EXTENSIONS TO THE DATA RECONCILIATION PROCEDURE

by

Mark Thomas Seager

Submitted to the University of Cape Town in fulfilment of the requirements for the degree of Master of Science in Engineering.

June 1996

ACKNOWLEDGEMENTS

I thank my supervisor Dr C.L.E. Swartz for all his assistance and guidance throughout the past two and a half years. Thanks also goes to Clifford Meyer, Julian Young, Roderick Ross and Trevor Hadley in the Process Systems Engineering group for their help and support.

I would like to thank the Foundation for Research and Development for funding this research.

I would also like to thank Tony Rund and Poovan Nader for all their help in formulating the industrial case study.

I would especially like to acknowledge the support of my parents throughout the duration of my studies, without which none of my achievements would be possible.

SYNOPSIS

Data reconciliation is a method of improving the quality of data obtained from automated measurements in chemical plants. All measuring instruments are subject to error. These measurement errors degrade the quality of the data, resulting in inconsistencies in the material and energy balance calculations. Since important decisions are based on the measurements it is essential that the most accurate data possible, be presented. Data reconciliation attempts to minimize these measurement errors by fitting all the measurements to a least-squares model, constrained by the material and energy balance equations. The resulting set of reconciled measurements do not cause any inconsistencies in the balance equations and contain minimum measurement error.

Two types of measurement error can occur; random noise and gross errors. If gross errors exist in the measurements they must be identified and removed before data reconciliation is applied to the system. The presence of gross errors invalidates the statistical basis of data reconciliation and corrupts the results obtained. Gross error detection is traditionally performed using statistical tests coupled with serial elimination search algorithms. The statistical tests are based on either the measurement adjustment performed by data reconciliation or the balance equations' residuals.

A by-product of data reconciliation, obtained with very little additional effort, is the classification of the system variables. Unmeasured variables may be classified as either observable or unobservable. An unmeasured variable is said to be unobservable if a feasible change in its value is possible without being detected by the measurement instruments. Unmeasured variables which are not unobservable are observable. Measured variables may be classified as either redundant, nonredundant or having a specified degree of redundancy. Nonredundant variables are those which upon deletion of the corresponding measurements, become unobservable. The remaining measured variables are redundant. Measured variables with a degree of redundancy equal to one, are redundant variables that retain their redundancy in the event of a failure in any one of the remaining measurement instruments.

The classification of variables into the above categories is important in the analysis of the reconciliation results, gross error detection and in the analysis and design of measurement structures. The presence of unobservable variables causes the results obtained for the unmeasured variables to be nonunique. Nonredundant measurements pass through the reconciliation procedure unadjusted. Statistical tests that are based on the reconciled adjustment cannot be applied to nonredundant measurements. Gross errors can therefore exist undetected in nonredundant measurements.

The objective of this dissertation is to identify certain extensions that can be applied to data reconciliation and gross error detection schemes. Imbedded in the data reconciliation procedure is the measurement variance matrix. The measurement variances are usually assumed as arbitrary percentages of the measured values. It was found that the reconciliation procedure can be sensitive to specific variances. The sensitivity of the reconciliation procedure to a pair of measurements of the same size and hence variance, but in different positions in a network, was investigated. It was found that the reconciliation procedure can be more sensitive to one variance than the other. Using differential sensitivities it is possible to rank the measurement variances in order of their effect on the reconciliation procedure. It is thus possible to identify which measurement variances need a high degree of accuracy and which may be roughly estimated.

The success of gross error detection schemes depends heavily on the measurement structure upon which they operate. If measurement structures contain many nonredundant measurements poor gross error detection results can be expected. Two algorithms; the Global Search Algorithm (GSA) and the Local Search Algorithm (LSA), have been developed to synthesise good measurement structures. The GSA performs an exhaustive search and may (in large problems) not reach a solution in an acceptable time. However, the GSA solution is guaranteed to be the globally optimal measurement structure constrained by variable classification specifications. The user may classify any variable as measured or unmeasured; and any measured variable as redundant, nonredundant or with a specified degree of redundancy; and any unmeasured variable as observable or unobservable. The LSA, although not guaranteed to find the global optimum, returns good local solutions in a fraction of the time needed for the GSA. It has been shown that multiple applications of the LSA, with

different starting points, may succeed in finding the global optimum.

