{\mathbf{T}} |\Delta\mathbf{z} - \mathbf{J}\Delta\hat{\mathbf{x}}| \tag{6.17}$$

This formulation can be solved as a linear programming problem by introducing artificial variables r_i and s_i (where the i subscript refers to the i^{th} element or row):

$$s_i = \begin{cases} \Delta z_i - J_i\Delta\hat{x} & \text{if } \Delta z_i \geq J_i\Delta\hat{x} \\ 0 & \text{otherwise} \end{cases}$$

$$r_i = \begin{cases} -(\Delta z_i - J_i\Delta\hat{x}) & \text{if } \Delta z_i < J_i\Delta\hat{x} \\ 0 & \text{otherwise} \end{cases}$$

and the linear programming problem is now:

$$\min_{\mathbf{r},\mathbf{s},\Delta\hat{\mathbf{x}}} \mathbf{C} = \mathbf{W}^{\mathbf{T}} \begin{vmatrix} \Delta\hat{x} \\ r \\ s \end{vmatrix} \tag{6.18}$$

subject to

$$\Delta\mathbf{z} - \mathbf{J}\Delta\hat{\mathbf{x}} + \mathbf{r} - \mathbf{s} = \mathbf{0}$$

where $\mathbf{W}^T = [\mathbf{0}^T : \mathbf{w}^T : \mathbf{w}^T]^T$.

One reason why the WLAV estimate is conceptually appealing is because the solution is just the intersection of the n hyperplanes with the least absolute error values. This means that the WLAV estimator interpolates between the n data points whereas the WLS simply finds the best fit to all the data points. In the case where there is no solution point (ie the system of equations are inconsistent), the stopping criteria ensures that the hunt for a better basis does not carry on interminably. A disadvantage with the linear programming method is that the solution times tend to increase with n^2 whereas the WLS minimisation increases approximately as n. Experience with this method [193] tends to suggest that this is conservative.

Note the similarities between the WLAV estimator and the HTI and IBE methods of attempting to calculate state estimates based on the set of best measurements and using the partitioned residual sensitivity matrix to calculate the errors in the states with large normalized residuals. Note also that the LP WLAV technique just uses the weighted residuals in the cost function, a method which is widely known to be unreliable in the presence of multiple interacting bad data. Christensen and Soliman [50] also use this partitioning idea to produce a WLAV solution directly from a WLS method.

It is now shown that the WLS and the WLAV methods can be combined. Firstly, the WLS cost function is minimised with $\mathbf{W} = \mathbf{R}^{-1}$ i.e. assuming Eqns. 6.1 and 6.2 are correct and that any errors are uncorrelated. (Chapter 7 discusses correlated and uncorrelated errors).

The weighted residual

$$r_i = (\Delta z_i - J_i \Delta x)/\sigma_i \tag{6.19}$$

can now be used to determine the next weight for the corresponding measurement.

Note that if

$$w_{i,i}^{k+1} = \frac{w_{i,i}^{0}}{|r_i^k|} \quad \text{for} \quad i = 1, 2, \ldots, m \tag{6.20}$$

where $w_{i,i}^0$ is the initial weight for the i'th measurement, the cost function effectively becomes:

$$\min_{\Delta \mathbf{x}} \mathbf{C} = |\Delta \mathbf{z} - \mathbf{J}\Delta \mathbf{x}|^{\mathbf{T}} \mathbf{U} \tag{6.21}$$

where $u_i^2 = w_{i,i}$ which is just the error modulus function or WLAV. Thus it is possible to transfer from a quadratic WLS estimator to a linear WLAV estimator at any time and for any measurement. This result has already been published in [193–195, 223, 243].

In Irving's method [144] the 're-weighting' is carried out for all but very small residuals, since this avoids numerical problems when $r_i \approx 0$.

6.3 State Estimation in Water Networks

The estimator uses a model of the network which is based upon the non-linear flow equations discussed in Chapter 5. (In this work both the Hazen-Williams and the Colebrook-White representations have been have been used in tests on the state estimation algorithms, however, the present chapter will not concentrate on any differences brought about by the different representations.) The basic formulation is presented now for completeness. A fuller derivation is provided in Appendix 2.

Given a set of measurements $\mathbf{z}$, the model calculates an equivalent set $\hat{\mathbf{z}}$, where ideally, $\mathbf{z} - \hat{\mathbf{z}} = \epsilon$. Hence we have the measurement equation:

$$\mathbf{z} = \mathbf{g}(\mathbf{x}) + \epsilon \tag{6.22}$$

where $\mathbf{z}$ is the measurement vector (updated at every scan), $\mathbf{x}$ is the unknown state vector and $\mathbf{g}$ is the vector of non-linear functions.

The vector ϵ represents the measurement errors consisting of noise, gross bad data (due to meter failure etc.) and any discrepancy between the behaviour of the model and the physical system. It is more usual to assume that each $\epsilon_i \in N(0, \sigma^2)$, i.e. Eqns. 6.1 and 6.2 apply since when this is so, the WLS estimate is termed BLUE (best, linear, unbiassed estimate) [225].

The state vector $\mathbf{x}$ is classically estimated by minimising the weighted least squares (WLS) function:

$$\min_{\hat{\mathbf{x}}} \mathbf{C} = [\mathbf{z} - \mathbf{g}(\hat{\mathbf{x}})]^{\mathbf{T}} \mathbf{R}^{-1} [\mathbf{z} - \mathbf{g}(\hat{\mathbf{x}})] \tag{6.23}$$

where $\mathbf{R}$ is the diagonal measurement covariance matrix.

The network equations for power systems state estimation are not as non-linear as the equivalent water network equations, specifically, they do not have non-integer powers.

This is exploited by Allam [7] who uses the fact that in power systems state estimation the Hessian matrix of second order derivatives is constant.

The solution to Eqn. 6.23 can be obtained by non-linear optimisation techniques such as those in [2] but due mainly to the common problems with computational overhead in these approaches (often emanating from the line search steps), the Newton-Raphson procedure is adopted. As shown in Appendix 2, linearisation leads to the formulation:

$$\min_{\Delta\hat{\mathbf{x}}} \mathbf{C} = [\Delta\mathbf{z} - \mathbf{J}(\hat{\mathbf{x}})\Delta\hat{\mathbf{x}}]^{\mathbf{T}}\mathbf{R}^{-1}[\Delta\mathbf{z} - \mathbf{J}(\hat{\mathbf{x}})\Delta\hat{\mathbf{x}}] \tag{6.24}$$

for which the solution is obtained from the normal equation,

$$\mathbf{J}(\hat{\mathbf{x}})^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J}(\hat{\mathbf{x}})\Delta\hat{\mathbf{x}} = \mathbf{J}(\hat{\mathbf{x}})^{\mathbf{T}}\mathbf{R}^{-1}\Delta\mathbf{z} \tag{6.25}$$

where $\mathbf{J}(\hat{\mathbf{x}})$ is the Jacobian matrix of partial derivatives, $\Delta\hat{\mathbf{x}}$ is the difference between the linearisation point $\hat{\mathbf{x}}^{(k)}$ and the next state estimate $\hat{\mathbf{x}}^{(k+1)}$ and $\Delta\mathbf{z}$ is the difference between the actual and calculated measurements at iteration k. The state estimates are calculated starting from an initial estimate $\hat{\mathbf{x}}^{(0)}$. In the tracking mode, the previous estimate is used as the starting guess when available, thus saving computation time. Thus we have the projection of the vector $\Delta\mathbf{z}$ onto the column space of $\mathbf{J}(\hat{\mathbf{x}})$ via the projection matrix

$$(\mathbf{J}(\hat{\mathbf{x}})^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J}(\hat{\mathbf{x}}))^{-1}\mathbf{J}(\hat{\mathbf{x}})^{\mathbf{T}}\mathbf{R}^{-1}$$

The basic least squares algorithm is very simple:

1 Compute $\mathbf{J}$, using Taylor series linearisation at point $\hat{\mathbf{x}}_{\mathbf{k}}$;

2 Compute the calculated measurements, $\hat{\mathbf{z}}$, from $\mathbf{g}(\hat{\mathbf{x}}_{\mathbf{k}})$, and hence obtain $\Delta\mathbf{z}$;

3 Project $\Delta\mathbf{z}$ onto $\mathcal{R}(\mathbf{J})$ using Eqn. 6.25 and hence obtain $\Delta\hat{\mathbf{x}}$;

4 Determine new $\hat{\mathbf{x}}_{\mathbf{k}}$, where $k = k + 1$;

5 If $length(\Delta\hat{\mathbf{x}}) > \delta$, where δ is suitably chosen, GOTO 1

Eqn. 6.25 is in the form $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is a square and invertable matrix and so the solution is relatively straight forward. In the present research on water network state estimation, the choice of solution method has been influenced by the poor conditioning of the problem caused in part by the sparse nature of network measurements. The condition number (κ) is made worse if the $\mathbf{J^T R^{-1} J}$ term is actually calculated since it is known (see Stoer, [245]) that:

$$\kappa(\mathbf{J^T J}) = \Big(\kappa(\mathbf{J})\Big)^2 \tag{6.26}$$

However, a formulation known as Hachtel's method [101] avoids forming the product $\mathbf{J^T R^{-1} J}$ by use of the following equations:

$$\mathbf{r} = \mathbf{\Delta z} - \mathbf{J}\mathbf{\Delta \hat{x}} \tag{6.27}$$

$$\lambda = \mathbf{R^{-1}} \mathbf{r} \tag{6.28}$$

$$\mathbf{J^T} \lambda = \mathbf{0} \tag{6.29}$$

which can be formed into the augmented matrix:

$$\begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{J} \\ \mathbf{-I} & \mathbf{R^{-1}} & \mathbf{0} \\ \mathbf{J^T} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \lambda \\ \mathbf{r} \\ \mathbf{\Delta \hat{x}} \end{pmatrix} = \begin{pmatrix} \mathbf{\Delta z} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \tag{6.30}$$

which is still in the form $\mathbf{Ax} = \mathbf{b}$. Although there are now $2m + n$ equations to be solved, the sparsity is such that the number of elements stored in the new $\mathbf{A}$ matrix has only risen from $2s + m$ to $2s + 3m$ where s is the number of non-zero elements in the Jacobian whereas the condition number is reduced to approximately that of the Jacobian. This is a similar structure to that reported in [125, 126, 139, 157] and also used by Bargiela in [12].

The subjects of linear algebra and matrix mathematics provide us with a wide choice of solution techniques based either on LU (or LDU) decomposition or an expansion leading to the so called 'product form of the inverse'. Sparsity is an important issue here but there are also a great many methods for sparse systems [34, 100, 101, 131]. In this

work, solution to Eqn. 6.30 is obtained by LU factorisation and back-substitution using the Harwell subroutines LA05A and LA05B respectively [206].

6.3.1 Proposed Estimation Algorithm

The proposed design uses a new method to determine the weights *intra* iteration and is a departure from Irving's method. In Irving's method, the initial weights are the measurement variances and these are adjusted for all measurements with the exception of those measurements with very small residuals (e.g. $< 1 \times 10^{-4}$). Thus the method is a 'true' LAV criterion. Fig. 6.4 shows the effect of re-weighting on the cost function. In what follows, it is demonstrated that there is no need to make the WLS window so small, and that by re-weighting only measurements with large residuals the result is an efficient NQC procedure. The normal equations are:

$$\mathbf{J^T R^{-1} J \Delta \hat{x} = J^T R^{-1} \Delta z} \tag{6.31}$$

let

$$\begin{aligned} \mathbf{R^{-1}} &= \mathbf{W^T W} \\ &= \begin{pmatrix} \frac{1}{\sigma_1} & & & \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_m} \end{pmatrix} \begin{pmatrix} \frac{1}{\sigma_1} & & & \\ & \frac{1}{\sigma_2} & & \\ & & \ddots & \\ & & & \frac{1}{\sigma_m} \end{pmatrix} \end{aligned} \tag{6.32}$$

then

$$\mathbf{J^T W^T W J \Delta \hat{x} = J^T W^T W \Delta z} \tag{6.33}$$

let $\mathbf{H = WJ}$, so that $\mathbf{H^T = J^T W^T}$ and write $\mathbf{\Delta s = W \Delta z}$, then

$$\mathbf{H^T H \Delta x = H^T \Delta s} \tag{6.34}$$

The diagonal weight control matrix, $\mathbf{D}$, is now introduced so that

$$\mathbf{H^T D H \Delta \hat{x} = H^T D \Delta s} \tag{6.35}$$

where, initially, $\mathbf{D} = \mathbf{I}$. Hence in Hachtel's form the normal equation is:

$$\begin{pmatrix} \mathbf{0} & \mathbf{I} & \mathbf{H} \\ -\mathbf{I} & \mathbf{D} & \mathbf{0} \\ \mathbf{H^T} & \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{DWr} \\ \mathbf{Wr} \\ \mathbf{\Delta \hat{x}} \end{pmatrix} = \begin{pmatrix} \mathbf{\Delta s} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix} \tag{6.36}$$

Note that in this formulation, the residual covariance matrix is the same as the residual sensitivity matrix. Using the result from Appendix 2, the sensitivity matrix for the above system:

$$\mathbf{S'} = \mathbf{I} - \mathbf{H}(\mathbf{H^T H})^{-1}\mathbf{H^T} \tag{6.37}$$

but the covariance matrix,

$$\mathbf{P'} = \mathbf{S'WRW^T S'^T} = \mathbf{S'S'^T} = \mathbf{S'} \tag{6.38}$$

since $\mathbf{S'}$ is idempotent.

The problem now is to choose the values of d_{ii} so as to achieve the best possible bad data rejection. Unfortunately, as reported in [157], the condition number of Eqn. 6.36 is affected by $\mathbf{D}$ and that choosing a value of $d_{ii} \neq 1$ may produce an undesirable κ. This possible drawback has not caused any problems in the results obtained so far and therefore, provided the normal precautions such as using double precision arithmetic are taken and not allowing the weights to become disproportionately large or small, ill-conditioning problems do not justify not proceeding with the method. This effect, if serious, might be helped by scaling the whole system of equations but scaling does not solve the problem of round-off error [78].

With a poor starting guess, the first few iterations could produce errors which were not indicative of topological or metering anomalies, therefore in the tests conducted a small number of purely WLS iterations were permitted and then the current set of residuals were examined for the presence of bad data. The two principal methods of bad data detection are the largest normalized residual test and the $C(\hat{x})$ test. If the local

redundancy is sufficient in the area of the bad data then large residuals do indeed result, however, in the case of poor local redundancy the set of residuals can be larger in number but smaller in size. Therefore the safest way of initiating the re-weighting routines is when either a large residual is discovered or when the size of $C(\hat{x})$ is significant in terms of the probability of a false alarms. Given that bad data is detected, the re-weighting method is as follows:

$$d_{i,i}^{k+1} = \frac{d_{i,i}^{k}}{|r_i^k|} \quad \text{for} \quad i = 1, 2, \ldots m \tag{6.39}$$

This is carried out for all residuals provided $|r_i^k| \geq 3\sigma_i$, otherwise the measurement carries its full weight. Note that in this scheme, the weights get progressively smaller but to avoid worsening the condition number too much, they are prevented from becoming less than 1×10^{-5}. The choice of which measurement to re-weight is merely based on the hypothesis that the largest normalized residuals directly indicate the suspect measurements. If the absolute value of a re-weighted residual decreases from one iteration to the next it is assumed that the above hypothesis is false. There are two ways of dealing with this situation:

1) Increase the weight in proportion to the reduction in the residual, viz

$$d_{i,i}^{(k+1)} = \frac{d_{i,i}^{(k)} |r_i^{(k-1)}|}{|r_i^{(k)}|} \tag{6.40}$$

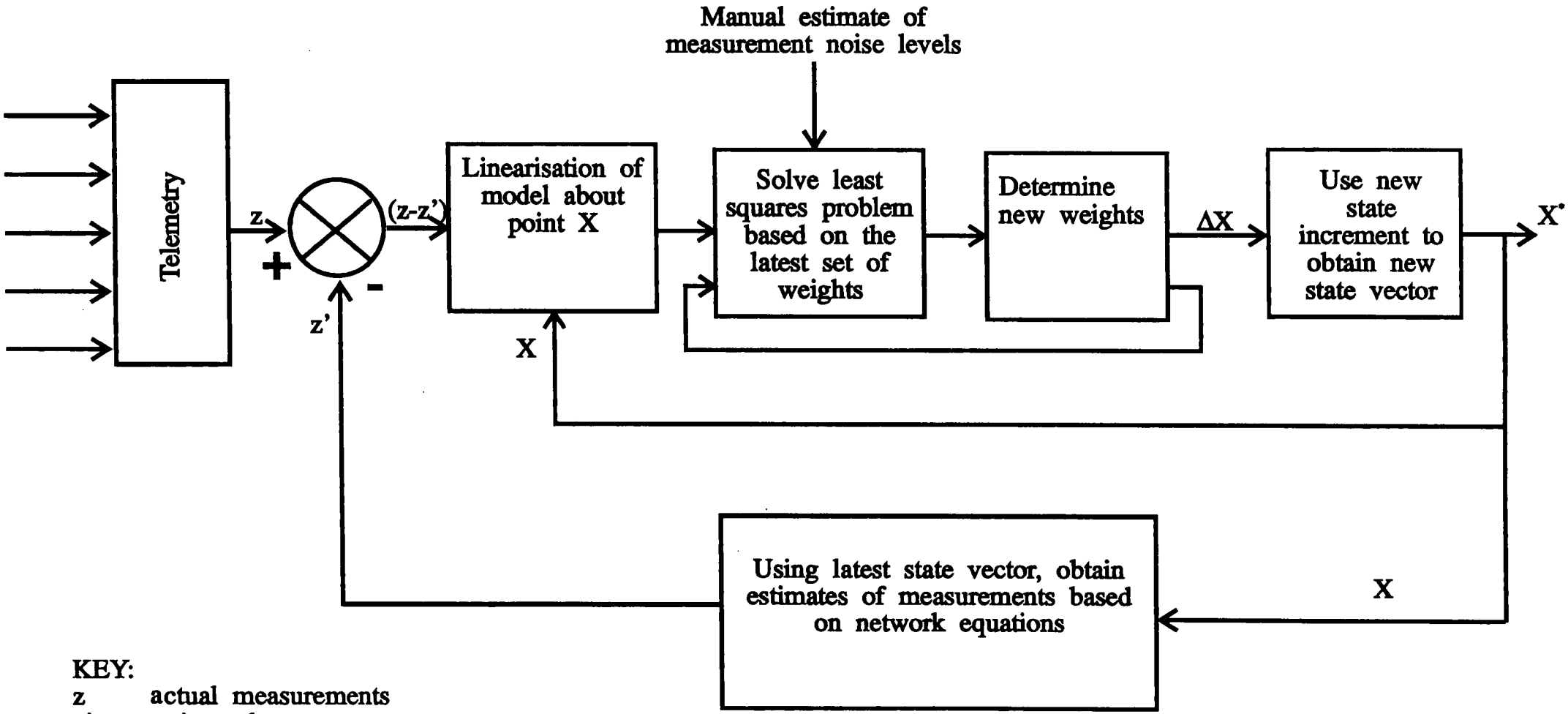
2) Return the weight to its full value provided

$$\frac{(|r_i^{(k-1)}| - |r_i^{(k)}|)}{|r_i^{(k-1)}|} \geq \beta \tag{6.41}$$

where β is some suitable value, say 0.05.