Genetic Algorithms (GAs) were proposed as a replacement for the serial elimination algorithms in gross error detection. GAs are reputed to be successful at finding optima on irregular surfaces. GAs were used in combination with statistical tests to find simulated gross errors in a flow network. This approach, however, met with little success. The nature of the statistical test functions is extremely nonconvex and depending on the size and position of the gross error, the minimum in the objective function was found to disagree with the position of the simulated gross error. Under these conditions the efficiency of the GAs was reduced to that of a random search.

The strategies developed were applied to an industrial case study consisting of a crude preheat train at a local refinery. Data reconciliation and gross error detection were applied to the network. Gross error detection was successful in identifying two gross errors in the network. The sensitivity analysis was successful in identifying sensitive measurement variances. Variable classification identified the presence of a number of nonredundant variables. The LSA was successfully applied and an alternative measurement structure containing fewer measurements and no nonredundant variables was found. In addition, modifications to retro-fit the original measurement structure to remove the nonredundant variables, were identified.

TABLE OF CONTENTS

NOMENCLATURE

$\boldsymbol{A}$	matrix of linearized balance equations containing only measured variables
b'	estimate of the size of the gross error
b	actual size of the gross error
b_i	size of gross error in measurement i
b_o	ratio of standard deviation of the outliers to that of the random errors
b_r	rank of $\boldsymbol{B}$
$\boldsymbol{B}$	matrix of linearized balance equations containing only unmeasured variables
$\boldsymbol{B_d}$, $\boldsymbol{E}$	constraint matrices in a generalised dynamic system
c_k	k^{th} column of $\boldsymbol{C}$
$\boldsymbol{C}$	projected balance equation matrix containing only measured variables
$\boldsymbol{D_i}$	vector of flow measurements at time instant i in dynamic systems
$\boldsymbol{D^*_{i+1}}$	vector of true flows at time instant $i + 1$ in dynamic systems
e_i	unit vector containing unity in the i^{th} position and zeros elsewhere
f_d	differential equation constraints dynamic data reconciliation
f	objective function for Genetic Algorithms
F_i	fitness of individual i
F, L	sets of suspect measurements under investigation
$\boldsymbol{h_d}$	algebraic equality constraints dynamic data reconciliation
$\boldsymbol{g_d}$	inequality constraints in dynamic data reconciliation
g	function converting the objective function value to a fitness function
h_j	j^{th} column of $\boldsymbol{H}$
h_{iz}	i^{th} coefficient of the z^{th} balance equation
h_r	rank of $\boldsymbol{H}$
$\boldsymbol{H}$	(z x $(n + m)$) matrix of balance equation coefficients
$\boldsymbol{H_1}$	incidence matrix describing the position of measured variables in a network
$\boldsymbol{H_2}$	incidence matrix describing the position of unmeasured variables in a network
$\boldsymbol{H_c}$	balance equation for component c
$\boldsymbol{J}$	least squares objective function

k	counter
K	number of rows in Γ
L	number of columns in Γ
n	number of measurements
m	number of unmeasured variables
$\boldsymbol{M}$	incidence matrix for dynamic systems
$\boldsymbol{M}_r$	matrix of stoichiometric coefficients
N	number of sets of process measurements
N_s	number of selections required in SUS
p_b	probability of a gross error
p_j	multiple of standard deviation, $b_j = p_j \sigma_j$
p_r	pointer in SUS
$\boldsymbol{P}$	projection matrix
$\boldsymbol{Q}$	orthogonal basis for $\boldsymbol{B}$ from QR decomposition
$\boldsymbol{R}$	matrix of constants from QR decomposition
r	rank of $\boldsymbol{B}$
r	residuals y - x
r_b	residuals of the balance equations, $r_b = \boldsymbol{H}y$
S	Set of all initial measurements
$\boldsymbol{S}$	estimate of covariance matrix $\boldsymbol{\Sigma}$
S_r	estimate of covariance matrix $\boldsymbol{\Sigma}$ using the balance equations residuals
T_c	critical test statistic for the GLR method
T_i	GLR test statistic
T_r	GLR statistic for the restricted measurements
T_u	GLR statistic for the unrestricted measurements
u	vector of unmeasured variables
v	rank of $\boldsymbol{C}$
$\boldsymbol{V}$	variance of residuals $\boldsymbol{C\Sigma C}^T$ or $\boldsymbol{H\Sigma H}^T$
$\boldsymbol{V}_u$	variance of the residuals for the unrestricted measurements $\boldsymbol{H}_u\boldsymbol{\Sigma}_{uu}\boldsymbol{H}_u^{\mathrm{T}}$
w_i	flows in minerals processing flowsheets
$\boldsymbol{W}_i$	vector of volume measurements at time instant i in dynamic systems