Both these schemes have been tried and they perform differently for different situations. Scheme (2) assumes that since the initial hypothesis (that the measurement

Block Diagram of State Estimator Loop



KEY:

z	actual measurements
z'	estimated measurements
X	state vector
X*	optimal state vector

Figure 6.2

Flow Chart for Re-Weighting State Estimator Main Loop

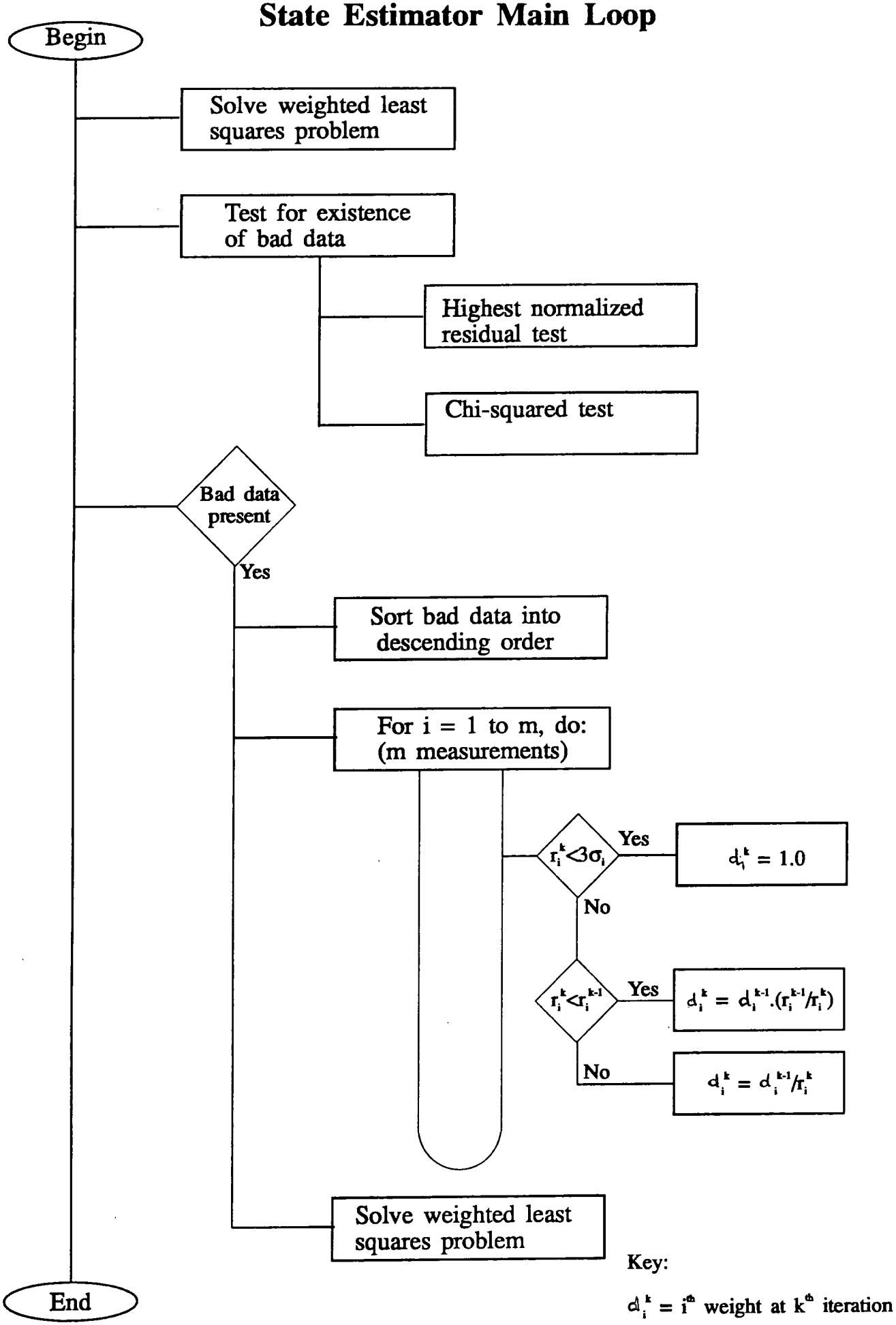
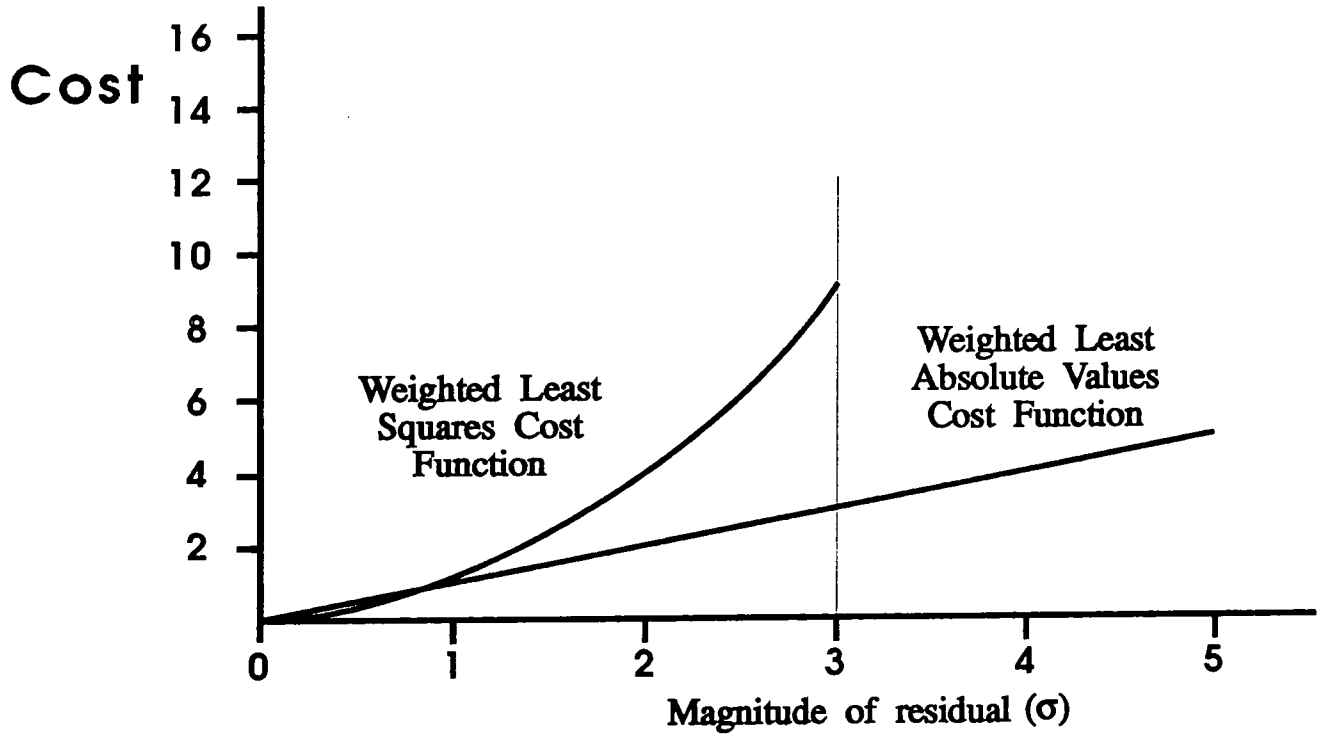


Figure 6.3

Variable Cost Function for Bad Data Identification



Variable Weight Function for Bad Data Identification

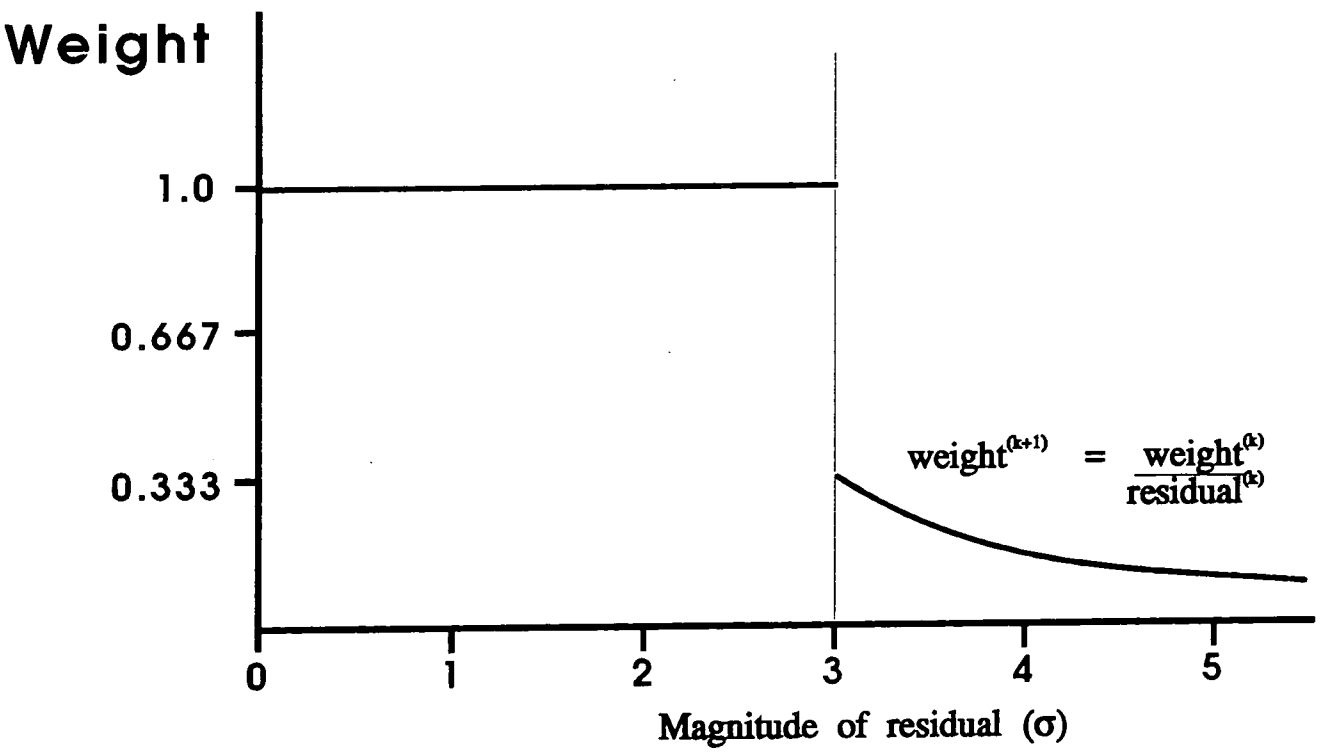


Figure 6.4

was bad) is probably false if the residual decreases from one iteration to the next, the full weight should be restored. Thus correcting a type 1 hypothesis error. This is not to say that the correct weight has yet been assigned, and since the process is *intra* iteration, the residual may not be monotonically increasing with each iteration following correct identification, therefore a decrease in its value should be accommodated without assuming that the hypothesis was false. This is the reason for β which should be adjusted so that bad data is not continually rejected and then accepted again. Scheme (1) makes a smooth adjustment to the measurement weights and is therefore preferable in the later iterations when the degree of rejection, which relates directly to the confidence that there is an error, is being determined.

Besides faulty measurements (and the high degree of uncertainty of the demand pseudo-measurements, in particular), the anticipated errors arising during state estimation of water networks will occur due to inaccuracies in the mathematical model resulting from unmodeled leaks, valve changes or other incorrect network parameters. However, by allowing those measurements with residuals of order 3σ to take on a weight appropriate to their residual size, in each iteration, the proposed method maps the uncertainty of the rejection into the degree of bias of the solution. Conversely, the greater the certainty of the rejection, the less the solution is biassed. The weights from the previous iteration are used as the initial weights for the next and so a high residual that is manifest from one iteration to the next causes the associated weight to be further reduced.

Note that this scheme begins by accepting all measurements and then recalculates the solution those previously identified as bad, suppressed. This is done without a new estimation step and since the Jacobian elements are unchanged, the system of equations to be solved do not require a full re-formulation. The discrimination method is somewhat analogous to the shrinking ellipsoid of [188] but the proposed method is much simpler.

6.3.2 Effect of re-weighting

Consider a residual at 3σ. Let the i^{th} measurement contribute to a measurement error such that:

$$\Delta z_i - J_i \Delta x = r_i = 3\sigma_i \tag{6.42}$$

or

$$\Delta s_i - H_i \Delta x = 3 = d_{ii} r_i = r_i' \tag{6.43}$$

Then the cost in the least squares cost function is

$$C_i^{(k)} = r_i \frac{1}{\sigma_i^2} r_i = 3\sigma_i \frac{1}{\sigma_i^2} 3\sigma_i = r_i' r_i' = 9 \tag{6.44}$$

However, after re-weighting

$$d_{ii}^{(k+1)} = \frac{d_{ii}^{(k)}}{3} \tag{6.45}$$

Hence the cost is reduced from 9 to 3 after the first re-weighting sequence for the i^{th} measurement. This step from one cost function to another aids discrimination. Similarly, a measurement at 4σ at the k^{th} iteration would represent a cost of 4 rather than 16 under the purely WLS scheme. A second re-weighting within the same Newton-Raphson iteration would produce a flat (constant) cost function. Thus the method presented allows for the construction of a family of NQC cost functions from a purely WLS algorithm with little extra programming, the simplest of these new functions being WLAV.

The conceptual view of re-weighting is the adjustment of the degree of influence of a particular measurement in accordance with its consistency in the calculation of the equivalent value from the state variables obtained in the previous iteration. By re-weighting, we are suppressing the effect of suspect Δz_i's on the solution vector $\Delta \hat{\mathbf{x}}$, and, subsequently, on the formation of $\mathcal{R}(\mathbf{J})$. Re-weighting is therefore only applicable as $\Delta \hat{\mathbf{x}}$ approaches the zero vector, since the knowledge that leads us to suspect the Δz_i's in the first place comes from $\mathbf{g}(\hat{\mathbf{x}})$ and we must not penalise the Δz_i's when the Jacobian terms are being formed at the wrong point in space. This latter point justifies the use

of the purely least squares scheme in early iterations when re-weighting might lead to incorrect measurement rejections and adversely affect the convergence.

The scheme proposed here is shown diagrammatically in the block diagram of Fig. 6.2 and the BS6224 flow chart of Fig. 6.3. The effect of weight adjustment is shown in Fig. 6.4.

6.3.3 Water Network Applications

In the control of water distribution networks, it would be desirable to know when and where transducers have failed or need calibration and furthermore it is essential that any change in the integrity of the network is identified rapidly. A burst pipe can lead to the ingress of detritus thus causing water quality problems, as well as loss of pressure and the waste of water.

Use of the state estimator proposed here would facilitate better on-line knowledge of adverse network conditions. The node numbers of the measurements which are identified as erroneous highlight the approximate location of the anomaly that caused them in the first place. If the local redundancy is sufficient, the location will be reliable (ie, the identification is correct), otherwise a group of residuals will occur in the vicinity of the anomaly and only some of those measurements will have been correctly identified. Furthermore, it could be the case that all measurements are faithfully reporting the situation but that the model is incorrect in some way. The method in Chapter 7 addresses these particular issues.

The development of the proposed estimator has been undertaken using full real-time simulations of power distribution and water distribution networks incorporating models for telemetry in both cases [225].

6.4 Test Results

The diagram of Fig 6.5 shows the network simulator and a telemetry program which have been designed to accept the on-line injection of faults which are due either to leaks, in the case of the simulator, or to transducer gross errors or noise, in the case of the telemetry program. The complete system is designed to simulate an actual SCADA system with measurements coming from a real water network. This then provides the necessary environment in which to design and test algorithms for actual network monitoring applications.

6.4.1 Noise Algorithm

The variance assumed in Eqn. 6.2 is that due to the expected noise level on each instrument signal. This value can be calculated by choosing a typical mean signal level of the instrument and calculating the likely total span of that instrument from the mean value. The accuracy can then represented by a percentage of the total span and this percentage say, $\pm 2\%$, would represent 3σ. The percentage chosen to represent the data point's variability under ideal operating conditions would of course depend on the type of the measurement. Pressure transducers are more accurate over a wider range of conditions than flow meters. Some flow meters are unidirectional, others operate in both directions, for these bi-directional meters an RMS value is obviously more appropriate than a value near zero. Demand measurements make up the bulk of the measurements available to the state estimator but are the most prone to error.

6.4.2 Bad Data Types

The first test is designed to show the bad data rejection capability of the proposed estimator. In this test, two non-adjacent measurement points, at nodes 1 & 15, (Fig. 6.6) were corrupted by adding a 25% error to each. The test results obtained for this test were

WATER SYSTEM SIMULATOR PROGRAM INTERACTION

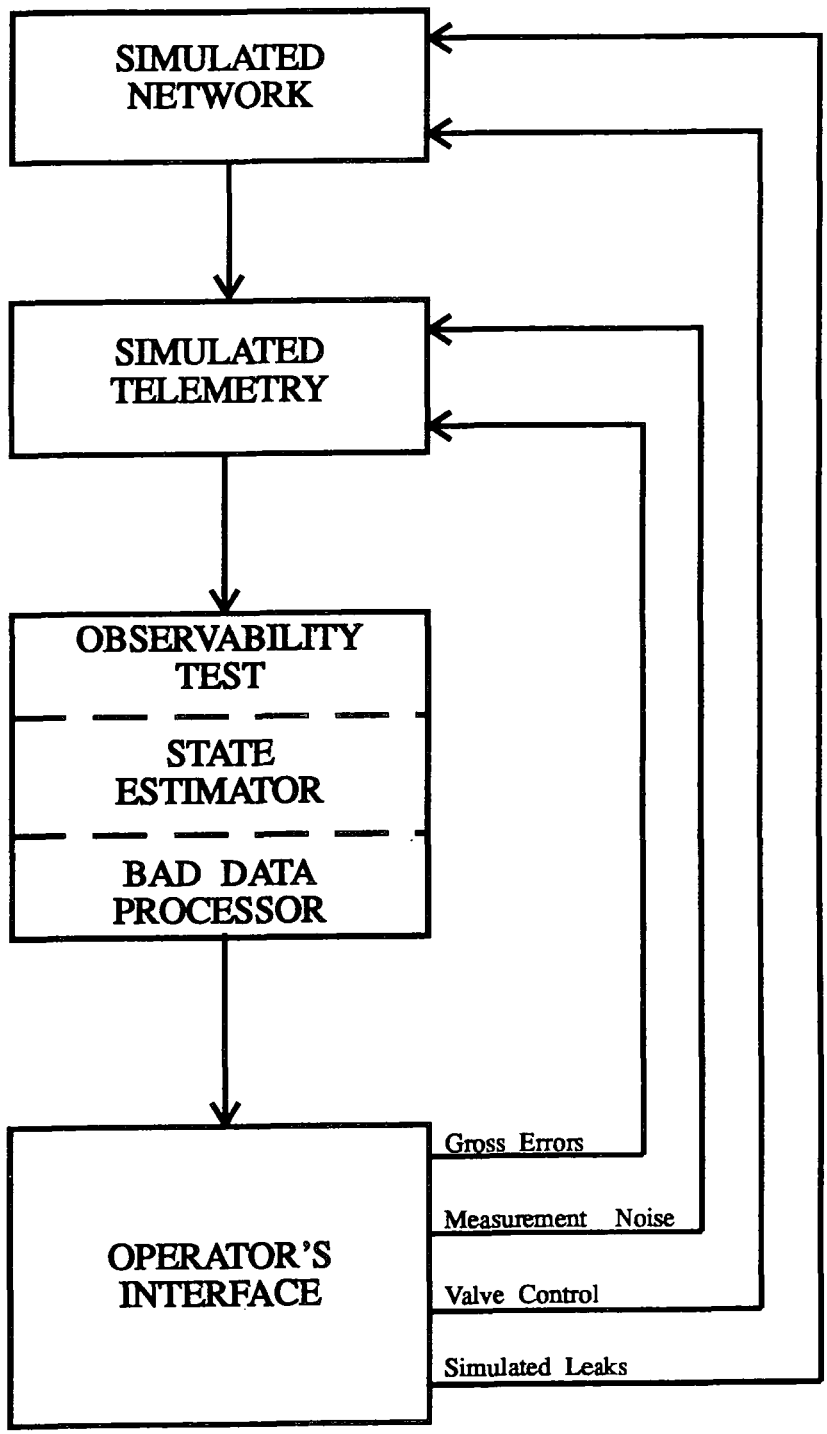


Figure. 6.5

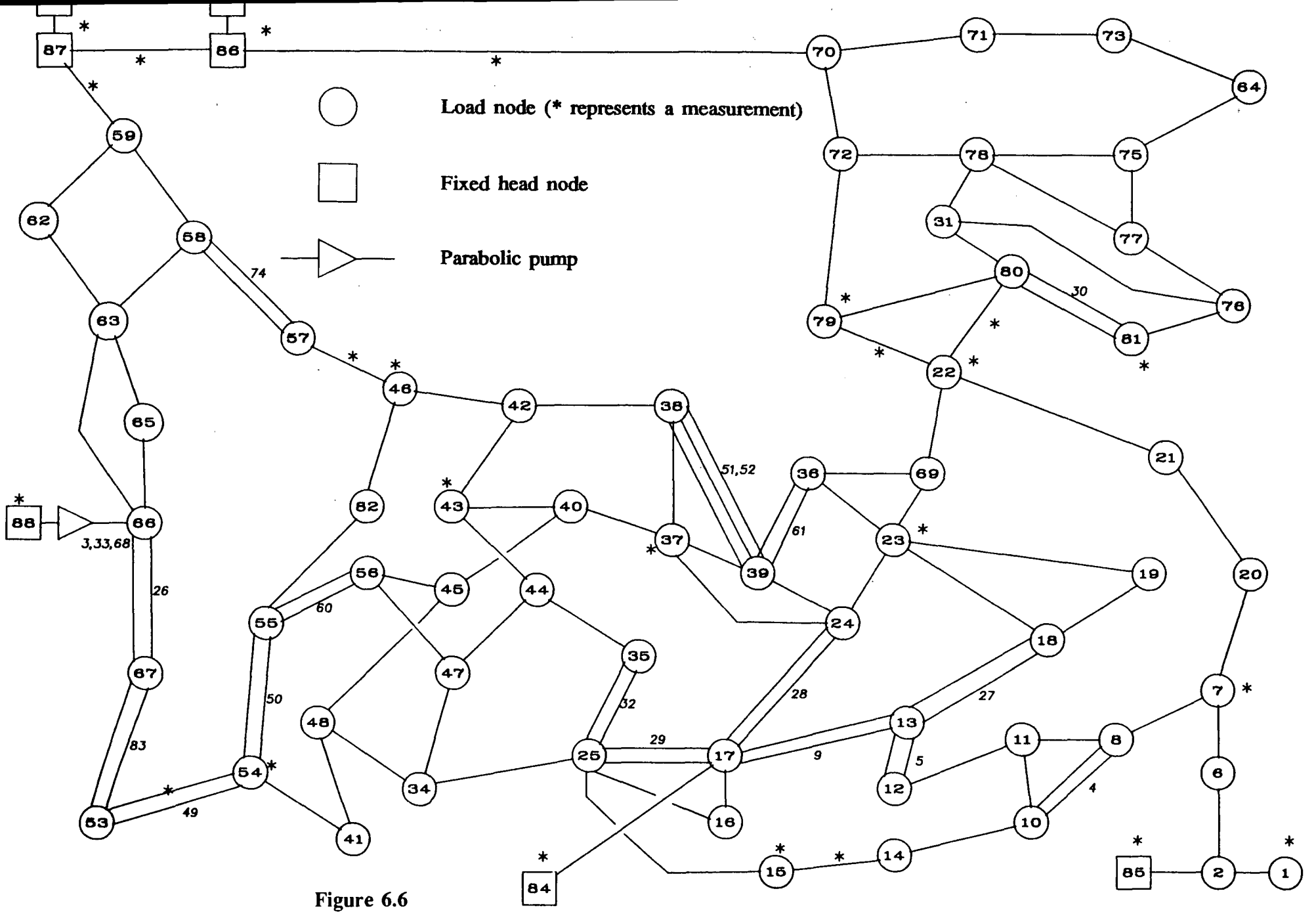

Load node (* represents a measurement)
Fixed head node
Parabolic pump
87
86
70
71
73
64
59
72
78
75
62
58
74
31
77
80
30
76
63
57
79
81
46
42
38
22
65
21
51,52
36
69
88
66
3,33,68
82
43
40
37
61
23
39
19
20
56
45
44
26
60
55
24
18
35
67
47
28
27
7
50
32
48
13
29
11
8
83
25
17
9
5
6
54
12
4
34
49
53
16
10
41
84
15
14
85
2
1

Figure 6.6

similar to others and so this is representative demonstration of the rejection properties of the estimator with multiple bad data present.

The results are shown in Table 6.1. The residuals are listed in the three right hand columns and show clearly that the bias present in the weighted least squares estimator (WLS) has been removed by re-weighting. Furthermore, one can see the equivalence with the results of the pure WLAV estimator which uses linear programming.

Errors from data corruption				
Node No.	WLS	LP-WLAV	RW-WLS$_w$	RW-WLS$_n$
1	22.022	33.575	33.578	33.577
7	-11.57	-0.002	0.000	0.000
15	23.772	33.977	33.980	33.979
22	-2.142	0.000	0.000	0.000
23	-2.058	0.000	0.000	0.000
37	-2.020	0.000	0.000	0.000
43	-1.976	0.000	0.000	0.000
46	-1.715	0.000	0.000	0.000
54	-2.054	0.000	0.000	0.000
79	-2.116	0.000	0.000	0.000
81	-1.954	0.000	0.000	0.000
84	-2.226	0.000	0.000	0.000
85	-11.474	0.002	0.000	0.000
86	-1.654	0.000	0.000	0.000
87	-1.709	0.000	0.000	0.000
88	-1.587	0.000	0.000	0.000

Table 6.1

Note, the values below 1×10^{-4} are rounded to 0.000 in the above table. It is a feature of the proposed algorithm that full weight is assigned to those measurements which do not produce a normalized residual over 3σ. This generally has the effect of ensuring very low residual errors for all data not carrying bad data.

6.4.3 Computational efficiency

In order to test the efficiency of three approaches to state estimation from a computational point of view a number of performance tests were undertaken using models of two different networks shown in Figs. 6.6 & 6.7, the parameters of which are given in [12 and 193].

The performance investigations were carried out on a Perkin-Elmer computer (type 3230) with 4Mbytes of RAM. This machine is rated at about 1 MIP.

6.4.3.1 Convergence

In each case the estimator was started using the same starting guess and the time and number of iterations to produce the correct solution were noted. The results are shown in tables 6.2:

Convergence				
Algorithm	34 Node Network		88 Node Network	
	Time (secs)	Iterations	Time (secs)	Iterations
WLS	6	4	34	7
LP (WLAV)	9	5	150	13
RW-WLS	6	4	34	7

Table 6.2

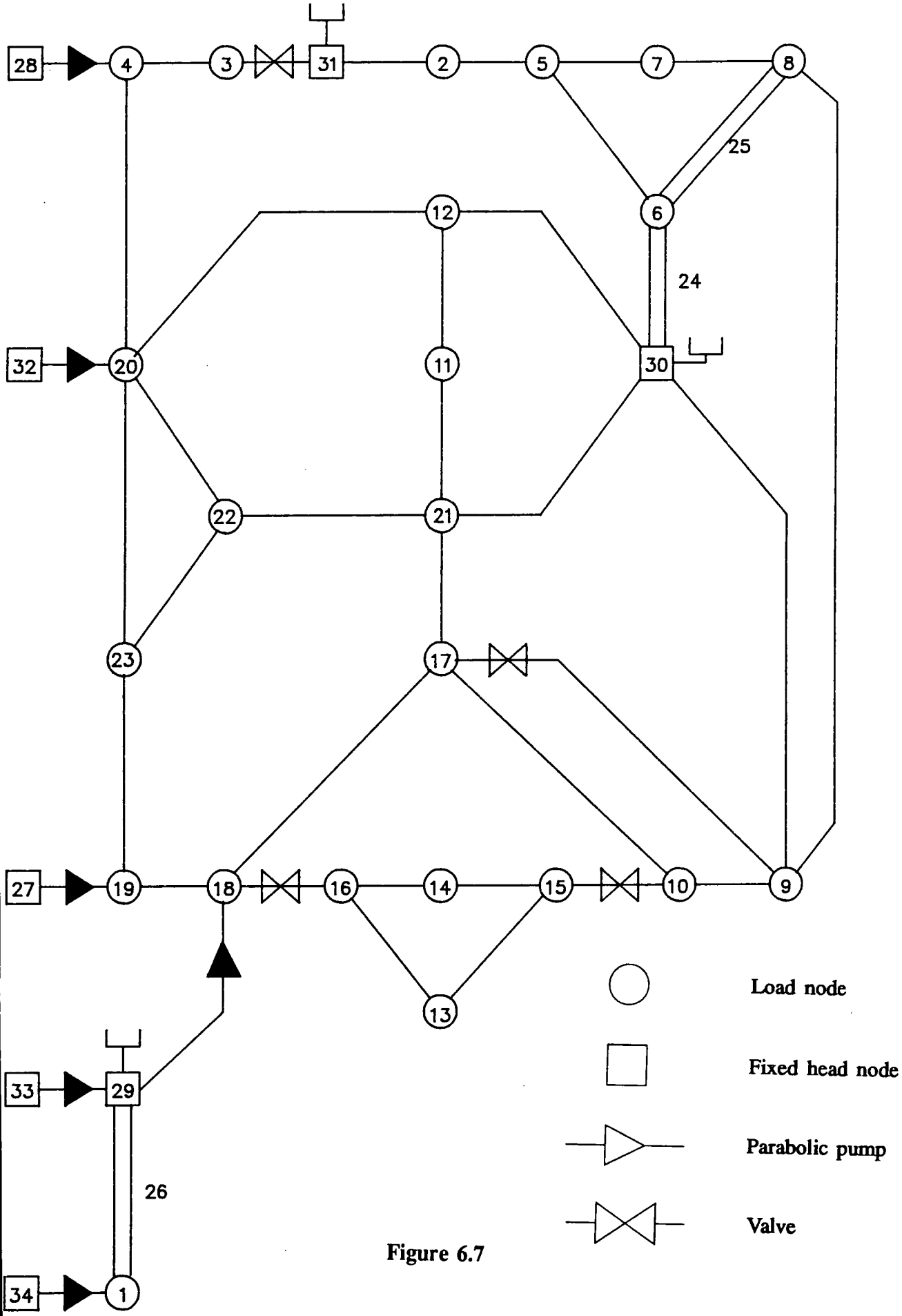

28
4
3
31
2
5
7
8
25
6
12
24
32
20
11
30
22
21
23
17
27
19
18
16
14
15
10
9
13
33
29
26
34
1
Load node
Fixed head node
Parabolic pump
Valve

Figure 6.7

Note: The 34 node network problem has 42 state variables and the 88 node network has 93.

It can be seen that the two estimators based on the least squares principal converge more quickly than the linear program, however, since no bad data was present in the above test the RW-WLS estimator is exactly the same as the WLS estimator. With bad data present the iteration times increase as shown below (the results were obtained on a VAX 3100 with 8Mbytes of RAM, the start-up statistic can be used to compare with the performance obtained in earlier results on the Perkin-Elmer computer):

Performance (seconds)						
	34 Node Network			88 Node Network		
Algorithm	Start-up	Tracking	Bad data proc.	Start-up	Tracking	Bad data proc.
WLS	3.5	0.6	2.8	8.56	1.0	8.44
RW-WLS$_w$	3.5	0.6	4.9	8.56	1.0	18.0
RW-WLS$_n$	3.5	0.6	8.7	8.56	1.0	34.5

Table 6.3

Notes:

1. The columns headed 'Bad data proc.' contains the times taken for the estimators used to obtain a state estimate with one head measurement corrupted by 25%.

2. The second row, 'RW-WLS$_w$', contains the results for the estimator using only weighted residuals, whereas the third row, 'RW-WLS$_n$', contains the results for the estimator using normalized residuals.

6.4.3.2 Noise Immunity

A further test of the robustness of the estimators was made by adding noise to the measurements. Each estimator was run in tracking mode and noise was added simultaneously to all measurements at a level calculated in the manner described above. The initial number of iterations and the subsequent number (in the next estimate) were recorded.

Robustness				
Algorithm	34 Node Network		88 Node Network	
	2% Noise	5% Noise	2% Noise	5% Noise
WLS	5, 1	7, 1	4, 1	4, 1
LP (WLAV)	6, 1	6, 4, 1	4, 1	N.C.*
RW-WLS	5, 1	8, 1	4, 1	5, 1

Table 6.4

* the linear program was unstable with 5% noise on the measurements (N.C. = not converged).

6.5 Conclusions

For water distribution systems, the reliability and availability of on-line measurements is considered more important than their absolute precision because the anticipated number of available measurements will be small due to cost considerations and hence any failures could lead to an unobservable system or sub-system. However, providing a state estimator can detect and identify gross measurement errors, within the constraints imposed by the measurement redundancy, reliable state estimates can be ensured.

In this chapter a new method of static state estimation has been proposed that is capable of accurately validating measurements of a distribution system and of simultane-

ously identifying and rejecting measurement anomalies. Such anomalies can indicate the change in state of the network topology and hence can give a measure of the appropriateness of the current model of the system.

The state estimator is robust, with high immunity to Gaussian noise and to the presence of gross measurement errors. In most cases, bad data do not effect the calculation of the final state vector. The proposed estimator combines the advantageous qualities of both the WLAV and WLS techniques. It reverts to a purely WLS estimator when there are no detectable bad data and only switches to rejection mode if bad data are detected with re-weighting carried out only for suspect measurements. As the solution converges, those erroneously selected bad data points are re-accepted as good data, thus maximising the total number of accepted good data points and improving noise immunity. The algorithm is simple to implement and therefore straight forward to maintain. It is thus suitable for on-line water network state estimation applications.

The proposed method has been successfully applied to both power system state estimation and water system state estimation, using the appropriate simulated environments with facilities for measurement corruption by noise and gross errors [225].

CHAPTER 7

PROCESSING MEASUREMENT ANOMALIES

7.1 Introduction

Reliable on-line control necessitates the use of validated on-line data from the system. Data validation was the subject of the previous chapter which developed some methods for the reliable identification of errors in the measurement set. However, merely detecting and identifying 'erroneous' measurements is insufficient to guarantee reliable control. The reason for this lies in the fact that although bad data may have been identified, the underlying cause of the bad data has not been understood. In fact, no account has been taken of the possibility that the instrument concerned may have been reporting the situation faithfully and that some other assumption upon which the current model is based has become invalid. In this chapter the possible causes of bad data are analysed so that a diagnosis of the cause can be postulated. A simple algorithm that is capable selecting groups of bad data related topologically is also presented.

This chapter comprises 7 sections: Section 7.2 reviews work related to the theme of the chapter; Section 7.3 contains a discussion on the causes of bad data in a water network state estimator; Section 7.4 presents an analysis of the interconnection between residuals, which leads to a bad data grouping algorithm presented in Section 7.5; a further

method for processing bad data based on failure probabilities is outlined in Section 7.6 and conclusions are drawn in Section 7.7.

7.2 Review of Previous Work

Algorithms referred to as 'bad data processing' algorithms in the literature are actually concerned with the correct identification of anomalous measurements and not concerned with the further analysis of the underlying cause of the bad data, one possible exception to this is [178]. In this work the process of state estimation contains within it a procedure that identifies, as far as possible, bad data points and therefore the term 'bad data processing' is reserved for the subsequent processing of the bad data in order to discover an underlying cause. Somewhat surprisingly, the literature is not replete with algorithms in this area.

The power systems literature on the subject tends to be divided between conventional single or multiple meter failure identification or the identification of topological changes which are not echoed in the model. Since the bad data detection/identification problem has already been covered, this survey will concentrate mostly on the topological anomaly case.

Bargeila introduced the idea of bad data processing in water distribution systems state estimation [12], in order to resolve between instrumentation faults and topological anomalies such as the change of state of switching valves or leaks. His method employs an analysis of large residuals observed in the demand measurements at each end of a pipe and uses a simple analysis of the size and sense of the residual pairs to discriminate between closed and open valves and leaking pipes. Figures 7.1, 7.2 & 7.3 are a pictorial representation of Bargiela's method but although these ideas are useful, they depend for their success upon the assumption that the measurements at the end of each pipe are

sufficiently redundant that the errors there are identifiable and unfortunately this cannot be relied upon.

A method by Clements et al [54], provides the bounds on topology error detection and identification for single branches. Their argument is a dual of the measurement error detection argument and considers the degree of detectability of topological errors in branches of an electrical network.

Let $\mathbf{f(x)}$ represent the flows in branches and let $\mathbf{\Delta f} = \alpha\mathbf{f(x)}$ represent an error in the branch flows due to a topology error, then the relationship between the residual and the branch flow error is:

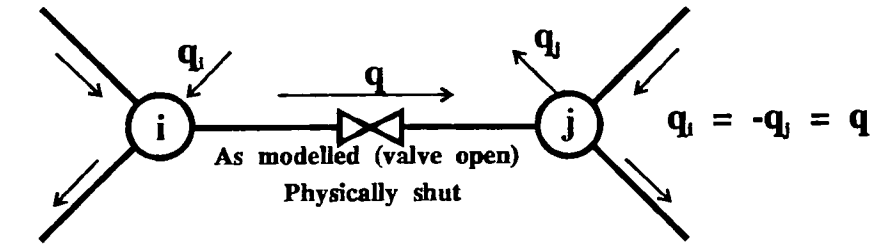
$$\tilde{\mathbf{z}} = \mathbf{z} - \hat{\mathbf{z}} = \mathbf{SM\Delta f(x)} \tag{7.1}$$

where $\mathbf{S}$ is the residual sensitivity matrix and $\mathbf{M}$ is the measurement to branch incidence matrix. By examining the rows of $\mathbf{SM}$ for linear dependency, the authors show that branches can be categorised into non error detectable; detectable but not identifiable; and both detectable and identifiable.

In their paper published in 1980, Lugtu et al [160], present a method not dissimilar to Bargiela's in that it uses the injection measurements at the end of each line to determine whether any of the measurements are suppressed (they use a WLS estimator with bad data suppression). From this observation a new line status is postulated and the bus injections are recalculated as a test. Their method assumes that buses at the ends of a line have redundant injection measurements, just as in Bargiela's method, where it was assumed that demand measurements were available. The method therefore suffers from exactly the same drawback as Bargiela's method.

Wu et. al. [262] go a step further than Lugtu et al and derive a mathematical relationship that gives the size and direction of the residuals from structural changes in the Jacobian. Their argument is as follows:

TOPOLOGICAL ERROR (1)

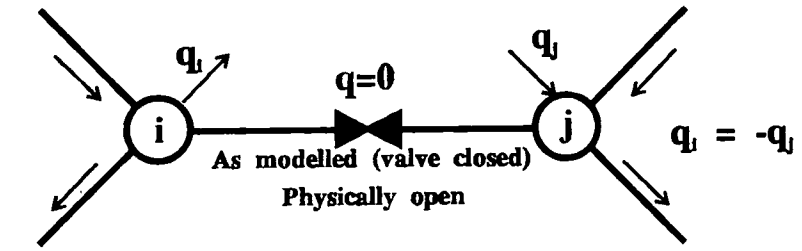


Synopsis

Node	Bad data type	Value
i	Demand	<0
j	Demand	>0

Fig. 7.1

TOPOLOGICAL ERROR (2)

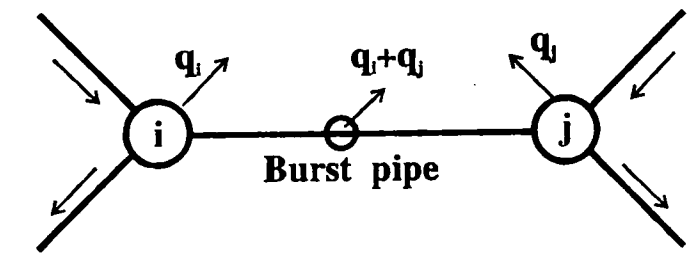


Synopsis

Node	Bad data type	Value
i	Demand	>0
j	Demand	<0

Fig. 7.2

TOPOLOGICAL ERROR (3)



Synopsis

Node	Bad data type	Value
i	Demand	>0
j	Demand	>0

Fig. 7.3

From Appendix 2 of this thesis, the residual covariance matrix is

$$\mathbf{P} = (\mathbf{I} - \mathbf{M})\mathbf{R} \tag{7.2}$$

where $\mathbf{M} = \mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}$. Following a topological change, the Jacobian matrix will cease to represent the true behaviour of the system. Let the unknown, true Jacobian matrix be $\mathbf{J_t}$ and let $\mathbf{B}$ be defined so that

$$\mathbf{J_t} = \mathbf{J} + \mathbf{B} \tag{7.3}$$

then the residual vector becomes

$$\mathbf{r} = (\mathbf{I} - \mathbf{M})(\mathbf{B}\Delta\mathbf{x} + \epsilon) \tag{7.4}$$

ie when a topology error is present, the residual becomes $(\mathbf{I} - \mathbf{M})\mathbf{B}\Delta\mathbf{x}$ instead of zero. It is the structure of $\mathbf{B}$ that is important and the authors show that only rows corresponding to the measurements adjacent to the line outage contain non-zero elements and that these rows have the following structure:

$$[-b_i \qquad b_j]$$

where b_i and b_j have equal magnitude and are simply the value of the line sensitivity.

The above analysis is in line with the arguments of Bargiela and Lugtu et al. and demonstrates that individual and, in particular, solitary residuals convey information about the *size* of the measurement errors whereas groups of residuals convey information about the *type* of problem. These observations motivate the proposed bad data analysis method given in Section 7.3.

KLos and Dobrzański present a method of topology identification in [149] which assumes that part of the topology is already known but some breaker status information is missing. In other words, there is a known number of unknown branches. Their approach uses a tree-cotree incidence matrix for the whole network and then processes

columns of the matrix containing the unknowns using a generalised inverse to determine a solution (not necessarily unique). The method is designed to be used with unvalidated measurements (assumed valid) but could be used in conjunction with the proposed state estimator. However, topology errors may not be the cause of the large residuals and so this method could only be used after topology errors have been detected.

The theory of diagnosis has been increasingly applied using expert systems to a variety of problems which involve some form of monitoring. The most applicable of such methods appears to be that of alarm processing which is found in both the power systems area and more generally in process control. To the author's knowledge no such diagnostic scheme has been used directly with state estimation. Singh and Glavitsch [218] present a scheme which is based on an expert's analysis of a power system diagram with an hitherto undetected breaker status error. Their method uses a topological search and they also propose an inference engine to report on the success of the search (although no clear results are given in their paper).

The method of combinatoric optimisation identification (COI) due to Monticelli et al [178] was mentioned in the last chapter but it is worth discussing again here because it is the only bad data processing algorithm that determines the transducer faults on a basis of the direct likelihood of transducer failure. The method leaves the framework of state estimation and residual analysis and instead examines the transducers themselves and their individual behaviour. This provides additional information which can aid in the determination of failures from a group of alternatives. The authors propose that the list of residuals be augmented by a list of failure probabilities and that the optimal decision is the one which maximises the likelihood function which is constructed from the product of the failure probabilities. A branch and bound algorithm is described for carrying out the optimisation. Although a highly useful method, it is based on the non-discriminating way in which the WLS estimator works. Note, for instance, the number of high residuals

in the left most column of Table 6.1 of the last chapter. In the proposed estimator, the number of residuals is reduced due to the discrimination that has already taken place.

7.3 Causes of Bad Data

In order to carry out an analysis of the causes of bad data, it is necessary first to consider the physical system and the measurements collected from it. The measurement equation was defined earlier as:

$$\mathbf{z} = \mathbf{g}(\mathbf{x}) + \epsilon \tag{7.5}$$

where the error term is often assumed to be a random variable with first and second moments given by:

$$\mathbf{E}(\epsilon) = \mathbf{0} \tag{7.6}$$

$$\mathbf{E}(\epsilon.\epsilon^{\mathbf{T}}) = \mathbf{R} \tag{7.7}$$

where $\mathbf{R}$ is the measurement covariance matrix which is assumed diagonal. This assumption means that the errors coming from any two transducers do not covary. Unfortunately, even this basic assumption becomes false under certain situations where, for instance, the topology of the model is incorrect.

The errors in the Eqn. 7.5 can arise from a number of situations:

Type 1) Measurement Noise. This is assumed uncorrelated between measurements and stationary. Measurement noise is certain to exist to some degree for all on-line measurements since water flow itself tends to be turbulent added to which, signals from transducers undergo A/D conversion and long distance transmission. In general, for this type of contamination (only), the errors have an $N(0, \sigma_i)$ or Gaussian distribution and hence, in this case alone, Eqns. 7.6 & 7.7 hold. Measurements that are only contaminated by noise contain useful information.

Type 2) Meter Semi-failure. For example, poor calibration or instrument drift. The distribution of errors is likely to be approximately Gaussian but will be flatter than that for measurement noise. Errors are still uncorrelated and some useful data may be provided depending on severity.

Type 3) Meter Total Failure. Here the transducer or transmission link has failed altogether and the incoming signal is either –ve or +ve full scale deflection or there is a zero reading or some other spurious reading. These errors have a totally flat distribution across the range of possible values and consequently no useful information is given. The errors are uncorrelated.

Type 4) Model Failure: Parametric. Errors included in this category are not due to instrumentation faults but arise from modelling errors resulting in the wrong parameters being used to represent diameters, lengths, pipe roughness etc. or are due to a poor choice of flow model. The resulting errors, identified as measurement errors, occur in groups and those in the same group are correlated. The actual distribution is not guaranteed to be Gaussian.

Type 5) Model Failure: Topological. In this case the transducer is again faithfully reporting its measurement but the local topology has changed and thus is different to that modelled. This situation will arise when a valve is closed or opened and the model not updated or if a pipe bursts. The errors are grouped and those in the same group are correlated.

It is thus the case that actual measurement errors are uncorrelated whereas model based errors whether topological or otherwise give rise to correlated changes in groups of incoming signals. As discussed in the previous chapter, however, single and multiple instrument faults can bias the state estimates and produce a smearing effect such that the identified errors are correlated even though the actual meter faults are not. Therefore

simply looking for the presence of correlated bad data is not a sufficient test for model based faults.

It is therefore necessary to consider a diagnosis of the cause of the bad data from its type (ie. flow, demand or pressure), magnitude and direction ($+ve$ or $-ve$). From our knowledge of how the estimator works, we can build up templates for the recognition of bad data groups and try to eliminate the bad data by cause assignment and re-modelling.

In the literature on the subject, e.g. [11], bad data groups are defined as sets of measurement points for which the presence of bad data is detectable but not identifiable. That is, although the group can be identified, the measurement point itself (within the group) cannot. (Note that this is not the definition that applied to the word 'group' in the preceding paragraphs.) Bad data groups exist because of a lack of individual measurement redundancy, in fact the available redundancy can be thought of as shared by members of the group. The physical manifestation of the group is that a single cause can result in a set of errors which appear unrelated because of the way they are scattered across the network.