$\boldsymbol{W}^*_{i+1}$ vector of true volume at time instant $i + 1$ in dynamic systems

x vector of true values for the measurements

x^* reconciled measurement values

x_i true value of measurement i

x_i^k measured assays of the k^{th} component of the i^{th} stream in minerals processing flowsheets

x_{ia} adjusted i^{th} variable

x_{ig} given i^{th} variable

X vector of measurement variance estimates

y vector of measurements

$\hat{y}$ estimate function in dynamic data reconciliation

$\hat{y}(t_j)$ value of the estimates at time t_j in a dynamic system

y_j vector containing the positions of the unmeasured variables in measurement structure synthesis

$\boldsymbol{Y}_k$ vector of unknown variables at time instant k in dynamic systems

$\boldsymbol{Y}_i^*$ vector of true values of the measurements $[\boldsymbol{W}_i^*, \boldsymbol{D}_i^*]^T$ in dynamic systems

z number of balance equations

z_c critical test statistic for the measurement test

z_i measurement test statistic for measurement i

$\boldsymbol{Z}$ vector of measurements $[\boldsymbol{W}_i, \boldsymbol{D}_i]^T$ in dynamic systems

α level of significance for the statisical tests

β modified level of significance for statistical tests

Γ matrix contain information of 'bad' measurement structures

δ parameter of noncentrality of the chi-square distribution

ϵ_{Di} error vector for $\boldsymbol{D}_i$ in dynamic systems

ϵ_{Wi} error vector for $\boldsymbol{W}_i$ in dynamic systems

λ vector of Lagrange multipliers

σ_i variance of the i^{th} measurement

Σ variance-covariance matrix of the measurements

Σ_D covariance matrix for $\boldsymbol{D}$ in dynamic systems

Σ_r variance of the residuals $y - x^*$

Σ_W covariance matrix for W in dynamic systems

τ_{GT} global test statistic

τ vector describing changing process conditions

Φ_1 simultaneous data reconciliation and gross error detection objective function

Φ_k matrix of balance constraints at time instant k in dynamic systems

$\psi(x)$ nonlinear constraints equations

1. INTRODUCTION

The increase in on-site computing and the need for more information about plant operations has been accompanied by a considerable increase in the amount of plant data available. However, there is often a lack of confidence in the data quality due to inconsistencies with known heat and mass balances. Data reconciliation procedures can be used to produce a reliable set of data -- that is one that should approximate real (true) values. Yet, the real values are not known and one can only judge data reliability on the basis of how realistic and consistent they are. When a set of data is realistic, all the data points have a believable sign and value.

Data reconciliation involves comparing actual instrument readings and analyses with calculated values from a model (Stephenson and Shewchuk, 1986). Process measurements are subject to two types of error: random errors, that are assumed by most researchers to be independent and normally distributed with zero mean and known variance. These are caused by normal process fluctuation, minor sampling or meter bias and random variation in the operation of an instrument. On the other hand, gross errors are caused by nonrandom events such as measurement biases and miscalibrated or malfunctioning instruments. These gross errors are characteristically so large that the measurement bears little relation to the underlying physical quantity. The measurements are not consistent with the model equations because they are subject to random and gross errors.