In using the state estimator outlined in this thesis, the only information about any of the errors listed above come from the residuals via the mapping (see Appendix 2 for derivation) :

$$\mathbf{r} = \mathbf{S}\epsilon \tag{7.8}$$

Since $\mathbf{S}$ is a projection matrix, it is singular. Furthermore, we do not compute any of the off-diagonal terms of $\mathbf{S}$ in carrying out the state estimation (the diagonal terms are computed and used to derive the normalized residuals) and therefore even if it were convenient to solve Eqn. 7.8, using an appropriate technique such as singular value decomposition, it would still be necessary to fully build the matrix in the first place.

In this work, an alternative, heuristic technique has been developed based on a simple search method which obviates the need for the full calculation of S.

7.4 Analysis Based on the Shortest Path Between Residuals

In order to gain a greater understanding of the effects of leaks on the state estimates, a number of trials were made on two test networks. In each test a (solitary) leak was introduced into a pipe in the network. The test was repeated for every pipe but in order to evaluate the sensitivity of the estimator, the size of the leak was determined as a percentage of the flow in that link. In most cases, depending on the soil outside the pipe, the magnitude of individual leaks in water distribution systems are primarily dependent on the pressure in the pipe. In the simulator, the leak is modelled as an additional demand lying midway between the two end nodes of the pipe. This additional demand is not modelled as a pressure dependent variable and thus can be set to any desired value.

The results of these tests showed that, in the cases where the redundancy was insufficient to detect an error at the adjacent measurement point, groups of bad data were clustered in such a manner that they surrounded the leaking pipe, sometimes up to 4 nodes distant. This spread of residuals can be explained by the bad data group concept referred to above. The method given below exploits the clustering of the group.

7.5 Algorithm for Processing Residuals

Residuals caused by network anomalies effect different types of measurements. Pressures in real distribution networks (as opposed to the simulated case) will decrease in an area near to a large leak. Therefore, one should be able to observe residuals representing a decrease on pressure measurements due to leakage. In the same way, the pressure upstream of a blocked pipe will be high, whereas, down stream, it will be lower. Demand measurements will be affected as shown in Figs. 7.1 – 7.3 but Bargeila's method

treats demand measurements immediately adjacent to an individual pipe, here we extend the reasoning to cover larger areas of the network.

The proposed method is as follows:

1) Sort normalized residual list into descending order;
2) Determine paths linking groups of high measurement residuals;
3) Any measurement error not linked is identified as a meter failure;
4) Select all errors in demand measurements;
    4.1) Search for leak type patterns;
    4.2) Search for blockage (closed valve) type patterns;
5) Test the diagnosis to determine its veracity.

Step 1), above, has been covered earlier, the other steps are explained below starting with the method to determine groups of localised residuals.

7.5.1 Shortest Path Algorithm

A simple algorithm based on network topology was developed for the purposes of finding the least set of links between any group of residuals. Starting with the arc-node incidence matrix for the full network, the method exploits the feature of this matrix that when one column (which represents a node) is added to another (representing an adjacent node), the result is a representation of a new network wherein the two nodes have become one and the link between them has been deleted. Thus, working from two starting nodes it is possible to 'collapse' the arc-node incidence matrix until the two nodes in question have become adjacent. A record is kept of the combinations and using this record, a backtracking procedure determines the path connection. A simple example illustrates the procedure.

Referring to the simple network in Fig. 7.4, it is desired to find the shortest route (least number of links) between nodes 5 and 11. The arc-node incidence matrix for the full network is $\mathbf{A}^0$:

$$\mathbf{A}^0 = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

adding all those columns which share non-zero row entries with column 5 to column 5 and likewise for column 11, gives:

$$\mathbf{A}^1 = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ & & & & & & \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ & & & & & & \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ & & & & & & \end{pmatrix}$$

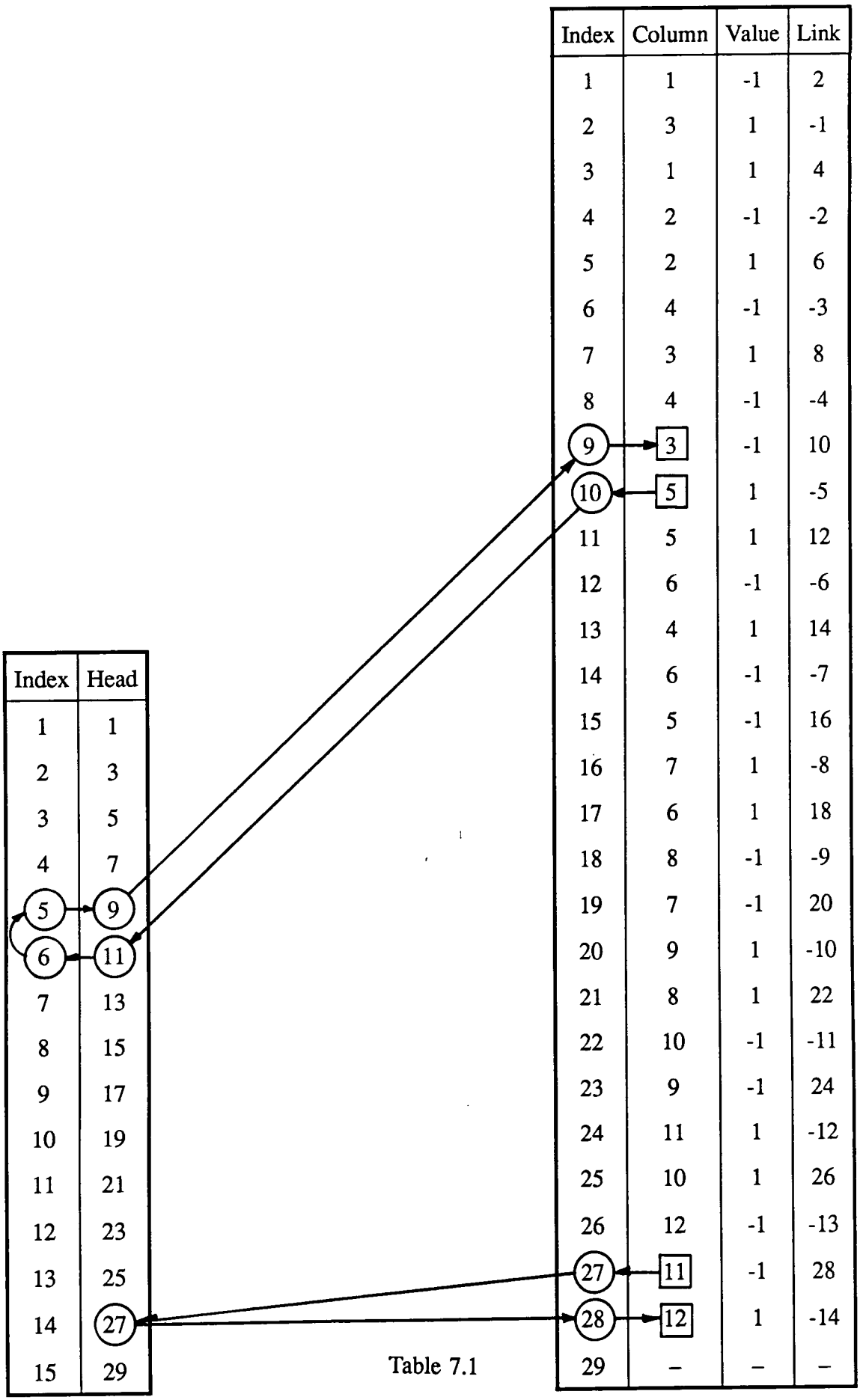
the rows and columns with no non-zero entries have been left blank to show the cancellations clearly, written more compactly, we have a new matrix corresponding to a new

network with the links adjacent to the two starting nodes deleted:

$$A^1 = \begin{pmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

in this case, the addition of columns can be halted since nodes 5 and 11 are now adjacent. Were this not to be the case the algorithm would continue until the two nodes in question were adjacent.

As with all matrix operations in this work, the scheme as implemented exploits the sparsity inherent in the arc-node incidence matrix. Due the nature of the operations on the columns, zero elements may be created where non-zero elements where previously stored and so a static storage scheme cannot be used. Instead the method employs a dynamic sparsity storage structure as shown in Table 7.1.



Index	Column	Value	Link
1	1	-1	2
2	3	1	-1
3	1	1	4
4	2	-1	-2
5	2	1	6
6	4	-1	-3
7	3	1	8
8	4	-1	-4
9	3	-1	10
10	5	1	-5
11	5	1	12
12	6	-1	-6
13	4	1	14
14	6	-1	-7
15	5	-1	16
16	7	1	-8
17	6	1	18
18	8	-1	-9
19	7	-1	20
20	9	1	-10
21	8	1	22
22	10	-1	-11
23	9	-1	24
24	11	1	-12
25	10	1	26
26	12	-1	-13
27	11	-1	28
28	12	1	-14
29	–	–	–

Index	Head
1	1
2	3
3	5
4	7
5	9
6	11
7	13
8	15
9	17
10	19
11	21
12	23
13	25
14	27
15	29

Table 7.1

Strictly speaking, the ordering of the rows of the 'head' array is arbitrary in a dynamic scheme provided the ordering of the other three arrays correspond, since the 'link' array always points to the next entry. In the ordering given here, the static storage scheme can be seen as well. It is just the above without the 'link' array. However, ordering is not arbitrary in the static storage scheme.

A draw-back with the above proposed method stems from the complexity of programming. Since, in the column combination process, zeros appear where non-zeros existed before, the sparse matrix storage vectors have to grow to accommodate the new terms and leave deleted elements unreferrenced, despite $dim(A)$ actually shrinking. This is untidy and costly of computing power. A conceptually less elegant alternative is to store the original arc-node incidence matrix in a static storage scheme and to compute 'adjacency vectors' which list the nodes connected to the i^{th} node from each row processed. A further advantage with this method over the column combination approach is that since the columns are not actually being combined, there is no need to store the values (1 and -1) thus saving computer operations and space. The arrows drawn on Table 7.1 refer to the algorithm given below.

```
Procedure Build_Vectors
        set I = 0
        outer loop:  new adjacency vector
            increment I by 1, set J = 1
            ADJ_VEC(I,J) = current node number
            inner loop:  fill current adjacency vector
                increment J by 1
                scan COLUMN for next instance of
                        current node number
                set IX = index of that entry
                scan HEAD for first number ≥ IX
```

```
                (Let this number be P)
        IF P = IX THEN
            set COL_NUM = (P+1)
        ELSE
            set COL_NUM = HEAD((index of P)-1))
        ENDIF
        ADV_VEC(I,J)=COLUMN(COL_NUM)
        repeat until no more occurrences of
                that node number in COLUMN
repeat for all node numbers
End
```

The above algorithm simply converts the static sparse form of the arc-node adjacency matrix into a set of row vectors each of which starts with the primary node number followed by those nodes directly adjacent to it. The next two procedures process this matrix to determine the paths with the least number of links. In fact, the adjacency matrix is further processed to remove the redundancy of node numbering sequences thus avoiding finding paths which simply loop back to the starting node.

Assume that the two nodes between which a path is to be determined are p & q respectively. The algorithm then compares successive elements of the p^{th} adjacency vector against the q^{th}. If no matching nodes are found, the vector corresponding to the node which is second in the p^{th} vector is processed against the q^{th} vector in the same way. Then the third, and so on until all the nodes in that vector have been used. Next, the same is done with the nodes in the q^{th} row. This sequence is repeated, row by row until a path is found. When a path has been found, the algorithm continues until it has found all paths with that number of links which connect the two nodes, this is because parallel paths often occur in heavily meshed networks. While this is happening, the current nodes

are held in another vector one by one as they are being processed. When a successful path is found, it is this latter vector that contains its description.

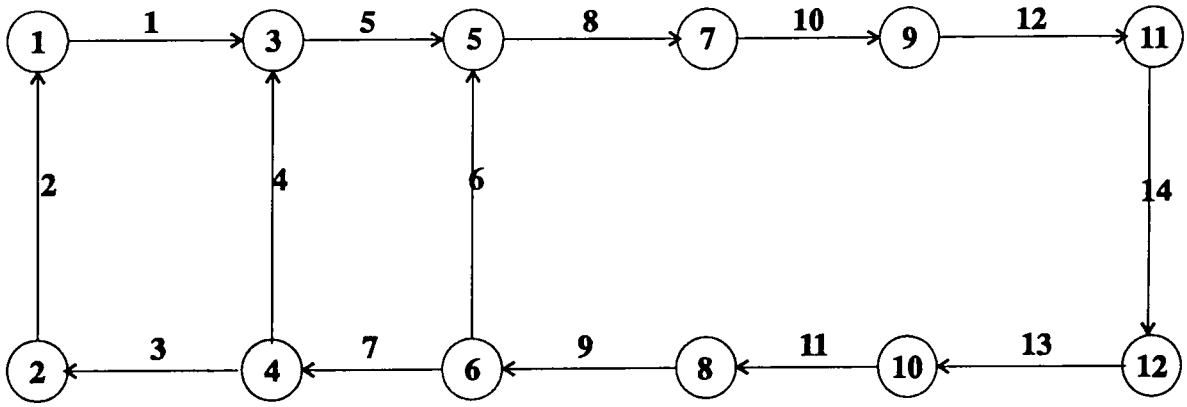
This algorithm costitutes a breadth first search starting from both the beginning node and the end node and spreading outwards until a connection is detected. The reason for staring from both nodes is to minimise the number of nodes visited in the search. If another pair of residuals is present the same search is initiated. This basic search method can be used for any pair of measurements of any type when it is necessary to determine the path with the least number of links between them. In the form described above, the algorithm will stop as soon as one path is found, however this is not desirable and so the actual stopping criteria is modified to allow the latest vector to be fully processed even if a path has already been found. The result is a set of parallel paths between any two measurements. Residuals separated by more than 5 or 6 links do not, as a rule, belong to the same group with the exception of boundary condition residuals (such as network in-flow measurements which are, for instance, adjusted negatively whenever positive adjustments are made to demands in an area local to a leaking pipe). Therefore the algorithm is artificially stopped if more than 5 links have been spanned. This aids in discriminating between residual groups arising from different events.

7.5.2 Processing Residual Groups

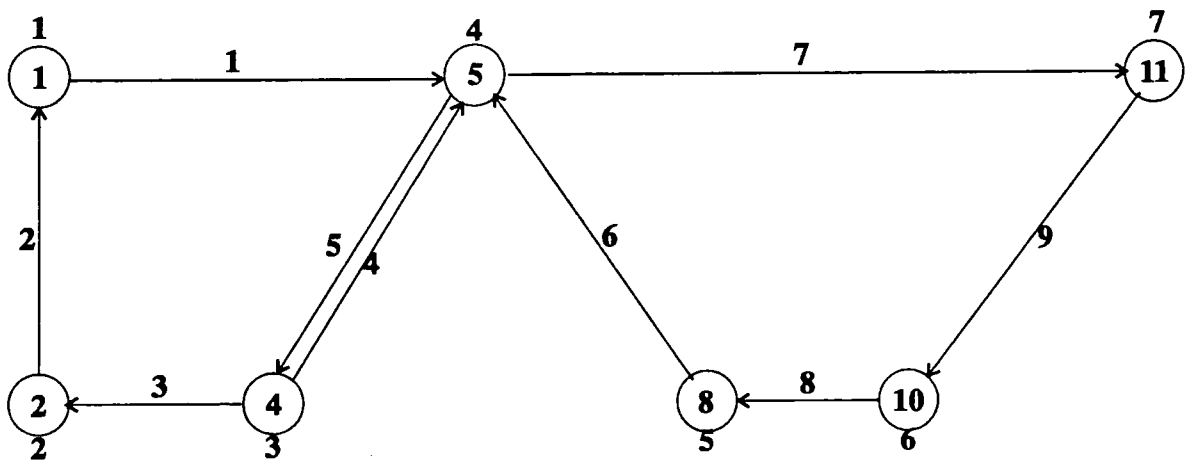
Once the connecting paths between sets of high residuals have been identified, heuristics can be applied to determine cooperating groups of residuals and, importantly, to discriminate between different sets of residuals arising from different causes.

1. *Common paths.* It is postulated that if the same paths are identified (or parts of the same paths) by the search between any three residuals, then that shared path is more likely to contain a topological anomaly by virtue of the fact that it is common to both sets of residuals.

Diagram showing topology processing step



Network corresponding to A^0



Network corresponding to A^1

Key:

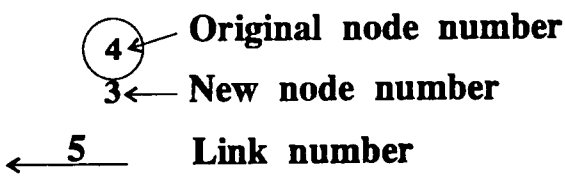


Figure 7.4

2. *Residual size.* If the normalized residuals are of the same size, according to [11] they could belong to a bad data group which suggests that the anomaly is a measurement anomaly that cannot be attributed to either of the two measurements with a probability greater than 0.5. However, if there exists a third (smaller) residual in the vicinity then it is more probable that the third residual is nearer to the actually aberrant residual than the other, erroneously rejected residual. If no such smaller residuals exist then the two equally large residuals could either be in a bad data group or, it could be the case that there is a common cause (they are correlated) and so it is postulated that the problem lies in a link equidistant between them.

The main difference and advantage with this method over those in the published literature is that the requirement for anomaly identification is only that there is global redundancy in the measurements rather than there be redundant demand measurements at each end of the link containing the actual fault.

7.5.3 Results

The algorithms outlined above were implemented as FORTRAN subroutines and a set of tests were conducted to measure the leak detection capabilities. The network used for this test is 88 node network, shown in Fig. 6.6. By systematically working through the network as modeled by the simulator, leaks were introduced, one at a time, in every single pipe. The detections are listed in the two tables below. Table 7.2 show the 100% successful detections. Table 7.3 shows the detections that gave an adjacent pipe rather than the correct one.

The parameters of the test were as follows:

Head measurements: 7, 15, 22, 46, 54, 79, 81 84, 85, 86, 87;

Flow measurements: 56–46, 80–22, 69–22, 22–79, 53–54, 15–14, 70–86, 59–87;

Fixed head in-flow measurements at nodes: 84, 85, 86, 87, 88;

Demand & pseudomeasurements: All nodes.

Total number of measurements: 112

Dimension of state vector: 93

Measurement Redundancy:

$$\rho = \frac{m}{n} = \frac{112}{93} = 1.2 \tag{7.9}$$

Successful leakage location		
Node nos.	Node nos.	Node nos.
1 — 2	30 — 81	74 — 57
6 — 7	31 — 80	57 — 46
84 — 17	70 — 71	53 — 67
49 — 53	70 — 86	53 — 83
22 — 79	86 — 87	83 — 67
80 — 81	87 — 59	3 — 83
81 — 76	58 — 57	14 — 15
80 — 30	58 — 74	

Table 7.2

Leakage detection in neighbouring pipe			
Actual location	Estimated location	Actual location	Estimated location
Node nos.	Node nos.	Node nos.	Node nos.
2 — 85	1 — 2	25 — 32	15 — 25
2 — 6	1 — 2	41 — 54	54 — 53
8 — 10	14 — 10	54 — 53	54 — 83
10 — 14	14 — 15	54 — 49	54 — 50
15 — 25	15 — 14	54 — 55	54 — 53
16 — 17	17 — 84	79 — 80	22 — 79
17 — 13	17 — 84	80 — 22	80 — 79
17 — 9	17 — 84	72 — 79	22 — 79
21 — 22	22 — 79	71 — 73	70 — 71
22 — 69	22 — 79	46 — 82	46 — 57
24 — 17	84 — 17	62 — 59	87 — 59
28 — 17	84 — 17	7 — 8	6 — 7
17 — 25	84 — 17	54 — 50	54 — 53
17 — 29	84 — 17	46 — 42	46 — 57
25 — 35	15 — 25		

Table 7.3

7.6 Failure Probability Method

In an analysis of the causes of bad data, several choices result. Set out below is a scheme which provides heuristics to rank the possible causes of the bad data in terms of likelihood. The following clauses have been written in a similar format to that used by PROLOG, the $\wedge$ symbol is the AND function, the :: symbol is the conclusion of the clause:

significant time since last relining ∧ material susceptible to corrosion ∧ corrosive water :: poor pipe condition

high operating pressure ∧ poor pipe condition :: increased likelihood of leakage

high operating pressure ∧ large number of appurtenances :: increased likelihood of leakage

estimator reports demand type errors indicating high localised demand :: increased likelihood of leakage

The combination of these clauses would provide insightful information as to the whereabouts of bursts or meter failures in the system that could not be inferred from meter failure probabilities as suggested by Monticelli et al [178]. The relational database can be used to store the required information and it is envisaged that in the course of operational events, the data held against the individual network components could be updated. However, this would require an increase in the level of operator interaction which might not be acceptable on top of the interaction needed to keep the other database information up to date.

If this system were to be implemented, the actual probabilities would be needed and this would require an input from the water company using the system. For that reason the method was not taken further in this work but it is hoped to develop the scheme further in future collaborative research.