Data reconciliation and gross error detection schemes provide a methodology which ensures that the heat and material balance equations are satisfied. The difference between the measured and calculated values is minimized, and gross errors are detected. The improved accuracy in the data can be used for better process evaluation and control. Without consistency, or a true awareness of the accuracy of the data, it is impossible to give an accurate figure for a product cost or its tolerance. For example, a $\pm$ 2.5 % tolerance on the kerosine flow from a 60 000 bpd crude unit, with a \$2/bbl differential between naptha and

kerosine would translate to a $250 000 per year uncertainty on the value of crude (Lawrence, 1989).

The flow of information in a typical on-line monitoring system is shown in Fig.1.1 (Kramer and Mah, 1993). Online measurements and control commands are the inputs. Sensor pre-checks can be performed on the raw data to look for 'dead' sensors, out-of-range values and other large errors. After removing the obvious sensor failures, an in-depth fault (gross error) detection step can be carried out. If no faults are detected, the data can be rectified (reconciled) to improve the state information. However, if a fault is discovered at the detection phase, reconciliation is not carried out immediately and diagnosis is activated.

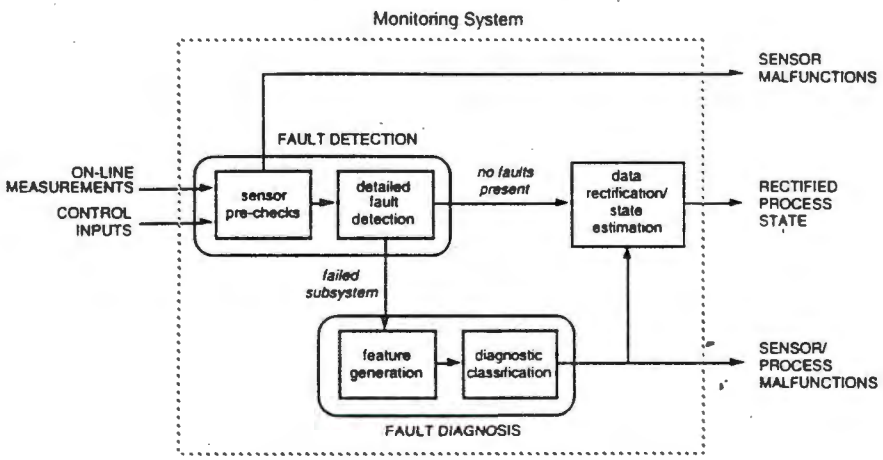


Figure 1.1 Flow of information in on-line monitoring, reproduced from Kramer and Mah, (1993)

The subject of data reconciliation and gross error detection has received much attention in the last thirty years. Algorithms for steady state and dynamic data reconciliation, based on linear and nonlinear model equations, have been presented. Gross error detection algorithms have also being developed, based on statistical tests coupled with search algorithms, although as with any procedure that involves a statistical test of significance, the methods have built-in probabilities of making mistakes of both Type I (erroneously identifying a gross error where

none exists) and Type II (failing to find an existing gross error) (Searth *et al.*, 1989).

1.1. Project Scope

Previous research surrounding data reconciliation and gross error detection has involved developing faster and more reliable calculation techniques, improved test statistics and gross error search algorithms. Little attention has been paid to the measurement covariance matrix, which forms an integral part of the solution. The measurement covariance matrix is normally assumed to be equal to a percentage of the corresponding measurement, or calculated through standard statistical methods. Although Rollins and Davis (1993) and Keller *et al.* (1992) have proposed alternative methods for calculating this matrix, an evaluation of the reconciliation procedure's sensitivity to the covariance has not been investigated.

The performance of data reconciliation and gross error detection methods depends on the system upon which they operate, especially if unmeasured variables are present, which is the norm for industrial measurement structures. For example: if a system contains nonredundant measurements and / or unobservable variables, then it is impossible to obtain a unique set of reconciled values and perform statistical tests on the nonredundant measurements. Properties such as measurement redundancy and observability play an important part in the success of gross error detection algorithms. The development of a scheme that, given a proposed network, can find the optimum positioning of measurements satisfying constraints of measurement redundancy and variable observability, will allow 'good' measurement structures to be synthesized. This would provide 'good' measurement structures upon which successful data reconciliation and gross error detection can be performed.

Serial elimination schemes that work hand in hand with statistical tests to identify gross errors have been proposed in the literature. However, the emergence of new optimization techniques, in particular Genetic Algorithms, may represent better vehicles for gross error detection.

The objectives of this project are therefore:

- To review data reconciliation and gross error detection strategies presented in the literature.
- To evaluate the sensitivity of the reconciliation procedure to the measurement variances.
- To investigate the development of an algorithm for measurement structure synthesis that would provide 'good' measurement structures upon which data reconciliation and gross error detection can readily be performed.
- To investigate the use of Genetic Algorithms as a replacement for the serial elimination search algorithms in gross error detection.
- To apply the techniques developed in an industrial case study.

The dissertation is organized as follows:

Chapter 2 contains a review of the different data reconciliation and gross error detection methods found in the literature.

Chapter 3 investigates the sensitivity of the reconciliation procedure to the measurement covariance matrix.

Chapter 4 contains the development of measurement structure synthesis algorithms. The Global Search Algorithm and the Local Search Algorithm are presented. The effectiveness of these algorithms is tested using two literature examples.

Chapter 5 applies Genetic Algorithms to gross error detection as a replacement for the serial elimination techniques. Two different test statistics are used in the Genetic Algorithm to search for gross errors, namely the global test statistic and the recursive method for gross

error detection.

Chapter 6 contains the results from an industrial case study in which data reconciliation, gross error detection, a sensitivity analysis, variable classification and measurement structure synthesis were performed on a nonlinear system.

2. DATA RECONCILIATION AND GROSS ERROR DETECTION - A REVIEW

This chapter presents the work on data reconciliation and related topics presented in the literature. It contains a description of steady state and dynamic reconciliation, variable classification, gross error detection, some related developments and a look into the minerals processing approach to data reconciliation.

2.1. Steady State Data Reconciliation

Adjustment of process data from continuous steady-state operations may indicate inconsistencies in the measurements. It is common practice in manual process calculations to adjust the measured data by small amounts so that discrepancies in the heat and material balances are eliminated. This adjustment can be done intelligently by taking the knowledge that one may have regarding the accuracy of certain measurements into consideration.

2.1.1. Initial development

The first application of data smoothing techniques, which later became known as data reconciliation, was presented by Kuehn and Davidson (1961). These researchers considered the problem of the adjustment of flow and enthalpy measurements to satisfy a set of simultaneous balance equations. The adjustment of measurements is performed judiciously so that measurements more prone to error will receive a larger adjustment. This is implemented once a value of measurement error variance has been assigned, as a weighting factor, to each measurement.

The data reconciliation problem was formulated as follows (Kuehn and Davidson, 1961):

$$\text{Min}_x \sum_{i=1}^{n} \frac{1}{\sigma_i^2}(y_i - x_i)^2$$

Subject to

$$\sum_{i=1}^{n} h_{iz} x_i = 0 \quad \forall z \qquad 2.1$$

where x_i = the i^{th} reconciled measurement

y_i = the i^{th} measured value

σ_i = the error variance of the i^{th} measurement

h_{iz} = the coefficient of the z^{th} material (or heat) balance equation.

The process flowsheet is represented as a network diagram in which the nodes represent the process units connected by a number of arcs which in turn represent the process streams. When only individual flow rates are considered, the resulting balance equations are linear in the variables and correspond to the incidence matrix of the process network. In this case h_{iz}=1 when variable i enters node z and conversely h_{iz}=-1 if variable i leaves node z.

Example

Consider an example flow network consisting of four nodes and seven streams, as shown in Fig. 2.1. The flowrate of each stream (A - G) is measured and x_A represents the flowrate measurement of stream A.