7.7 Conclusions

This chapter has presented an analysis of possible causes for high residuals in the state estimator equations. It has also presented a simple algorithm for bad data

Water Network Fault Identification Scheme

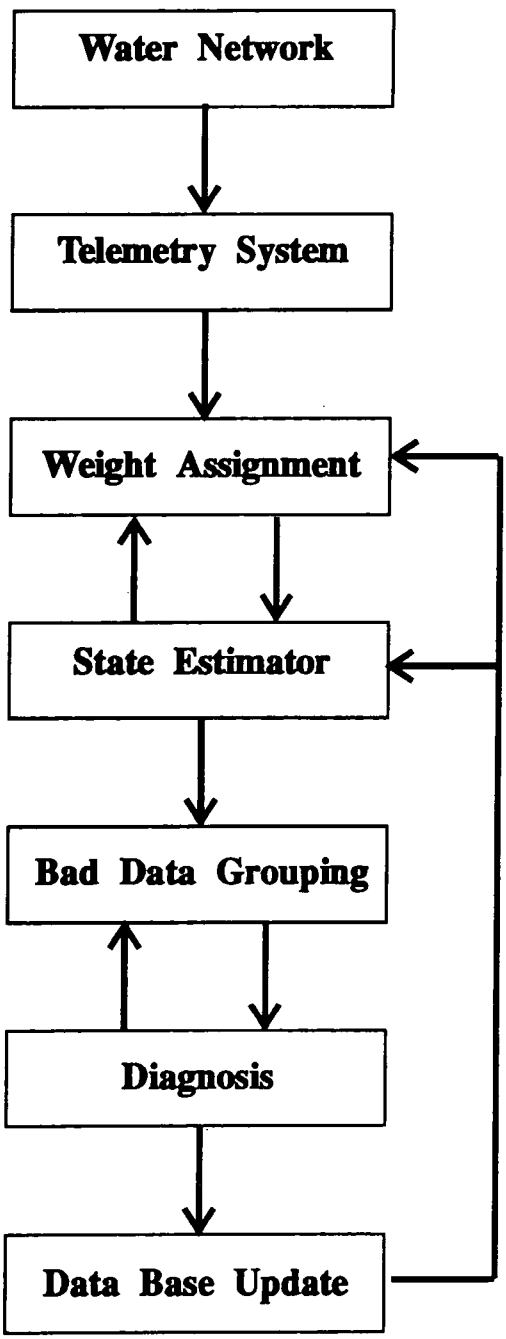


Figure 7.5

grouping based on network search methods. Topological anomalies can be found by the methods already known but such methods rely on redundant measurements adjacent to the topological anomaly.

Provided that more than one residual is present, the proposed algorithm sorts bad data by type, magnitude, direction and location. It employs a shortest path analysis to effect the grouping necessary to attempt to connect between members of the same bad data groups. The presence of multiple bad data is assumed to imply that a fault other than a measurement error is present (e.g. a burst or a blocked pipe) if that hypothesis fails because a link between the residuals cannot be found, then the residuals are assumed to be caused by multiple transducer faults. The overall structure of this system, together with the weight assignment ideas of Chapter 6, are shown in Fig 7.5.

For future work, it may be interesting to consider a geographically local area (using the database) to discover 'closed pipe' cases where, in fact, the pipe has been left out of the model. A topological search would not find the proximity of the two nodes whereas a search taking into account of geographical proximity might.

CHAPTER 8

CONCLUSIONS & SUMMARY

8.1 Introduction

The science of state estimation for power distribution networks is very well developed and there have been a very large number of publications on the subject over the last two decades. However, the majority of these state estimation methods remain unused and there are only a small number of commercial state estimators on the market. In the case of the water industry, the situation is rather more depressing. To this author's knowledge, there are no state estimators currently in use or on the market for the on-line monitoring of water distribution systems. The reasons for this situation fall into two broad categories: the first is that the science of state estimation is couched in terms of abstract mathematical models and simplifying assumptions; secondly, the water industry has not had the computer resources on hand nor the measurements available for such a system. There has also been little perceived need for on-line state estimation until very recently.

The central theme of this work has been to bridge the gap between the science of state estimation and the physical system and in so doing, a number of problems have been uncovered and solutions found, these include: the need for an easy to maintain model of the system that makes sense to the operator; the consequent need to determine

the mathematical abstraction of what the operator sees without operator intervention; the requirement to carry out combinations of certain network elements when constructing the mathematical model; the requirements of an on-line state estimator for the water industry in terms of robustness and reliability and lastly the need to make some sense of the results of the estimator.

There are many reasons why a state estimator is relevant to the water industry, which range from the detection of bursts to the control of pressure. The state estimator provides reliable network state information to which an optimal pressure profile can be compared. If the pump scheduling suite is in use, the determination of an energy cost optimised solution to the scheduling problem based on an accurate demand forecast and the network hydraulic constraints will be carried out daily. However, during each operational day, the actual demand will inevitably deviate from the forecast. Trying to control leakage using the scheduler would be nearly impossible and so the best form of control is on-line pressure control to ensure that the supply security is maintained at minimum cost. The prerequisite to on-line pressure control is the availability of on-line state information.

8.2 Mathematical Model Maintenance

In the design of an on-line monitoring scheme, it is desirable that the description held in the computer is adaptable if it is to have anything other than a very short useful life. In power system state estimation, the switch states are usually measured and hence are assumed known to a reasonable degree of certainty, whereas in water networks, the great number of switch valves (generally at least two for every pipe in the network) are not monitored nor is it likely that they will ever be. The vast number of these valves coupled with both the uncertainty as regards their status and the likelihood of this status changing quite frequently, albeit for a small number only, makes the task of maintaining

a network model of sufficient accuracy for use in state estimation both highly important and very difficult.

Adapting a model continually to a changing plant is not a new problem in control theory but in such approaches the *parameters* of the model are tuned (or identified) rather than the *structure* and this makes the problem tackled here quite different. Secondly, the problem is not limited to the determination of topology, since as switch valve states are changed then the location of demand points also change with respect to the network elements in the model. Furthermore, rebuilding an old model is considered important for training or investigating a serious event etc. Since the full plant description is held in the database, an advantage with the approach is that previously held descriptions can be created via the use of triggers, as described in Chapter 3. Lastly, and most crucial of all, is that the description of the system held in the database should be directly mapped into the mathematical model used in simulation, state estimation or any other analysis function applicable to the system.

This thesis has provided methods which solve all the fundamental problems outlined in the preceding paragraph associated with automating the interaction between the database description of the plant and the mathematical model. The techniques rely on human interaction since the telemetry in U.K. water supply systems does not provide information about switch valve status for all valves in distribution networks. The methods presented could be applied to all utility distribution systems directly without fundamental change.

The on-line database assumes the role of data manager, providing a common repository for all data from asset data to dynamic data and from which the network topology can be computed directly. From the network topology and the network data one can build the network model and hence can thus provide the facility to integrate any number of analysis tools.

8.3 Description of Pipe Flow

The thesis has reviewed some of the flow formulæ commonly used in models of water networks and presented new techniques which can be used in conjunction with the relational database system. In particular, the applicability of different formulæ has been examined from the point of view of accuracy over a wide range of conditions (Reynold's numbers) with emphasis on the problems arising from automatic pipe combinations in rudimentary network simplification. A number of new formulations have been tried and insights have been gained on the relative merits of different formulations.

It has been shown that the well known Colebrook-White formulation lends itself to accurate network simplification procedures involving the combination of either serial or parallel pipes rather better than the Hazen-Williams formula and, as expected, the errors in flow or pressure calculation for an equivalent pipe using the Colebrook-White formula do not increase as much with deviations in Reynold's Number from the design case as they do if the equivalent pipe were calculated and modelled using the Hazen-Williams approach. This fits in with the database approach in that the detailed descriptions of pipe parameters can be held in the database but more simplified descriptions can be used in the model. The operator has complete control over this simply through the designations given to links in the `Links` relation.

The work in this section of the thesis was aimed at ensuring that the mathematical model was accurate in the parametric sense as opposed to the structural sense of the previous section. Clearly both are essential for the reliable application of state estimation techniques.

8.4 The Application of State Estimation Methods

For water distribution systems, the reliability (ie the existence) of on-line measurements is considered more important than their precision for a number of reasons. That the precision of transducers is not important, can be understood by remembering that, as was shown, the state estimator presented is not effected by noisy measurements and is therefore tolerant to small scale meter inaccuracy. The *existence* of suitable measurements is important because the anticipated number of available measurements will be small due to cost considerations and hence any failures could lead to an unobservable system or sub-system. The objective of an on-line monitoring scheme is to accurately observe the system under a wide range of conditions including meter failure. Providing a state estimator can detect and identify gross measurement errors, within the constraints imposed by the measurement redundancy, reliable state estimates can be ensured.

In this work a new method of static state estimation has been proposed which is capable of providing accurate measurements of a distribution system and of simultaneously identifying and rejecting measurement anomalies. Such anomalies can indicate the change in state of the network topology and hence can give a measure of the appropriateness of the current model of the system.

This thesis has presented solutions to those problems associated with the provision of on-line monitoring capabilities for water distribution networks. It is a move towards the realistic application of state estimation and in so doing it has provided new light on the techniques involved. Many of the techniques and results presented have a wider application than the one focussed upon here.

8.5 Suggestions for Further Work

Relational database technology has been shown to be appropriate for storing network data although the major drawback is that of performance. Relational databases were developed primarily for corporate systems and not for real-time control applications. Further work in this area will be fruitful now that the functionality of the problems and their solutions has been completely identified. The method suggested to other researchers in this field is to explore more efficient database technologies such as object oriented databases. The main advantage is that this type of database is RAM based rather than disk based and should therefore provide the required real-time performance. However, as has been clearly demonstrated here, the design of schemas to efficiently store data in a consistent way and the design of retrieval methods providing a reliable linkage to a mathematical model is far from trivial.

As computer performance continually increases, the possibility of performing more calculations in real-time also increase. In on-line monitoring applications the size of the network and its structure are both important. It will be necessary, in the future, to design reliable decomposed state estimators which can solve very large systems which comprise smaller sub-system with few interconnections. Residual processing is very important and, as has been suggested in Chapter 7, there are many unexplored avenues in this area. A singular value decomposition of the residual sensitivity matrix would provide a method for determining the meaning of groups of residuals. However, by the time residuals are available, there is not much more information to be gleaned from the results and so it is necessary to bring in additional knowledge to augment the analysis, such as failure history, failure probability etc.

8.6 Summary

This thesis has presented a set of solutions to the problems of obtaining, in real-time and over extended periods, an optimal observation of a generalised water network. It has covered the storage of asset data which is the 'raw' form of network data not normally directly used in network analysis. In Chapters 4 & 5, a number of analyses have been presented to show that it is possible to map from the network data model produced by the database routines to a mathematical model of the network wherein parallel and serial pipes are combined. Furthermore, a new method of state estimation has been proposed which is capable of efficiently identifying bad data points and produces unbiassed estimates even when there are a number of measurements that are corrupted with gross errors. Lastly, schemes for determining the underlying cause of any errors identified by the estimator have been presented in order to provide a complete picture of the network under observation.

The work has suceeded in making an important step towards the realisation of the composite control loop for optimal on-line monitoring and control as discussed in Chapter 2 and shown in Fig. 2.3. It is now expected that the component parts of this loop will each be validated and the loop configured in the near future and that optimal, real-time operational control of water distribution systems will become possible.

References

1. AAM, S., HOLTEN, L., GJERDE, O. "Design of the measurement system for state estimation in the Norwegian high-voltage transmission network" *IEEE Trans PAS*, Vol. PAS-102, No. 12, pp 3769–3777, Dec. 1983.

2. ABBASY, N.H., SHAHIDEHPOUR, S.M. "Application of non-linear programming in power system state estimation", *Electrical Power Systems Research*, Vol. 12, pp 41–50, 1987.

3. ABOYTES, F., CORY, B.J. "Identification of measurement parameter and configuration errors in static state estimation", *Conf. Proc. PICA* pp 298–302, 1975.

4. ABUR, A., KEYHANI, A., BAKHTIARI, H. "Autoregressive filters for the identification and replacement of bad data in power system state estimation", *IEEE Trans Power Systems*, Vol. PWRS–2, No. 3, pp 552–560, Aug. 1987.

5. AGNEW, S.C. "Water overview", *Proc. ICE*, Part 1, Vol 82, pp 381–396, Apr. 1987

6. ALEXANDER, S.M., GLENN, N.L., BIRD, D.W. "Advanced techniques in the mathematical modelling of water distribution systems", *Jour. AWWA*, 67, 7, pp 343, July 1975.

7. ALLAM, M.F. "Exact static state estimation algorithm retaining non-linearity", 7^{th} PSCC, pp 1007–1013, July 1981.

8. AMERICAN NATIONAL STANDARDS INSTITUTE DATA BASE LANGUAGE SQL, Document ANSI X3.135-1986, 1986.

9. ANDERSON, M.D., SMITH, R.A. "Data bases for energy management systems – a real-time operational approach", *Power Industry Computer Applications Conf.*, pp 74–82, 1977.

10. ARIAV, G. "Design requirements for temporally oriented information systems", *Proc. Conf. Temporal Aspects in Information Systems*, AFCET, France, pp 9–23, May 1987.

11. AYRES, M., HALEY P.H. "Bad data groups in power system state estimation", Vol. PWRS-1, No. 3, pp 1–9, Aug. 1986.

12. BARGIELA, A.B. "On-line monitoring of water distribution networks", Ph.D. thesis, University of Durham, May 1984.

13. BARGIELA, A.B., IRVING, M.R., STERLING, M.J.H., "Observability determination in power systems state estimation using network flow techniques". *IEEE Trans PWRS*, PWRS-1, (2), pp 109–114, 1985.

14. BARLOW, J.F., MARKLAND, E. "Computer analysis of pipe networks", *Proc. I.C.E.*, 43, pp 249–259, May–Aug 1969.

15. BARLOW, J.F., MARKLAND, E. "Computer design of pipe networks", *Proc. I.C.E.*, 51, pp 225–235, Aug 1972.

16. BARR, D.I.H. "Two additional methods of direct solution of the Colebrook-White function", *Proc. I.C.E.*, Part 2, 59, pp 827–835, 1975

17. BAUMANN, R., BAYER, R. "Data bases for power system calculation", Proc. 7^{th} Power Systems Control Conference (PSCC), Lausanne, pp 3–8, 1981.

18. BAUMANN, R., BAYER, R. "Data bases for power system calculation", *Electrical Power and Energy Systems*, Vol. 4, No. 1, pp 7–10, Jan. 1982.

19. BAZARAA, M.S., JARVIS, J.J. LINEAR PROGRAMMING AND NETWORK FLOWS, John Wiley & Sons, 1977.

20. BAZARAA, M.S., SHETTY, C.M. NONLINEAR PROGRAMMING: THEORY AND PRACTICE, John Wiley & Sons, 1979.

21. BECHER, A.E. (Jr.), BIZJAK, G.J., SHULZ, J.W. "Computer techniques for water distribution analysis", *Jour. AWWA*, 64, 7, pp 410, July 1972.

22. BECK, P., LASDON, L., ENGQUIST, M. "A reduced gradient algorithm for non-linear network problems", *ACM Trans. Mathematical Software*, Vol. 9, No. 1, pp 57–70, March 1983.

23. BEN-YAACOV, G.Z. "Interactive computation and data management for power system studies", *The Computer Journal*, Vol. 22, No. 1, pp 76–79, 1979.

24. BESSEY, S.G. "Progress in pressure control", *AQUA*, No. 6, pp 325–330, 1985.

25. BESSONEAU, M. "Remote control of the interconnected water supply system in Vendee", *Techniques et Science Municipales*, 77, 12, pp 565–568, 1982.

26. BOLDY, A.P. "Simulation of transient phenomena in internal flow systems", in

COMPUTER APPLICATIONS IN WATER SUPPLY, VOL 1, COULBECK B., ORR C-H., Wiley, pp 383–408, 1988.

27. BONANOMI, P., GRAMBERG, G. "Power system data validation and state calculation by network search techniques", IEEE PES Summer Meeting, S. Fransisco, July 18–23, 1982.

28. BONANSIGA, A.P. "Computer analysis of water distribution systems", *Jour. AWWA*, 67, 7, pp 347, July 1975.

29. BOND, P.G. "Delta-4 Dependable data base", ESPRIT II Delta-4, Phase 3, W90.157-/I1/C. 1990.

30. BOYCE, R.F. CHAMBERLIN, D.D. "Using a structured query language as a data definition facility", IBM Research Report RJ 1318, Dec. 1973.

31. BRIGGS, R. "Instrumentation and control in the U.K. water industry", *AWWA, Sem. Proc. Water Plant & Automation*, 1976.

32. BROUSSOLLE, F. "State estimation in power systems: Detecting bad data through the sparse inverse matrix method", *IEEE Trans PAS*, Vol. PAS-97, No. 3, pp 678–682, May/June 1978.

33. BROOKER, A. "New technology: Improving the efficiency of the process", *Municpal Journal*, 91, No. 10, 1983, pp 306–308.

34. BUNCH, J.R., ROSE, D.J. SPARSE MATRIX COMPUTATIONS Academic Press Inc. London, 1976.

35. BURCH, R.H., MARLOW, K.C. "A computer based telecontrol and communication system for a water supply network", *Jour. Hydrology*, 28, pp 407–427, 1976.

36. BURCH, R.H., MARLOW, K.C. "Seven years operational experience of computer based telemetry and control applied to a water supply network", *Jour. Inst. of Water Engrs. & Scientists*, 32(5), pp 443, 1978.

37. CAGNON, C.R., JACOBY, S.L.S. "Computer simulation of water distribution networks", *Jour. Trans. Eng. Div. ASCE*, 101, TE3, pp 553–567, Aug 1975.

38. CARPENTIER, P., COHEN, G. "State estimation and leak detection in water distribution networks", French/UK Workshop on Advances in Operational Control of Water Distribution Networks, Heriot-Watt University, 3-5 April 1991.

39. CAVES, J.L., EARL, T.C. "Computer applications: A tool for water distribution engineering", *Jour. AWWA*, pp 231–235, May 1979.

40. CESARIO, A.L. "Computer modelling programs: Tools for model operations", *Jour. AWWA*, pp 508–513, Sept. 1980.

41. CHAMBERLIN, D.D., BOYCE, R.F. "SEQUEL: A structured English query language", *Proc.ACM* Workshop on Data Description, Access and Control, Michigan, May 1974.

42. CHAMBERLIN D.D., et.al. "SEQUEL2: A unified approach to data definition, manipulation and control", *IBM Jour.R&D*, **20**, No. 6, November 1976. (also errata: *IBM Jour.R&D*, **21**, No. 1, January 1977.

43. CHANACHOWICZ, P., SCHUBERT, J. "Transformation methods in state estimation – existence, uniqueness and stability theorems (a geometric approach)", *IEEE Trans PAS*, Vol. PAS-100, No. 2, pp 449–457, Feb. 1981.

44. CHANDRASHEKA, M., STEWART, K.H. "Sparsity analysis of large pipe networks", *Jour. Hydr. Div. ASCE*, HY4, **101**, pp 341–355, Apr. 1975.

45. de CHANTERAC, J "A new remote control system for a water distribution network", *Eau et Industrie*, **61/62**, pp 37–42, 1982.

46. CHEN, P.P-S. "The entity-relationship model – toward a unified view of data", *ACM Trans on Data Base Systems*, Vol. 1, No. 1, pp 9–36, March 1976.

47. CHEN, Y-C. "Analysis, modelling and optimal control of water supply and distribution systems", Ph.D. thesis, Liecester Polytechnic, Dec. 1989.

48. CHEN, Y-C., COULBECK, B., ORR, C.H. "Network simplification by equivalent modelling", Research Report 64, School of Electronic and Electrical Engineering, Leicester Poly., March 1988.

49. CHEN, Y-C., POWELL, R.S. "Pressure reduction through on-line monitoring", *Digest of IEE Colloquium on Control and Optimisation Techniques in the Water Industry*, No. 1990/071, pp 6/1–6/4, May 1990.

50. CHRISTENSEN, G.S., SOLIMAN, S.A. "A new technique for linear static state estimation based on weighted least absolute value approximations", *Jour. Optimization Theory and Apps.*, Vol. 61, No. 1, pp 123–136, April 1989.

51. CHRISTENSEN, G.S., SOLIMAN, S.A. "Optimal filtering of linear discrete dynamic systems based on least absolute values approximations", *Autommatica*, Vol. 26, No. 2, pp 389–395, 1990.

52. CLARK, R.M., GOODRICH, J.A. "Developing a data base on infrastructure needs", *Jour.AWWA*, Vol 81, Part 7, pp 81–87, July 1987.

53. CLARKE, D.A., McBEAN, E.A., AL-NASSRI, S.A. "Uncertainties in water distribution systems", *Jour. Hydr. Div. ASCE*, HY10, **107**, pp 1263–1267, 1981.

54. CLEMENTS, K.A., DAVIS, P.W. "Multiple bad data detectability and identifiability: A geometric approach", IEEE Proc. PICA, pp 461–466, 1985.

55. CLEMENTS, K.A., DAVIS, P.W. "Detection and identification of topology errors in electrical power systems", *IEEE Trans on Power Systems*, Vol. 3, No. 4, pp 1748–1753, Nov. 1988.

56. CLEMENTS, K.A., KRUMPHOLZ, G.R., DAVIS, P.W. "Power system state estimation residual analysis: An algorithm that uses network topology", *IEEE Trans PAS*, Vol. PAS-100, No. 4, pp 1779–1787, April 1981.

57. CLEMENTS, K.A., KRUMPHOLZ, G.R., DAVIS, P.W. "Power system state estimation with measurement deficiency: An algorithm that determines the maximal observable subnetwork", *IEEE Trans PAS*, Vol. PAS-101, No. 9, pp 3044–3052, Sept. 1982.

58. CLEWER, B.C. "State estimation and bad data detection in electrical power systems", Ph.D. Thesis, Univ. of Durham, 1986.

59. CODASYL SYSTEMS COMMITTEE *Introduction to* "Feature analysis of generalised data base management systems", *BCS Comp.Bull.* **15**, No. 4, April 1971.

60. CODD, E.F. "A relational model of data for large data banks", *Comm. ACM*, Vol. 13, No. 6, pp 377–387, June 1970.

61. CODD, E.F. "Further normalization of the data base relational model" *Data Base Systems*, Courant Computer Science Symposia Series, Vol. 6, Englewood Cliffs, N.J.: Prentice-Hall 1971.

62. CODD, E.F. "Normallised data base structure: a brief tutorial" *Proc.ACM*, SIGFIDET Workshop on Data Description, Access and Control, San Diego, Calif., 1971.

63. CODD, E.F. "Extending the data base relational model to capture more meaning", *ACM Trans.Data Base Systems*, Vol. 4, No. 4, pp 397–434., Dec. 1979

64. CODD, E.F. "Is your data base really relational?", *Computerworld*, October, 1985.

65. CODD, E.F. "Does your data base run by the rules?", *Computerworld*, October, 1985.

66. COHEN, G., CARPENTIER, P. "State estimation and leak detection in water networks", in COMPUTER APPLICATIONS IN WATER SUPPLY, VOL 1, COULBECK B., ORR C-H., Wiley, pp 317–332, 1988.