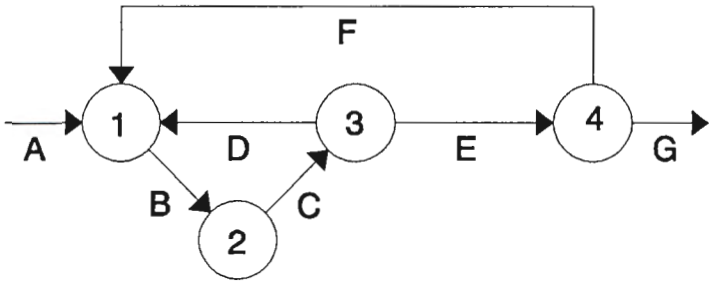


Figure 2.1 Example flow network

Balance equations are performed around each node:

NODE 1: $x_A + x_F + x_D - x_B = 0$

NODE 2: $x_B - x_C = 0$

NODE 3: $x_C - x_D - x_E = 0$

NODE 4: $x_E - x_F - x_G = 0$

giving the incidence matrix $\boldsymbol{H}$.

$$H = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & -1 \end{bmatrix} \qquad 2.2$$

$$x = \begin{bmatrix} x_A & x_B & x_C & x_D & x_E & x_F & x_G \end{bmatrix}^T$$

The solution is obtained by using Lagrange multipliers. In matrix notation the new problem, involving Lagrange multipliers (λ), can be rewritten as follows:

$$\text{Min}_x \quad (y - x)^T \Sigma^{-1} (y - x) + \lambda^T H x \qquad 2.3$$

Equation (2.3) is then differentiated with respect to x and λ and the stationary points are found by setting the resulting differentials equal to zero and solving simultaneously for x and λ:

$$\begin{bmatrix} \Sigma^{-1} & H^T \\ H & 0 \end{bmatrix} \begin{bmatrix} x \\ \lambda \end{bmatrix} = \begin{bmatrix} d \\ 0 \end{bmatrix} \qquad 2.4$$

where Σ = a diagonal matrix with elements (σ_i^2 / 2)

H = the matrix (h_{zi}) of mass and energy balances

x = the vector (x_i) of adjusted measurements

λ = the vector (λ_z) of Lagrange multipliers

d = the vector ($2y$ / σ_i^2) of transformed measurements

Rather than solving Eq. (2.4) by the direct approach, requiring the inversion of the left most matrix in Eq. (2.4), Kuehn and Davidson (1961) recommend the following solution as being more computationally efficient:

$$\lambda = (H\Sigma H^T)^{-1} Hy$$
$$x^* = y - \Sigma H^T \lambda \qquad 2.5$$

The above formulation applies only to linear balance equations such as material balances. Nonlinear balances occur with the inclusion of heat balances, or when more than one species is present where concentrations together with the total stream flow rate are measured.

Problems involving nonlinear balance equations can be solved by applying the Newton-Raphson method for finding the roots of simultaneous equations. Following the method outlined in Kuehn and Davidson (1961), one replaces the solution of Eq. (2.4) with the following:

$$\begin{bmatrix} \Sigma^{-1} & H^T \\ H & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \lambda \end{bmatrix} = - \begin{bmatrix} 0 \\ r_b \end{bmatrix}$$

where

$$\Delta x = x^i - x^{i-1} \qquad 2.6$$

$$r_b = Hy$$

The solution obtained through Eq. (2.6) is updated at each iteration, where x^0 is the initial trial solution which would generally be the measurement values, y. From Eq. (2.6) one is led, in place of Eq. (2.5) to:

$$x^{i+1} = x^i - \Sigma H^T (H \Sigma H^T)^{-1} H x^i \qquad 2.7$$

The methods developed by Kuehn and Davidson (1961) form the basis of data reconciliation and is the theory upon which most researchers in this field base their work (Britt and Luecke, 1971; Mah *et al.*, 1975; Crowe *et al.*, 1983; Swartz, 1989).

2.1.2. Data reconciliation with measured and unmeasured variables

Another approach was presented by Britt and Luecke (1971), where data reconciliation is performed on a system containing measured and unmeasured variables with nonlinear balance equations.