67. COLEBROOK, C.F. "Turbulent flow in pipes, with particular reference to the transition region between the smooth and the rough pipe laws", *Jour. ICE*, Vol 11, pp 133–156, 1938–1939.

68. COLLINS, P.R. "Notes on the collection of field data for mains network analysis", *Jour. Inst. Water Engrs.*, **28**, pp 332-337, August 1974.

69. COLLINS M., COOPER, L., HELGASON, R., KENNINGTON, J., LE BLANC, L. "Solving the pipe network analysis problem using optimisation techniques", *Management Science*, **24**, 7, pp 747–760, March 1978.

70. COULBECK, B. "Optimisation and modelling techniques in dynamic control of water distribution systems", Ph.D. thesis, Dept. of Control Eng., Univ. of Sheffield, July 1977.

71. COULBECK, B. "Optimal operations in non-linear water networks", *Optimal Control Applications & Methods*, Vol. 1, pp 131–141, 1980.

72. COULBECK, B., ORR, C.H. "A network analysis and simulation program for water distribution systems", Civ. Engng. Syst., Vol. 1, pp 139–144, Mar. 1984.

73. COULBECK, B., STERLING, M.J.H., "Modelling techniques in dynamic control of water distribution systems", *Trans. Inst.M.C.*, Vol 11, pp 385–389, Oct 1978.

74. COULBECK, B., STERLING, M.J.H. "Optimised control of water distribution systems", *Proc IEE*, Vol. 125, No. 9, pp 1039–1044, Oct 1978.

75. CRABBE R., FIELD D.B., LACEY, R.F. "A guide to water network analysis", Technical Report, TR78, Water Research Centre, 1978.

76. CROSS, H. "Analysis of flow in networks of conduits or conductors", Bulletin 286, Univ. of Illinois, Eng. Experimentation Station, Urbana, Illinois, 1936.

77. CROW, W.B., EIDNESS, F.A. "Savings through instrumentation and control in two water systems", *Jour. AWWA*, 57, 12, pp 1509, Dec 1965.

78. CURTIS, A.R., REID, J.K. "On the automatic scaling of matrices for Gaussian elimination", *Jour. Inst. Maths Apps*, **10**, pp 118–124, 1972.

79. DAMBOURG, M.J., RAMASWAMI, R., JAMPALA, A.K., VENKATA, S.S. "Application of relational data base to computer aided engineering of transmission protection systems", *IEEE Trans. Power Systems*, Vol. PWRS-1, No. 2, May 1986, pp 187–193

80. DATA BASE TASK GROUP: PROGRAMMING LANGUAGE COMMITTEE Report, April 1971.

81. DATE C.J. A GUIDE TO INGRES, Addison-Wesley (pubs.), 1987.

82. DATE C.J. AN INTRODUCTION TO DATABASE SYSTEMS, Vol 1, Fifth Ed., *The Systems Programming Series*, Addison-Wesley (pubs.), 1990.

83. DATE C.J. AN INTRODUCTION TO DATABASE SYSTEMS, Vol 2, *The Systems Programming Series*, Addison-Wesley (pubs.), 1985.

84. DEMOYER, R. "A statistical approach to modelling and control of water distribution systems", Ph.D. thesis, Polytechnic Inst. of New York, Brooklyn, New York, 1973

85. DEMOYER, R. "A statistical approach to dynamic modelling and control of water distribution systems", NTIS, PB-266 604, June 1973.

86. DEMOYER, R., GILMAN, H.D., GOODMAN, M.Y. "Dynamic computer simulation and control methods for water distribution systems", Gen. Elect. Doc No. 73SD205, Philadelphia, PA, Feb 1973.

87. DEMOYER, R., HORWITZ, L.B. "Automatic control of water distribution", *ASME Jour. Dyn. Sys. Meas. & Control*, pp 126–130, June 1975.

88. DEMOYER, R., HORWITZ, L.B. "Macroscopic distribution system modelling", *Jour. AWWA*, **67**, 7, pp 377–380, Jul 1975.

89. DEMOYER, R., HORWITZ, L.B. A SYSTEM APPROACH TO WATER DISTRIBUTION MODELLING AND CONTROL, Lexington Books, Mass., 1975.

90. DEO, N. GRAPH THEORY WITH APPLICATIONS TO ENGINEERING AND COMPUTER SCIENCE, Prentice Hall Inc., Englewood Cliffs, N.J. 1974.

91. DICKENSON, C.C. "Waterworks telemetry objectives", Inst. Water Engrs., Symposium at Sutton Coldfield. Feb 1969.

92. DILLINGHAM, J.H. "Computer analysis of water distribution systems, (part 2)", *Water & Sewage Works*, 114, 2, pp 43–45, Feb 1967.

93. DILLINGHAM, J.H. "Automating distribution system operations", *Jour. AWWA*, pp 133–135, March 1979.

94. DIVENOT, A. "Localisation centralisee des rupturse dans un reseau de distribution d'eau", Techniues et Sciences Municpales (in French), 76, 8–9, pp 477–481, 1981.

95. DIVENOT, A. "Pinpointing pipe ruptures", *Water Eng. & Management*, 129, No.12, pp 36–37, 1982.

96. DODGE, E.P., HOELLEIN, H.R., TETMAJER, L. "The analysis of large, complex water networks with small computer systems", *Jour. AWWA*, 70, 7, pp 366, July 1978.

97. DONACHIE, R.P. "Digital program for water network analysis", *Jour. Hydr. Div. ASCE*, 100, No. HY3, pp 393–403, March 1974.

98. DOPAZO, J.F., KLITIN, O.A., SASSON A.M. "State estimation for power systems: Detection and identification of gross measurement errors", 8^{th} PICA conf., Minneapolis, pp 313–318, 1973.

99. DORAISWAMI, R. "An algorithm for pre-filtering of gross measurement errors in power system static state estimation", 7^{th} PSCC, pp 1014–1018, July 1981.

100. DUFF, I.S. "Practical comparisons of codes for the solution of sparse linear systems", from SPARSE MATRIX PROCEEDINGS, Duff & Stewart (Eds.), SIAM. ISBN: 0-89871-160-6, 1979

101. DUFF, I.S., REID, J.K. "A Comparison of Some Methods for the Solution of Sparse Overdetermined Systems of Linear Equations", *Jour. IMA.*, **17**, pp 267–280, 1976.

102. DY LIACCO, T.E., RAMARAO, K.A., WEINER, A.W. "Network status analysis for real-time systems", Conf. Proc. 8^{th} PICA, pp 356–362, 1973.

103. EGAWA, K., FUKAO, T. "Convergence and its improvement of bad data suppression (BDS) estimator in power systems", *Electrical Eng. in Japan*, Vol 95, No. 6,

pp 70–74, 1975.

104. EPP, R., FOWLER, G.A. "Efficient code for steady state flows in networks", *Jour. Hydraulics Div. ASCE*, Jan 1970.

105. FAGIN, R. "Multivalued dependencies and a new normal form for relational data bases", *ACM Trans on Data Base Systems*, Vol. 2, No. 3, pp 262–278, Sept. 1977.

106. FAGIN, R. "Normal forms and relational data base operators", ACM SIGMOD Conf. Boston, Mass., pp 153–160, 1979.

107. FALCAO, D.M., COOKE, P.A., BRAMELLER, A. "Power system tracking state estimation and bad data processing", *IEEE Trans PAS*, Vol. PAS-101, No. 2, pp 325–333, Feb. 1982.

108. FALCAO, D.M., KARAKI, S.H., BRAMELLER, A. "Non-quadratic state estimation: A comparison of methods", 7^{th}, PSCC, pp 1002–1006, Lausanne, 1981.

109. FALLSIDE, F., GRAY, D.F. MOSS, S.M., BURCH, R.H., JENKINS, D., MARLOW, K.C. "On-line control of water supply systems", *Proc. Water Tech.*, 9, 5–6, pp 289–298, 1977.

110. FALLSIDE, F., PERRY, P.F. "Hierarchical optimization of a water supply network", *Proc. IEE*, 122, 2, pp 202–208, Feb 1975.

111. FALLSIDE, F., PERRY, P.F. "On-line prediction of consumption for water supply control", Proc. 6^{th} IFAC World Congress, Boston, Mass. Aug 1975.

112. FIELD, D.B., GOODWIN, S. "U.K. gets to grips with water leakage control problems",*Municipal Jour.*, 91, **10**, pp 310–312, 1983.

113. FIELD, D.B., RATCLIFFE, B. "Location of leaks in pressurised pipelines using sulphur hexafluoride as a tracer", Rept. TR80, Water Research Centre, 1978.

114. FOX, J.A. HYDRAULIC ANALYSIS OF UNSTEADY FLOW IN PIPE NETWORKS, MacMillian.

115. FRONTINUS, J.S. "De aquis urbis Romæ", Loeb Classical Library

116. GANN, J.O., IRVING M.R., STERLING M.J.H. "Programming in logic for power systems control", I.E.E. Conf. Proc. Power Systems Monitoring and Control, No266, 8-11 July 1986.

117. GANN J.O. "Integrated graphical simulation and analysis environment for power distribution systems – data base evaluation", University of Durham, School of Engineering and Applied Science, Progress Report, April 1989.

118. GANN J.O. "Integrated graphical simulation and analysis environment for power distribution systems – data base evaluation (cont'd.)", University of Durham, School of Engineering and Applied Science, Progress Report, May 1989.

119. GENTLE, J.E. "Least absolute values estimation: An introduction", *Comm. Stat. Simul. Comp.*, (B), 6, pp 313–328.

120. GERLT, J.L., HADDIX, G.F., "Distribution system operation analysis model", *Jour. AWWA*, 67, 7, pp 381, July 1975.

121. GERMANOPOULOS, G. "Modelling and operational control of water supply networks", Ph.D. Thesis, University of London, March 1988

122. GERMANOPOULOS, G., JOWITT, P.W. "Leakage reduction by excess pressure minimization in a water supply network", Proc. ICE, Part 2, Vol. 87, pp 195–214, June 1989.

123. GILMAN, H.D., GOODMAN, M.Y., DEMOYER, R. "An experiment in computer assisted supervisory control of a water distribution system", NTIS, PB-213, June 1971.

124. GILMAN, H.D., GOODMAN, M.Y., METKOWSKI, R. "Final report — Mathematical modelling of water distribution systems", Gen. Elec. Doc. No. 71SD241, Philadelphia, Pa. July 1971.

125. GJELSVIK, A., AAM, S., HOLTEN, L. "Application of sparse matrix linear least squares methods in static power system state estimation", 8^{th} PSCC, pp 600–604, Aug. 1984.

126. GJELSVIK, A., HOLTEN, L., AAM, S. "Solution methods for sparse weighted least squares problems in power system static state estimation", The Norwegian Research Institute of Electricity Supply, Technical Report TR-3099, July 1984.

127. GODERYA, F., METWALLY, A.A., MANSOUR, O. "Fast detection and identification of islands in power networks", *IEEE Trans PAS*, Vol. PAS-99, No. 1, pp 217–221, 1980.

128. GOODWIN, S.J. "The results of an experimental programme on leakage and leakage control", Rept. TR154, Water Research Centre, 1980.

129. GRUBBS, F.E. "Procedures for detecting outlying observations in samples", *Technometrics*, Vol. 11, No. 1, pp 1–21, Feb. 1969.

130. GRUNWELL, D., RATCLIFFE, B. "Location of underground leaks using the leak noise correlator", Rept. TR157, Water Research Centre, 1981.

131. GU, J.W., CLEMENTS, K.A., KRUMPHOLZ, G.R., DAVIS, P.W. "The solution of ill-conditioned power system state estimation problems via the method of Peters and Wilkinson", *IEEE Trans PAS*, Vol. PAS-102, No. 10, pp 3473–3480, Oct. 1983.

132. HAMAM, Y.M., BRAMELLAR, A. "Hybrid method for the solution of pipe networks", *Proc. IEE*, 118, (11), pp 1607–1612, Nov. 1971.

133. HANDSCHIN, E. "Survey paper A: Acquiring and securing data", IFAC Symposium s 39 – 83, Florence, pp 1–6, 1983.

134. HANDSCHIN, E., SCHWEPPE, F.C., KOHLOAS, J., FLETCHER A. "Bad data analysis for power system state estimation", IEEE, PES, Summer Meeting & Energy Resources Conf. Anahiem, Cal. July 14–19 1974.

135. HARTLEY, J.A., POWELL, R.S. "A combined approach to demand prediction", *Digest of IEE Colloquium on Control and Optimisation Techniques in the Water Industry*, No. 1990/071, pp 2/1–2/3, May 1990.

136. HARTLEY, J.A., POWELL, R.S. "The developement of a combined demand prediction system" *Civil Engineering Systems*, Vol 8, No. 4, Nov. 1991.

137. HARTLEY, J.A., POWELL, R.S. "A neural network application in water demand forecasting", submitted to the I.C.E., Proc. 2, Jan. 1992.

138. HASSAN, M., ALDEMIR, T. "A data base oriented dynamic methodology for the failure analysis of closed loop control systems in process plants", *Reliability Engineering and System Safety*, **27**, pp 275–322, 1990.

139. HOLTEN, L., GJELSVIK, A., AAM, S., WU, F.F., LIU, W-H.E. "Comparison of different methods for state estimation", *IEEE Trans PAS*, Vol. 3, No. 4, pp 1798–1806, Nov. 1988.

140. VAN HUFFEL, S., VANDEWALLE, J. "On the accuracy of total least squares and least squares techniques in the presence of errors on all data", *Automatica*, Vol 25, No. 5, pp 765–769, 1989.

141. HURSCH C.J., HURSCH J.L. SQL THE STRUCTURED QUERY LANGUAGE, Tab Books Inc. (pubs), 1988.

142. IRVING, M.R., OWEN, R.C., STERLING, M.J.H. "Power system state estimation using linear programming", IEE Proc., 125, 9, pp 879–885, Sept. 1978.

143. IRVING, M.R., STERLING, M.J.H. "State Estimation Methods for Large Scale Power Systems", I.E.E. Conf.: 'Control 85', Cambridge, 1985, pp 223–226.

144. IRVING, M.R., STERLING, M.J.H., "Robust State Estimation for Large Scale Power Systems", I.E.E. Conf.: 'Power Systems Monitoring and Control', Durham, England, 1986.

145. JEPPSON, R.W. ANALYSIS OF FLOW IN PIPE NETWORKS, Ann Arbour Science (Pubs), 1979.

146. JEPPSON, R.W. "Equivalent hydraulic pipe for parallel pipes" *Jour. AWWA*, 108, No. HY1, pp 33–45, 1982.

147. JOWITT, P.J., XU, C. "Optimal valve control in water distribution networks",*Jour. of Water Resources Planning and Management*, Vol. 116, No. 4, pp 455–472, July/Aug. 1990.

148. KESAVAN, H.K., CHANDRASHEKAR, M. "Graph-theorectic models for pipe network analysis", *Jour. Hydraulics Div. ASCE*, HY 2, pp 345–364, Feb. 1972.

149. KŁOS, A., DOBRZAŃSKI, A. "Identification of network topology", 7^{th} PSCC, pp 976–981, July 1981.

150. KOGLIN, H.J., OEDING, D., SCHMITT, K.D. "Local redundancy and identification of bad data", 9^{th} PSCC, Lisbon, pp 528–534, July 1987.

151. KOLLER, H., FRÜHAUF, B. "PRIMO Data base management system", Brown Boveri, Rev. 3-79, 1979, pp 204–209.

152. KOTIUGA, W.W., VIDYASAGAR, M. "Bad data rejection properties of weighted least absolute value techniques applied to static state estimation", *IEEE Trans PAS*, Vol. PAS-101, No. 4, pp 844–853, April 1982.

153. KRUMPHOLZ, G.R., CLEMENTS, K.A., DAVIS, P.W. "Power system observability: A practical algorithm using network topology", *IEEE Trans PAS*, Vol. PAS-99, No. 4, pp 1534–1542, July/Aug. 1980.

154. LEHMAN, R., MATTIONI, J. "Data base methods in power system control centres", *Electrical Power and Energy Systems*, Vol. 5, No. 4, pp 241–246, Oct. 1983.

155. LEITE da SILVA, A.M., DO COUTTO FILHO, M.B., de QUEIROZ, J.F. "State forecasting in electric power systems", *IEE PROC*, Vol. 130, Pt. C, No. 5, pp 237–244, Sept. 1983.

156. LEMIEUX, P.F. "Efficient algorithm for distribution networks", *Jour. Hydraulics Div. ASCE*, HY 11, pp 1911–1920, Nov. 1972.

157. LIU, W-H.E., HOLTEN, L., GJELSVIK, A. "Computational issues in the Hachtel's augmented matrix method for power system state estimation", Proc. 9^{th} PSCC, Portugal, pp 509–515, 1987.

158. LO, K.L., ONG, P.S., McCOLL, R.D., MOFFATT, A.M., SULLEY, J.L. "Development of a static state estimator part 1: Estimation and bad data suppression", *IEEE Trans PAS*, Vol 102, No. 8, pp 2486–2491, Aug 1983.

159. LO, K.L., ONG, P.S., McCOLL, R.D., MOFFATT, A.M., SULLEY, J.L. "Development of a static state estimator part 2: Bad data replacement and generation of pseudo-measurements", *IEEE Trans PAS*, Vol 102, No. 8, pp 2492–2500, Aug 1983.

160. LUGTU, R.L., HACKETT, D.F., LIU, K.C., MIGHT, D.D. "Power system state estimation: Detection of topological errors", *IEEE Trans PAS*, Vol. PAS-99, No. 6, pp 2406–2412, Nov/Dec 1980.

161. MANNING, A.W. "Digital control of water treatment plants and distribution systems", Proc. AWWA, Plant Instrumentation Seminar, 96^{th} Annual Conf., New Orleans, June 1976.

162. MANSOUR, A.E., NOUR ELDIN, H.A. "Fast and robust bad data detection, identification and correction with subsequent direct recursive state estimation", 9^{th} PSCC,

pp 516–522, Sept. 1987.

163. MARLOW, K.C., FALLSIDE, F. "Minicomputer, microprocessor and telecontrol applications to a water supply network", *Jour. Inst. Water Engrs. & Scientists*, 34, 6, pp 517–545, 1980.

164. MARTIN D.W., PETERS, G. "The application of Newton's method to network analysis by digital computer", *Jour. Inst. Water Engrs.*, 17, (2), pp 115–129, March 1963.

165. MASSEY, B.S. MECHANICS OF FLUIDS, 4^{th} edition, Van Nostrand Reinhold.

166. MATTHEW G.D. "The Colebrook-White equation — an oft cited result but neglected derivation?", *Proc. ICE*, Part 2, Vol 89, pp 39–45, Mar. 1990.

167. McCORMICK, M., de NEUFVILLE, R., HESTER, J. "Discussion on a paper by U. Shamir and C. Howard: 'Water Distribution Analysis' ", *Proc. ASCE, Jour. Hydr. Div.* Vol HY1, pp 481–486, Jan. 1969.

168. McCREDIE, A. "Monitoring and control in the water and sewage services: 1. Experience with automatic control and telemetry in water supply", *Public Health Engineer*, 11, No. 2, pp 13–17, 1983

169. McKENZIE, E., SNODGRASS, R., "An evaluation of historical algebras", TR87-020, Univ. of North Carolina, Dept. of Computer Science, October 1987.

170. MERRILL, H. M., SCHWEPPE, F. C. "Bad data suppression in power system static state estimation", *IEEE Trans PAS*, 90, No. 6, pp 2718–2725, Nov/Dec 1971.

171. MEYERS, S., SHAMIR, U. "Optimal annual operation of a supply distribution system", *Advances in Water Resources*, 5, No. 4, pp 240–247, 1982.

172. MILI, L., VAN CUTSEM Th., RIBBENS-PAVELLA, M. "Hypothesis testing identification: A new method for bad data analysis in power systems state estimation", *IEEE Trans PAS*, 103, No. 11, pp 3239–3252, Nov 1984.

173. MILI, L., VAN CUTSEM Th., RIBBENS-PAVELLA, M. "Bad data identification methods in power systems state estimation — A comparative study", *IEEE Trans PAS*, 104, No. 11, pp 3037–3049, Nov. 1985.

174. MILLER, D.S. INTERNAL FLOW. A GUIDE TO LOSSES IN PIPE AND DUCT SYSTEMS, Cranfield, BHRA, 1971.

175. MIYAOKA, S., FUNABASHI, M. "Optimal control of water distribution systems by network flow theory", *IEEE Trans Automatic Control*, Vol. AC-29, No. 4, pp 303–311, April 1984.

176. MONTICELLI, A. GARCIA, A. "Reliable bad data processing for real-time state estimation", *IEEE Trans PAS*, 102, 5, pp 1126–1139, 1983.

177. MONTICELLI, A., MURARI, C.A.F., WU, F.F. "A hybrid state estimator: solving normal equations by orthogonal transformations", *IEEE Trans PAS*, Vol. PAS-104, No. 12, pp 3460–3468, Dec. 1985

178. MONTICELLI, A., WU, F.F., MAOSONG, Y. "Multiple bad data identification for state estimation by combinatorial optimization", *IEEE Trans. Power Delivery*, Vol PWRD-1, No. 3, pp 361–369, July 1986.

179. MOON, Y.H., PARK, Y.M., LEE., K.J. "Observable island identification for state estimation using incidence matrix", (brief paper), *Automatica*, Vol.24, No.1., pp 71–75, 1988.

180. MOORE W.L. "Relationships between pipe resistance formulas", *Proc. ASCE, Hydraulics Div.*, HY-3, Vol. 85, pp 25–41, Mar. 1959.

181. MOSS S.M. "On-line optimal control of a water supply system", Ph.D. Thesis, Dept. of Eng., Univ. of Cambridge, Sept. 1979.

182. NAGDY, N.M., RUMPEL, D. "Methods of potential network topology description", *Electrical Power & Energy Systems*, Vol. 5, No. 1, pp 55–60, Jan. 1983.

183. NATIONAL WATER COUNCIL "Water industry review 1982", NWC, Feb. 1982.

184. NATIONAL WATER COUNCIL "Water industry review 1982 — Supporting analysis", NWC, March 1982.

185. de NEUFVILLE, R., SCHAAKE, J.(Jr.), STAFFORD, J.H. "Systems analysis of water distribution networks", *Jour. Sanitary Eng., Div. ASCE*, 97, 6, pp 825, Dec 1971.

186. NISHIYA, K-I., TAKAGI, H., HASEGAWA, J., KOIKE, T. "Dynamic state estimation including detection of innovation process for electric power systems", *Electrical Engrg. in Japan*, Vol. 98, No. 1, pp 52–61, Jan 1978.

187. OLSSON, G., ANDERSSON, B., HELLSTRÖM, G.G., HOLMSTRÖM, H., REINIUS, L.G., VOPATEK, P. "Measurements, data analysis and control methods in wastewater treatment plants — state of the art and future trends", *Wat. Sci. Tech.*, Vol. 21, Brighton, pp 1333–1345, 1989.

188. PELJTO, H. "A new method for bad data processing in power system state estimation", 8^{th} PSCC, pp 592–599, Aug. 1984.

189. PERRY, P.F. "Computer simulation of pumping policies in a water distribution network", *Simulation*, 37, No. 1, pp 19–26, 1981.

190. PILZER, J.E. "Leak detection — Case histories", *Jour. AWWA*, 73, No 11, pp 565–567, 1981.

191. POWELL, R.S. "WASMACS: Functional description", Brunel Univ. Internal Report, Rev. 3.2, Oct. 1991

192. POWELL, R.S., HARTLEY, J.A. "The role of expert systems and numerical algorithms in measurement validation and time series prediction", International Unicom conference on Expert Systems and Optimisation in Process Control, Brunel University, March 1991.

193. POWELL, R.S., IRVING, M.R., STERLING, M.J.H., USMAN, A. "A Comparison of Three Real Time State Estimation Methods for On-line Monitoring of Water Distribution Systems" in COMPUTER APPLICATIONS IN WATER SUPPLY, VOL 1, COULBECK B., ORR C-H., Wiley, pp 333–348, 1988.

194. POWELL, R.S., SMITH, M.M., STERLING, M.J.H. "Robust on-line data validation for real-time control of water distribution networks", COMPUTATIONAL MODELLING & EXPERIMENTAL METHODS IN HYDRAULICS, Macsimovic and Radojkovic (Eds), pp 279–288, Elsevier Applied Science, 1989.

195. POWELL, R.S., STERLING, M.J.H. "A technique for on-line monitoring of water distribution systems", *Proc. IWSA, WATER '89*, Bangkok, Thailand, 1989.

196. POWELL, R.S., STERLING, M.J.H., BARGEILA, A.B. "Real-time network analysis and leakage detection research project", Final Report, Severn-Trent Water Authority, October 1985.

197. PRICE, M. "Leaks that could sink our cities", *New Scientist,* pp 58–59, 24^{th} July, 1986.

198. QUEVEDO, J., CEMBRANO, G., MONTOLIU, J.M., CASANOVA, F. "Robust digital control of valves in a large water network", in COMPUTER APPLICATIONS IN WATER SUPPLY, VOL 2, COULBECK B., ORR C-H., Wiley, pp 146–164, 1988.

199. QUINTANA, V.H., SIMÕES-COSTA, A. "Bad data detection and identification techniques using estimation orthogonal methods", *IEEE Trans PAS*, Vol 101, No. 9, pp 3356–3364, Sept. 1982.

200. VON RADER, J.E. "Telemetric control of water distribution systems", *Jour. AWWA, Management and Operations*, pp 542–549, Nov. 1983.

201. RAHAL, C.M., STERLING, M.J.H., COULBECK, B. "Parameter tuning for simulation models of water distribution networks", *Proc.I.C.E.*, Part 2, No. 69, pp 751–762, Sept. 1980.

202. RAHAL, C.M., STERLING M.J.H. "Dynamic parameter tuning for water distribution network models in an extended period simulation", *Proc I.C.E.*, Part 2, No. 71, pp151–164, March 1981.

203. RAHAL, C.M. "Parameter estimation for water distribution network modelling and control", Ph.D. thesis, University of Sheffield, April 1980.

204. RAO, H.S., BREE, D.W., BENZVI, R. "Extended period simulation of water distribution networks", Rept. PB–230 148, Systems Control Inc., Palo Alto, Feb 1974

205. REID, J.K. (Ed.) LARGE SPARSE SETS OF LINEAR EQUATIONS, Academic Press, 1971.

206. REID J.K. "A Sparsity exploiting variant of the Bartels-Golub decomposition for linear programming bases", A.E.R.E., Harwell Report No. CSS20, 1975.

207. REISNER, P., BOYCE, R.F., CHAMBERLIN, D.D. "Human factors evaluation of two data base query languages: SQUARE and SEQUEL", *Proc.NCC*, **44**, AFIPS Press, May 1975.

208. RUMPEL, D., POST, U., ZALUK, R. "Application of a grid data language for power system data base definition and query", IFAC Symposium, Brussels, pp 73–79, 5-8 Sept., 1988.

209. RUMPEL, D., ZALUK R., POST, U. "Concept of an on-line data base supporting grid data language", Power System Control Conference (PSCC), Oissbon, pp 345–351, 1987.

210. SALGADO, R., TODINI, E., O'CONNELL, P.E. "A comparison of the gradient method with some traditional methods for the analysis of water supply ditribution networks", in COMPUTER APPLICATIONS IN WATER SUPPLY, VOL 1, COULBECK B., ORR C-H., Wiley, pp 38–62, 1988.

211. SANSOM. P. "Twelve million gallons a day controlled by one man and telemetry", *Water and Waste Treatment*, 25, No. 10, pp 30–31, 1982.

212. SASAKI, H., NAKASHIMA, M., TAMARU, Y. "A bad data detection algorithm in power system state estimation", *Electrical Eng. in Japan*, Vol. 98, No. 5, pp 108–115, Oct. 1978.

213. SASSON, A.M., EHRMANN, S.T., LYNCH, P., VAN SLYCK, L.S. "Automatic power system network topology determination", *IEEE Trans PAS*, Vol. PAS-92, pp 610–618, 1973.

214. SCHLAEPFER, F., STEWART, D., JORDAN, J.A. "Data bases in power systems operations and planning", in SYSTEMS ENGINEERING FOR POWER: STATUS AND PROSPECTS, Fink, L.H., Carlsen, K. (Eds.), Energy Research and Development Administration, Washington D.C., USA. 1975

215. SCHWAB, B.C. et. al. "A developed data base management system for operation and planning systems", *IEEE Trans.PAS*, Vol. PAS-104, No. 7, pp 1637–1643, July 1985

216. SCHWEPPE, WIDES, J., ROM, D.B. "Power system static state estimation. Parts I, II, III", *IEEE Trans PAS.*, Vol PAS-89, No. 1, pp 120–135, Jan. 1970.

217. SIMÕES-COSTA, A., do AMARAL, S. "Multiple bad data identification techniques for orthogonal row-processing state estimators", 9^{th} PSCC, pp 523–527, Sept. 1987.

218. SINGH, N., GLAVISTCH, H. "Detection and identification of topological errors in online power systems analysis", *IEEE Trans on Power Systems*, Vol 6, No. 1, pp 324–331, Feb. 1991.

219. SIRISENA, H.R., BROWN, E.P.M. "Fast identification of bad data in power system tracking state estimation", *Electric Power Systems Research*, 4, pp 297–308, 1981.

220. SIRISENA, H.R., BROWN, E.P.M. "Convergence analysis of weighted least-squares and fast decoupled weighted least-squares state estimation", (short paper), *Electrical Power & Energy Systems*, Vol. 6, No. 2, pp 75–78, April 1984.

221. SLUTSKER, I.W. "Bad data identification in power system state estimation based on measurement compensation and linear residual calculation", *IEEE Trans Power Systems*, Vol. 4, No. 1, pp 53–60, Feb. 1989.

222. VAN SLYCK L.S., ALLEMONG, J.L. "Operating experience with the AEP state estimator", *IEEE Trans PAS*, Vol. 3, No. 2, pp 521–528, May 1988.

223. SMITH M.M., POWELL R.S., IRVING M.R., STERLING M.J.H. "A Robust Algorithm for State Estimation in Electrical and Water Networks", *IEE Proc. C*, Vol 138, No. 4, pp 283–288, July 1991.

224. SNODGRASS, R. "The temporal query language: TQUEL", *ACM Trans. Database Systems*, **12**, No. 2, pp 247–298, June 1987.

225. SORENSON H.W. PARAMETER ESTIMATION, Series on Control and Systems Theory, Vol. 9, Marcel Dekker Inc. (Pubs.), 1980.

226. SOWBY, S.E. "Leak detection programmes recover revenues", *Jour. AWWA*, 13, No. 11, pp 562–564, 1981.

227. STANDING TECHNICAL COMMITTEE ON SEWERS AND MAINS "Sewer and main records", Report 25, NWC/DoE, June 1980.

228. STANDING TECHNICAL COMMITTEE ON SEWERS AND MAINS "Leakage control: Policy and Practice", Report 26, NWC/DoE, 1981.

229. STANDING TECHNICAL COMMITTEE ON SEWERS AND MAINS "Progress report 1978–81", Report 35, NWC/DoE, July 1982.

230. STANDING TECHNICAL COMMITTEE ON WASTEWATER TREATMENT "Working party on control systems for the water industry – First Interim Report", No. 9, NWC/DoE, June 1978.

231. STANDING TECHNICAL COMMITTEE ON WASTEWATER TREATMENT "A glossary of terms used in measurement and control systems for the water industry", No. 11, NWC/DoE, June 1978.

232. STANDING TECHNICAL COMMITTEE ON WASTEWATER TREATMENT "Working party on control systems for the water industry – Final Report", No. 27, NWC/DoE, May 1981.

233. STERLING, M.J.H., ANTCLIFFE, D.J. "A technique for the prediction of water demand from past consumption data", *Trans Inst. Water Engrs.*, Vol 28, No. 8, pp413–420, Nov 1974.

234. STERLING, M.J.H., BARGIELA, A.B. "An adaptive forecasting technique based on time series modelling", IEE Coloquium on Time series analysis and forecasting methods, London, pp 2/1–2/2, 1983.

235. STERLING, M.J.H., BARGIELA, A.B. "Minimum norm state estimation for water distribution networks", *Proc IEE*, Vol 131, Pt. D., No. 2, pp57–63, 1984.

236. STERLING, M.J.H., BARGIELA, A.B. "Leakage reduction by optimised control of valves in water networks", *Trans Inst M.C.*, Vol 6, No. 6, Oct–Dec 1984.

237. STERLING, M.J.H., BARGIELA A.B. "The enhancement of time series modelling techniques for adaptive forecasting", Paper in SHORT TERM FORECASTING, BUNN D.W., FARMER, E.D., (Eds), J. Wiley & Sons, 1984.

238. STERLING, M.J.H., BARGIELA, A.B. "Adaptive forecasting of daily water demand", Bookchapter in COMPARATIVE MODELS FOR ELECTRICAL LOAD FORECASTING, ISBN 0471 906352, BUNN, D.W., FARMER, E.D., Wiley, pp 213–225, 1985.

239. STERLING, M.J.H., COULBECK, B. "Optimisation of water pumping costs by hierarchical methods", *Proc. (pt. 2) ICE*, Vol 59, pp 789–797, Dec 1975.

240. STERLING, M.J.H., COULBECK, B. "Linear dynamic models in water systems", *Proc. (pt. 2) ICE*, Vol 61, pp 582, (full paper 30pp, deposited in Institution Library), Sept. 1976.

241. STERLING, M.J.H., COULBECK B. "Analysis of pump interaction and efficiency in control of water supply", *Trans Inst.M.C.*, Vol 12 No. 2, pp 79–84, April 1979.

242. STERLING, M.J.H., COULBECK, B. "A dynamic programming solution to optimisation of pumping costs", *Proc. (pt. 2) ICE*, Vol 59, pp813–818.

243. STERLING, M.J.H., POWELL, R.S. "Real-time state estimation and control for water distribution networks", *Proc. ASCE* 16^{th} *annual conference on Water Resources Planning and Management*, Harris (Ed), pp 516–519, 1989.

244. STIMSON, K.R., BRAMELLER, A. "An integrated mesh-nodal method for steady-state water distribution network analysis", *Jour. Water Engrs. & Scientists*, 35, 2, March 1981.

245. STOER J. Einführung in die Numerische Mathematik 1. Springer, 1972.

246. STRANG G. LINEAR ALGEBRA AND ITS APPLICATIONS, Third Edition, Harcourt Brace Jovanovich, International Edition, 1988.

247. SULLIVAN, A.C., REICHERT, K., SALY, S. "An on-line technique for network topology determination for use in security assessment studies", IEE Conf. on 'On-line Operation and Optimisation of Transmission & Distribution Systems', London, pp 61–66, 1976.

248. THORLEY, A.R.D., WOOD, D.J. "Fault analysis and fluid transient suppression in pipelines and networks", in COMPUTER APPLICATIONS IN WATER SUPPLY, VOL 1, COULBECK B., ORR C-H., Wiley, pp 363–382, 1988.

249. TINNEY, W.F., MEYER, W.S. "Solution of large sparse systems by ordered triangular factorization", *IEEE Trans. Automatic Control*, 18, 4, pp 333–346, Aug. 1973.

250. TINNEY, W.F., WALKER, J.W. "Direct solutions of sparse network equations by optimally ordered triangular factorization", *Proc. IEEE*, Vol 55, pp 1801–1809, Nov. 1967.

251. TODD, S.J.P. "The Peterlee relational test vehicle", *IBM Systems Journal*, Vol 15, Part 4, pp 285–308, 1976.

252. TODINI, E., PILATI, S. "A gradient algorithm for the analysis of pipe networks", in COMPUTER APPLICATIONS IN WATER SUPPLY, VOL 1, COULBECK B., ORR C-H., Wiley, pp 1–20, 1988.

253. TWIGG, D., KEILLER, A. "Control should cut Oman's cost", *Jour. World Water*, Jan/Feb, 1985.

254. USMAN A., POWELL R.S., STERLING M.J.H. "Comparison of the Colebrook-White and Hazen-Williams flow models in real-time water network simulation", in COMPUTER APPLICATIONS IN WATER SUPPLY, VOL 1, COULBECK B., ORR C-H., Wiley, pp 21–37, 1988.

255. WATER AUTHORITIES ASSOC. WATER FACTS, W.A.A., London, 1985.

256. WATER RESEARCH CENTRE "Organising information on water distribution networks", Non-series publication, N58, W.R.C., 1984.

257. WESSEX WATER AUTHORITY , WESSEX WATER AUTHORITY ANNUAL REPORT AND ACCOUNTS, 1983–84. Published by Wessex Water Authority, 1984.

258. WILLIAMS, J.D., PESTOWSKI, T. "Distribution system analysis by digital computer", *Jour. Inst. Water Engrs.* Vol 26, pp 381–392, Oct. 1972.

259. WILSON, J.N. "A network program using either the Hazen-Williams or the Colebrook-White formulæ", *Jour. Inst. Water Engrs.*, Vol 29, pp 139–148, May 1975.

260. WOOD, D.J., CHARLES, C.O.A. "Hydraulic network analysis using linear theory", *Jour. Hydr. Div., ASCE*, 98, 7, pp 1157–1170, 1972.

261. WOOD, D.J., RAYES, A.G. "Reliability of algorithms for pipe network analysis", *Proc. ASCE, Hydraulics Div.*, HY-10, Vol. 107, pp 1145–1161, Oct. 1981.

262. WU, F.F., WEN-HSIUNG, E.L. "Detection of topology errors by state estimation", *IEEE Trans Power Systems*, Vol. 4, No. 1, pp 176–183, Feb. 1989.

263. XIANG NIAN-DE, WANG SHI-YING "Estimation and identification of multiple bad data in power system state estimation", 7^{th} PSCC, pp 1061–1065, July 1981.

264. XIANG NIAN-DE, WANG SHI-YING "A new approach for detection and identification of multiple bad data in power system state estimation", *IEEE Trans PAS*, Vol 101, No. 2, Feb. 1982.

265. XU, C., POWELL, R.S. "Water supply system reliability: Concepts and measures", *Civil Engineering Systems*, Vol. 8, No. 4, Nov. 1991.

266. XU, C., POWELL, R.S. "Leakage reduction and security enhancement for water distribution networks by on-line pressure control", submitted for publication in *Civil Engineering Systems*, Jan. 1992.

267. ZESSLER, U., SHAMIR, U. "Optimal operation of water distribution systems", *Jour. of Water Resources Planning and Management*, Vol. 115, No. 6, pp 735–751, Nov. 1989.

268. ZHI-QIANG, M. "Bad data reestimation-identification using residual sensitivity matrix", 7^{th}, pp 1056–1060, Lausanne, July 1981.

269. ZHUANG, F., BALASUBRAMANIAN, R. "Bad data suppression in power system state estimation with a variable quadratic-constant criterion", IEEE Trans PAS, Vol 104, No. 4, pp 857–863, April 1985.

270. ZHUANG, F., BALASUBRAMANIAN, R. "Bad data processing in power system state estimation by direct data deletion and hypothesis tests", *IEEE Trans on Power Systems*, Vol. PWRS-2, No. 2, pp 321–330, May 1987.

APPENDIX 1

A.1.1 Constitution of a Relational Database

The concept of a relational database was first introduced in 1970 by E.F. Codd [60] who suggested that the adoption of a relational model of data would permit the development of a universal data sublanguage based upon applied predicate calculus. Further development work by Chamberlin and Boyce [41, 42, 174], and work on IBM's database, System R, resulted in the query language S.Q.L. (Structured Query Language). To maintain order in this rapidly developing field, Codd laid down a set of twelve rules [64, 65], preceded by an all encompassing Rule 0:

0. **Relational Database Management.** Any system advertised as, or claiming to be a relational database management system must be able to manage databases entirely through its relational capabilities.

1. **Representation of information.** All information in a relational database is represented explicitly at the logical level and in exactly one way — by values in a table.

2. **Guaranteed logical accessibility.** Each and every datum in a relational database is guaranteed to be logically accessible by resorting to a combination of table name, primary key value, and column name.

3. **Systematic representation of missing information.** Null values are supported in fully relational database management systems for representing missing information

and inapplicable information in a systematic way independent of data type.

4. **Dynamic on-line catalogue.** The database description is represented at the local level in the same way as an ordinary data, so that authorized users can apply the same relational language to its interrogation as they apply to the regular data.

5. **Comprehensive data sublanguage.** A relational system may support several languages and various modes of terminal use. There must be at least one language whose statements are expressible as character strings, and that is comprehensive in supporting all of the following ; data definition, view definition, data manipulation, integrity constraints, authorization and transaction boundaries.

6. **Updatable views.** All views that are theoretically updatable are also updatable by the system.

7. **High-level insert, update and delete.** The capability of handling a base relation or a derived relation as a single operand applies not only to the retrieval of data, but also to the insertion, update and deletion of data.

8. **Physical data independence.** Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage structure or access methods.

9. **Logical data independence.** Application programs and terminal activities remain logically unimpaired when information-preserving changes of any kind that theoretically permit unimpairment are made to the base tables.

10 **Integrity independence.** Integrity constraints specific to a particular database must be definable in the relational data sublanguage and storable in the catalogue, not

in the application program.

11 **Distribution independence.** Whether or not a system supports database distribution, it must have a data sublanguage that can support distributed databases without impairing application programs or terminal activities.

12 **The nonsubversion rule.** If a relational system has a low-level (single record at a time) language, this cannot be used to subvert or bypaths the integrity rules and constraints expressed in the higher level relational language (multiple records).

A.1.2 Functional Dependencies and Normalisation

The design of a relation in a relational data base is founded upon an understanding of the *functional dependencies* of the data items [46]. The functional dependencies arise from the nature of the relationships between related attributes in a tuple. For example, an attribute is functionally dependant if it's value is determined by (or depends upon) the value of some other attribute in the tuple. To explain this term in the context of the present work, consider the following simplified relation, called NODE. (The relation could hold all the details that are pertinent to a node in a water network but ignoring some of these does not detract from the discussion.)

NODE:

Node_i; Zone_i; Elevation; Min_Pressure; Population; No_Industry; No_Commerce; OS_Ref; Name.

In order that unique tuples of data can be retrieved independently from the data base, it is necessary to arrange that at least one attribute (or combination of attributes) is unique. The unique attributes (or combinations of attributes) are called *candidate keys*

and the candidate key chosen as the indexing term for the tuple is called the *primary key*. (It is not always necessary to use a key when updating or selecting records, any of the attributes can be used. However, use of a non-key attribute will often result in a set of tuples being selected.)

Let us assume in this network that 're-zoning' cannot take place and that within our network there are uniquely numbered zones and within each zone there are several uniquely numbered nodes. Hence in the NODE relation, the pair of attributes Node_i and Zone_i uniquely define one node in the whole network. The data values for any other attribute are dependent on the values that appear under this pair of attributes. We therefore say that Node_i and Zone_i functionally determine all the other *non-key* attributes in the i^{th} tuple. For this reason, this pair of attributes constitute a *composite candidate key* ('composite' because there are two attributes that make up the key). If 'Name' were unique this would become another candidate key but since no stipulation is to be put on the uniqueness of 'Name' it is rejected as a candidate key. Therefore in this simple case, there is only one candidate key and so this is the primary key for relation NODES.

A tuple (or row) in a relation R consists of a number of attributes $A_1, A_2, \ldots, A_n$. The i^{th} attribute in R is normally written as $R.A_i$ except that the relation name (R in this case) can be dropped when ambiguity cannot result. We can express the functional dependencies on a relation R as follows:

$$R.A_i \rightarrow R.A_j \tag{1}$$

which states that $R.A_j$ is functionally dependent on $R.A_i$. This functional dependency is known as *non-transitive* and $R.A_i$ is known as the *determinant*. A *transitive* functional dependency is written thus:

$$R.A_i \rightarrow R.A_j \rightarrow R.A_k \tag{2}$$

Two or more attributes are *mutually independent* if they can be updated independently of one another, for example the attributes Min_Pressure and Population.

Suppose, on the other hand, that these two attributes were dependent, perhaps the resident population determining the statutory minimum pressure via some formula,

$$pressure = f(population)$$

then if a change was made to the population it follows that a change must be made to the set minimum pressure. Note also that if the node were for some reason deleted then this particular relationship between the population and the minimum pressure would also be lost. In fact where a relationship like this exists within a tuple it is best to decompose the table into two tables (this is known as normalisation):

NODES_1:

Node_i; Zone_i; Elevation; Population; No_Industry; No_Commerce; OS_Ref; Name.

POP:

Population; Min_Pressure.

Here, the deletion of a tuple in the NODES_1 relation will not lose the relationship between a given population and the minimum pressure. Thus it is apparent that careless design of the data base tables can lead to difficulties with update and delete operations. These difficulties have long been recognised and have lead to the development of normalisation theory [61–63, 81–83, 105, 106].

Any relation in a relational data base can be in one of six *normal forms*. The simplest normal form is first normal form, known as 1NF and the most advanced normal form is fifth normal form or 5NF. There is also a variation on 3NF called Boyce/Codd normal form or BCNF. A relation that is in 2NF is also in 1NF, a relation in 3NF is also in 2NF and so on. Thus we can write:

$$5NF \supset 4NF \supset BCNF \supseteq 3NF \supset 2NF \supset 1NF.$$

Thus each normal form corresponds to a level in a hierarchy, the further down the hierarchy, the more strict the normalisation. Apart from some simple examples in the context of this work no formal exposition of normalisation practice is presented here, instead see the excellent discussion by Date in [82].

As can be seen from the above example with the NODES relation, the importance of normalisation becomes apparent when data base updates, deletes and selects are performed. A relation that is in 1NF simply means that each data item is atomic (a fact that is necessarily guaranteed in all relational data base systems) but 1NF does not carry any information about functional dependencies and therefore does not guarantee a robust data base design. Higher normal forms result in better data base design from a robustness point of view.

The first version of the NODES relation (where Min_Pressure is not dependent on Population) is in third normal form[1]. In the case where

$$\text{NODES.Population} \rightarrow \text{NODES.Min\_Pressure}$$

the NODES relation reverts to only in first normal form.

It is therefore obvious that functional dependence is purely a semantic matter and can only be decided upon when the meaning of the information stored as data is clear.

[1] In fact, because of the simplicity of the relation, it is actually in 5NF but the complexities that BCNF, 4NF and 5NF are designed to handle are not present in the NODES relation.

APPENDIX 2

FORMULATION OF THE STATE ESTIMATION PROBLEM

A2.1 Introduction

This appendix presents a brief formulation of the state estimation problem in order to support the discussions in Chapters 6 & 7. The notation is explained in the list at the beginning of this thesis.

A2.2 Measurement Equations

If

$$\mathbf{z}_{\mathbf{k}}^{0} = \mathbf{g}(\mathbf{x}_{\mathbf{k}}) \tag{1}$$

represents the k^{th} exact mapping (ie at the k^{th} time step) from the state variables $\mathbf{x}_{\mathbf{k}}$, $dim(\mathbf{x}_{\mathbf{k}}) = n$, to the set of values $z_{i_k}^{0}$ $i = 1, \ldots m$, corresponding to the k^{th} set of $m(> n)$ actual measurements $\mathbf{z}_{\mathbf{k}}$ via known (non-linear) functions, $\mathbf{g}(\cdot)$, where the k^{th} actual measurements are obtained from transducers and/or from *a priori* knowledge (e.g. nodal, lumped demand values). Then it follows that

$$\mathbf{z}_{\mathbf{k}} = \mathbf{g}(\mathbf{x}_{\mathbf{k}}) + \varepsilon_{\mathbf{k}} \tag{2}$$

represents the relationship between the k^{th} state variables and the actual measurements. ε_k represents both the errors in the measurements and the errors in the model. Assuming

that $\mathbf{E}\{\varepsilon_k\} = 0$, (where $\mathbf{E}\{\cdot\}$ is the expectation operator), the state estimation problem can be written as:

$$\min_{\mathbf{x_k}} \mathbf{C} = \|\mathbf{z_k} - \mathbf{z_k^o}\| \tag{3}$$

However, since $\mathbf{z_k^o}$ is unobtainable, the estimate $\hat{\mathbf{z}}_\mathbf{k}$ is used instead, where:

$$\mathbf{z_k} = \hat{\mathbf{z}}_\mathbf{k} + \mathbf{r_k} = \mathbf{g}(\hat{\mathbf{x}}_\mathbf{k}) + \mathbf{r_k} \tag{4}$$

where $\mathbf{r_k}$ are the residuals. Hence, the required minimisation is:

$$\min_{\hat{\mathbf{x}}_\mathbf{k}} \mathbf{C} = \|\mathbf{z_k} - \hat{\mathbf{z}}_\mathbf{k}\| \tag{5}$$

The norm of Eqn. 3 is usually either L_1 or L_2 (first or second Euclidean).

In what follows a number of results will be obtained for the L_2 norm to provide the statistical properties of the least squares estimator.

A2.3 Newton Raphson Process

The conventional procedure for solving the estimation problem is to minimise a suitable error criteria within an iterative linearisation scheme such as the Newton-Raphson method.

At iteration $^{(p)}$, this can be written as:

$$\mathbf{x}^{(p+1)} = \mathbf{x}^{(p)} - \left(\mathbf{H_f}|_{\mathbf{x}^{(p)}}\right)^{-1} \nabla_f|_{\mathbf{x}^{(p)}} \tag{6}$$

where $\mathbf{H_f}|_{\mathbf{x}^{(p)}}$ is the Hessian matrix evaluated at $\mathbf{x}^{(p)}$. $\nabla_f|_{\mathbf{x}^{(p)}}$ is the gradient and $\mathbf{x}^{(p+1)}$ is a better solution than $\mathbf{x}^{(p)}$.

Rewriting Eqn. 6 gives:

$$-\mathbf{H_f}|_{\mathbf{x}^{(p)}} \Delta\mathbf{x} = \nabla_f|_{\mathbf{x}^{(p)}} \tag{7}$$

where $\Delta \mathbf{x} = \mathbf{x}^{(p+1)} - \mathbf{x}^{(p)}$. The Newton-Raphson procedure can now be stated as:

$$\mathbf{x}^{(p+1)} = \mathbf{x}^{(p)} + \alpha \Delta \mathbf{x} \tag{8}$$

with $\Delta \mathbf{x}$ being obtained from Eqn. 7. α can be obtained using a linear search although reasonable performance is usually achieved providing α is in the range $0.5 \leq \alpha \leq 1.0$.

The Newton-Raphson process is halted by some stopping criteria, say:

$$|\Delta \mathbf{x_i}| \leq \xi \quad \forall \, i \tag{9}$$

where ξ is small.

A2.3 Least Squares Formulation

The least squares error criterion is:

$$\min_{\hat{\mathbf{x}}_\mathbf{k}} \mathbf{f}(\hat{\mathbf{x}}_\mathbf{k}) = (\mathbf{z_k} - \mathbf{g}(\hat{\mathbf{x}}_\mathbf{k}))^T \mathbf{R}^{-1} (\mathbf{z_k} - \mathbf{g}(\hat{\mathbf{x}}_\mathbf{k})) \tag{10}$$

where R is the measurement error covariance matrix, assumed diagonal.

For the Newton-Raphson scheme:

$$\nabla_f|_{\hat{\mathbf{x}}^{(p)}} = -2\mathbf{J}(\hat{\mathbf{x}}^{(p)})^T \mathbf{R}^{-1} (\mathbf{z_k} - \mathbf{g}(\hat{\mathbf{x}}_\mathbf{k})) \tag{11}$$

and the Hessian is approximated by:

$$\mathbf{H_f}|_{\hat{\mathbf{x}}^{(p)}} = 2\mathbf{J}(\hat{\mathbf{x}}^{(p)})^T \mathbf{R}^{-1} \mathbf{J}(\hat{\mathbf{x}}^{(p)}) \tag{12}$$

where $\mathbf{J}(\hat{\mathbf{x}}^{(p)})$ is the Jacobian matrix evaluated at the p^{th} iteration.

Substituting Eqns. 11 and 12 into 7 produces the normal equation (dropping the iteration indices):

$$\mathbf{J}(\hat{\mathbf{x}})^T \mathbf{R}^{-1} \mathbf{J}(\hat{\mathbf{x}}) \Delta \hat{\mathbf{x}} = \mathbf{J}(\hat{\mathbf{x}})^T \mathbf{R}^{-1} \Delta \mathbf{z} \tag{13}$$

the solution of which minimises:

$$\min_{\Delta\hat{\mathbf{x}}} \mathbf{C}(\hat{\mathbf{x}}) = (\Delta\mathbf{z} - \mathbf{J}(\hat{\mathbf{x}})\Delta\hat{\mathbf{x}})^{\mathbf{T}}\mathbf{R}^{-1}(\Delta\mathbf{z} - \mathbf{J}(\hat{\mathbf{x}})\Delta\hat{\mathbf{x}}) \tag{14}$$

A2.5 Alternative Normal Equation Formulation

Using the Taylor expansion $\mathbf{g}(\mathbf{x})$ can be written as:

$$\mathbf{g}(\mathbf{x}) \approx \mathbf{g}(\hat{\mathbf{x}}_\mathbf{k}) + \frac{\partial \mathbf{g}(\hat{\mathbf{x}})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\hat{\mathbf{x}}_\mathbf{k}} (\mathbf{x} - \hat{\mathbf{x}}_\mathbf{k}) + \dots \tag{15}$$

ignoring higher order terms and substituting from Eqn. 2 gives,

$$\mathbf{z}_\mathbf{k} = \hat{\mathbf{z}}_\mathbf{k} + \mathbf{J}(\hat{\mathbf{x}}_\mathbf{k})\Delta\hat{\mathbf{x}} \tag{16}$$

hence

$$\Delta\mathbf{z} = \mathbf{J}(\hat{\mathbf{x}}_\mathbf{k})\Delta\hat{\mathbf{x}} \tag{17}$$

where $\Delta\mathbf{z} = \mathbf{z}_\mathbf{k} - \hat{\mathbf{z}}_\mathbf{k}$ and $\Delta\hat{\mathbf{x}} = \mathbf{x} - \hat{\mathbf{x}}_\mathbf{k} = \hat{\mathbf{x}}^{(p+1)} - \hat{\mathbf{x}}^{(p)}$. Hence the minimisation of:

$$\min_{\Delta\hat{\mathbf{x}}} \mathbf{C}(\hat{\mathbf{x}}) = \|\Delta\mathbf{z} - \mathbf{J}(\hat{\mathbf{x}}_\mathbf{k})\Delta\hat{\mathbf{x}}\|_\mathbf{2} \tag{18}$$

leads to the same formulation as given above in A2.5.

A2.6 Residual Sensitivity Matrix

Theorem: Given:

$$\mathbf{z} = \mathbf{g}(\mathbf{x}) + \epsilon \tag{19}$$

the residual vector, defined as:

$$\mathbf{r} \triangleq \mathbf{z} - \mathbf{g}(\hat{\mathbf{x}}) \tag{20}$$

can be determined from:

$$\mathbf{r} = \mathbf{S}\epsilon \tag{21}$$

where $\mathbf{J} = \mathbf{J}(\hat{\mathbf{x}})$ and

$$\mathbf{S} = \mathbf{I} - \mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1} \tag{22}$$

and is known as the residual sensitivity matrix.

Proof: From the normal equation we have the projection from $\Delta\mathbf{z}$ to $\Delta\hat{\mathbf{x}}$, viz:

$$\Delta\hat{\mathbf{x}} = (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\Delta\mathbf{z} \tag{23}$$

hence

$$\Delta\mathbf{z} - \mathbf{J}\Delta\hat{\mathbf{x}} = \Delta\mathbf{z} - \mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\Delta\mathbf{z} \tag{24}$$

or

$$\begin{aligned} \mathbf{r}' &= (\mathbf{I} - \mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1})\Delta\mathbf{z} \\ &= \mathbf{S}\Delta\mathbf{z} \end{aligned} \tag{25}$$

In the limit, as:

$$\lim_{\Delta\mathbf{x}\to\mathbf{0}} \mathbf{g}(\hat{\mathbf{x}}) = \mathbf{g}(\mathbf{x}) \tag{26}$$

hence

$$\mathbf{r} = \mathbf{S}\epsilon$$ □

A2.7 Covariances of Errors in the State Estimates

Theorem: While

$$\begin{aligned} \mathbf{E}\{\epsilon\} &= \underline{0} \\ \mathbf{E}\{\epsilon\epsilon^{\mathbf{T}}\} &= \mathbf{R} \end{aligned} \tag{27}$$

Then the state estimate error covariances are given by:

$$\mathbf{E}\{(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})\}^{\mathbf{T}} = (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1} \tag{28}$$

Proof:

$$\begin{aligned} \Delta\hat{\mathbf{x}} &= (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}(\mathbf{J}\Delta\hat{\mathbf{x}} + \epsilon) \\ &= \Delta\hat{\mathbf{x}} + (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\epsilon \end{aligned} \tag{29}$$

hence,

$$\begin{aligned} \mathrm{E}\{\Delta\hat{\mathbf{x}}\} &= \mathrm{E}\{\Delta\hat{\mathbf{x}}\} + \mathrm{E}\{(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\} \times \mathrm{E}\{\epsilon\} \\ &= \mathrm{E}\{\Delta\hat{\mathbf{x}}\} \end{aligned} \tag{30}$$

Using,

$$\hat{\mathbf{x}}^{(p+1)} = \mathbf{x}^{(p)} + \Delta\hat{\mathbf{x}}^{(p)} \tag{31}$$

let the real state equal the previous one plus an optimal increment:

$$\mathbf{x} = \mathbf{x}^{(p)} + \Delta\mathbf{x}^{*} \tag{32}$$

then the errors in the estimates can be obtained by subtracting Eqn. 32 from Eqn. 31

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}}^{(p+1)} - \mathbf{x} = \Delta\hat{\mathbf{x}}^{(p)} - \Delta\mathbf{x}^{*} \tag{33}$$

hence, from Eqn. 29,

$$\tilde{\mathbf{x}} = (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\epsilon \tag{34}$$

and, writing Eqn. 28 in the form of Eqn. 33, we have:

$$\begin{aligned} \mathrm{E}\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathbf{T}}\} &= \mathrm{E}\{(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\epsilon\epsilon^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\} \\ &= (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{R}\mathbf{R}^{-1}\mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1} \\ &= (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1} \end{aligned}$$

□

A2.8 Covariances of Residuals

Theorem: The residual covariance matrix $\mathbf{P}$ is given by:

$$\mathbf{P} = \mathbf{R} - \mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}} \tag{35}$$

Proof:

$$\begin{aligned} \mathbf{P} &\triangleq \mathrm{E}\{\mathbf{r}\mathbf{r}^{\mathbf{T}}\} \\ &= \mathrm{E}\{\mathbf{S}\epsilon\epsilon^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\} \\ &= \mathbf{S}\mathrm{E}\{\epsilon\epsilon^{\mathbf{T}}\}\mathbf{S}^{\mathbf{T}} \\ &= \mathbf{S}\mathbf{R}\mathbf{S}^{\mathbf{T}} \end{aligned}$$

Now,

$$\mathbf{S} = \mathbf{I} - \mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1} \tag{36}$$

let,

$$\mathbf{\Sigma} = (\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})^{-1} \tag{37}$$

(note that $\mathbf{\Sigma}$ is symmetric) then

$$\begin{aligned}
\mathbf{P} &= (\mathbf{I} - \mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1})\mathbf{R}(\mathbf{I} - \mathbf{R}^{-1}\mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}}) \\
&= \mathbf{R} - \mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}} - \mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}} + \mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}} \\
&= \mathbf{R} - \mathbf{2}\mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}} + \mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}} \\
&= \mathbf{R} - \mathbf{J}\mathbf{\Sigma}\mathbf{J}^{\mathbf{T}}
\end{aligned}$$

□

A2.9 Statistics of residuals

Theorem: The residuals r_i belong to the distribution $N(0, p_{ii})$, where p_{ii} are the diagonal elements of the residual covariance matrix $\mathbf{P}$.

Proof:

i) From proof A2.7, the estimates are unbiassed, therefore $\mathbf{E}\{\mathbf{r}\} = 0$

ii) From proof A2.8, the variances are given by:

$$p_{ii} = trace\{\mathbf{R} - \mathbf{J}(\mathbf{J}^{\mathbf{T}}\mathbf{R}^{-1}\mathbf{J})\mathbf{J}^{\mathbf{T}}\} \tag{38}$$

hence $r_i \in N(0, p_{ii})$ □

APPENDIX 3

HUMAN COMPUTER INTERFACE

A.3.1 Purpose

Since WASMACS is mostly intended for use in the control room, the HCI must be reliable, easy to use, consistent, powerful in its functionality and fast. It will display the information from a host of applications and from the telemetry or the data base. It also acts as the command centre allowing the operator to drive the various applications that are currently available.

A.3.2 Method

Use is made of DEC-Windows to provide all the pull down menus, dialogue boxes and mouse functions. The system is, of course, fully colour graphic. The network itself can be drawn using a wealth of draw functions with a wide choice of colours including flashing or animated. Each component can be drawn to an exact size on an accurate grid representing the real world co-ordinates, it can be automatically switched on or off depending on current zoom level and it can be automatically switched from one colour to another (e.g. steady blue to flashing red) by a change in some logical value that might be derived from the telemetry or from an applications package.

The zoom ratio is continuously (steplessly) adjustable over a 107 zoom ratio. The zoom action is mouse driven and the rate of change of zoom ratio is proportional to the mouse displacement. There is minimal lag whilst zooming so that near instant changes in zoom level are possible. Pan works in a similar way, the rate of panning is proportional to the mouse displacement and in the direction of the displacement. This allows easy traversal of long pipes to discover end points. Multiwindow operation is permitted. Using this function, several separate windows can be set up containing the same (or different) network(s) with full window independence possible. Windows can, however, be set up to follow each other so that any panning operation carried out in one will be immediately translated to the other.

Very importantly, an applications menu allows any available applications to be run. This menu is fully user customisable which permits third party software packages to be run from one common interface. Other customise options include the security menu which allows a 'super user' to define which functions are available to other users. Furthermore, the keyboard is fully user definable and a facility is provided to allow each user to individually save their current keyboard settings in a private file for future use.

Another key feature of this system is the direct interface it offers to the network relational data base. Graphical objects can be drawn together with numerical objects (known as 'variables') which instantiate the relations in the data base. These two types of object can be grouped together as a single entity in which the numerical attribute corresponds to a data base entry. In this way, anything that is currently in the data base can be displayed on the screen no matter what the original source of that data was. Hence data from third party software can be displayed quite easily on the screen, which together with the customise options described above makes the HCI a truly 'open system'.

A.3.3 Software Details

Program name: Water PACS.

(PACS :- 'Planning And Control System')

APPENDIX 4

INSTALLATION SCENARIO

A.4.1 Introduction

Prior to the completion of the project, the telemetry system must be configured so that the measurements required by the state estimator are available. The WASMACS engineer will convey these requirements to the telemetry engineers as the project progresses.

Since different telemetry systems will be at different stages of installation or commissioning at the start of a project, or in fact may be complete, this description may not be representative for any particular case.

Phase 1: Analysis

On the commencement of the project data will be transferred pertaining to the network topology, pipe dimensions, positions of network inflows, number of consumers at each node, typical demands, inflows, reservoir levels, the number and type of pumps, P.R.V.'s, typical operating pressures and any other information relevant to the network in question.

From this information it will be possible to quickly set up an approximate network model and thereby obtain a reasonable balance that is as accurate as the data so far provided.

As this work proceeds, it is anticipated that the client may need to undertake a 'C-value' survey to obtain more accurate representations of pipe roughness. The engineers concerned will use their past experience to ascertain which pipes warrant further checking. This survey will at the same time shed new light on the network topology, showing up blocked pipes or closed valves previously thought to be open. At the end of Phase 1 the water engineers and the WASMACS engineer will report their findings and decide on any new strategy for further surveys of the network.

Phase 2: Meter Placement

During Phase 2 a more detailed study of the network will be made based on the most up to date topological and dimensional information and using recently measured flows and pressures in the system.

The objectives of this phase are to produce a more accurate network model and to ascertain the measurement requirement of the WASMACS system. The meter placement algorithm described in [191] will be used to produce an optimised choice of meter locations. The chosen measurement set will render the entire network observable with sufficient redundancy to allow accurate state estimation even if measurement errors occur. The measurement requirement will take into account the measurements already available and will attempt to minimise the cost of extra instrumentation. In parallel with this activity, further surveys may be necessary (identified at the completion of Phase 1). These may be new C-value surveys, load flow surveys in localised areas or "trouble-shooting" surveys to try to pin-point the cause of any anomalies found to exist in the network.

On the successful completion of these two activities, sufficient information will be available to produce a computer model that closely simulates the real system, including the water network, telemetry and state estimator.

Phase 3: Off-line Validation

This is the final phase prior to on-line operation. The objectives of this phase are:

(i) Complete the telemetry system commissioning;

(ii) Validate the network model, the telemetry model and the state estimator;

(iii) Complete any changes necessary to configure the WASMACS to run on the target platform;

(iv) For the purposes of demand prediction, complete the logging of network demand as specified by the WASMACS engineer.

With the initial meter placement problem complete, it will be necessary to run a full simulation of the suite of programs which will be used on-line together with simulations of the system itself and the telemetry. This will show whether the measurement requirement specified actually provides an observable system with sufficient measurement redundancy to allow leak detection to take place. A sensitivity analysis can be carried out at this stage. The results of this analysis will also be used to further check and fine tune the network model. When these two activities are complete the original measurement requirement may need slight revision. It is also possible that certain measurement installations are physically difficult to realise and so the final measurement set may be different from the original proposal.

Final model validation can begin when the measurement set has been finalised wherein the estimated states of the system over many 24 hour periods will be checked against real data logged in the field. Various 'events' could be introduced into the network to check the state estimator's robustness and accuracy. The more exhaustive this period of off-line testing is, the more reliable the total system will be prior to on-line testing. With on-line testing, some unforeseen problems may arise and it is therefore necessary to have experience of as many types of event as possible during the off-line phase.

Phase 4: On-line Validation

A short period of on-line testing will be necessary to validate the telemetry to WASMACS data transfer to ensure reliable operation. It is in this final stage that the assumptions of the transducer noise signals will be tested and consequently a period of state estimator tuning may also be required. Problems such as those discussed in [224] may also be encountered.