Britt and Luecke (1971) formulate the following objective function where x and u represent the measured and unmeasured variables respectively.

$$\text{Min} \quad \frac{1}{2} (y - x)^T \Sigma^{-1} (y - x)$$

Subject to

$$H(x,u) = 0 \qquad 2.8$$

The solution is obtained by joining the balance equations to the objective function by a vector of Lagrangian multipliers.

$$\text{Min}_x \ \frac{1}{2}(y - x)^T\Sigma^{-1}(y - x) + \lambda^T H(x,u) \qquad 2.9$$

Performing a Taylor series expansion about the point (x_i,u_i), where (x_i,u_i) represent the latest estimate of (x,u) in an iterative process, and dropping higher order terms, a partially linearized objective function is obtained:

$$\text{Min}_x \ \frac{1}{2}(y - x)^T\Sigma^{-1}(y - x) + \lambda^T \left[H(x_i,u_i) + A(x - x_i) + B(u - u_i)\right] \qquad 2.10$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ represent the values of the Jacobians of the balance equations $\boldsymbol{H}$ with respect to the measured and unmeasured variables, evaluated at the previous iterate (x_i,u_i). By differentiating the objective function, Eq. (2.10), with respect to x, u and λ and setting these derivatives to zero, simultaneous equations are set up. Upon rearrangement the following solutions are obtained.

$$\begin{aligned} u - u_i &= -(B^T[A\Sigma A^T]^{-1}B)^{-1}B^T[A\Sigma A^T]^{-1}\{H(x_i,u_i) + A(y - x_i)\} \\ x &= y - \Sigma A^T[A\Sigma A^T]^{-1}\{H(x_i,u_i) + A(x - x_i) + B(u - u_i)\} \end{aligned} \qquad 2.11$$

The (x,u) calculated from Eq. (2.11) is a stationary point of Eq. (2.10). Since Eq. (2.10) is only an approximation to the true objective function Eq. (2.9), (x_{i+1},u_{i+1}) is set equal to (x,u), the constraint relinearized, and a new estimate is obtained. The iterations are continued until convergence criteria are met for the measured and unmeasured variables.

2.1.3 Graphical solution techniques

Vaclavek (1969) addressed the question of how to eliminate unmeasured variables from the reconciliation optimization problem. He presented the reduced balance scheme which combines units connected by an unmeasured stream and deletes units which have an unmeasured feed or product stream. In the reduced balance scheme one is limited to the case where a stream either has all of its species flows measured, or none.

A similar approach was presented by Mah *et al.* (1976) to manage the presence of unmeasured variables. Their formulation of the objective function is similar to the above approaches, but the balance equations are formulated as two incidence matrices: $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$ corresponding to the measured and unmeasured streams respectively.

$$\begin{aligned} &\text{Min}_x\ (y - x)^T \Sigma^{-1} (y - x) \\ &\text{Subject to} \\ &\quad H_1 x + H_2 u = 0 \end{aligned} \qquad 2.12$$

These researchers developed their data reconciliation theory based on the following two graph-theoretic results.

1) Reconciliation with missing measurements can be decomposed into two disjointed problems: reconciliation on a graph which is formed by pairwise aggregation of the nodes linked by arcs of unmeasured flows, and the estimation of unmeasured flows in the tree arcs of the process graph.
2) Missing flow measurements can be determined uniquely if and only if the unmeasured arcs form an acyclic graph (i.e. trees).

The first leads to a reduction of the dimension of the computational problem. When two arcs are aggregated, the arcs external to the two nodes are preserved, but all internal links (arcs) between them are removed from consideration. The second result pinpoints the unmeasured streams whose flow rates can and cannot be uniquely determined.

The solution to the data reconciliation problem is set up by forming the following matrices:

1) A_1 an incidence matrix, which represents a graph generated from the process graph (H) by node aggregation such that all the streams of A_1 are measured.
2) B_1 represents the subgraph of unmeasured arcs (variables), where the nodes spanned by this subgraph may form a part of the aggregated nodes of A_1, but are not individually represented in A_1
3) B_{12} delineates the adjacency of the measured streams with the nodes of the unmeasured subgraphs, and is constructed from the measured streams and the nodes linked by unmeasured streams.

Once these matrices have been set up, the following solutions for the measured and unmeasured variables are possible.

$$x^* = y - \Sigma A_1^T [A_1 \Sigma A_1^T]^{-1} A_1 y$$
$$u = B_1^{-1} \{B_{12} \Sigma A_1^T [A_1 \Sigma A_1^T]^{-1} A_1 - B_{12}\} y \qquad 2.13$$

This graphical approach eliminates the unmeasured variables from the constraints of the objective function. The unmeasured variables are then calculated after the reconciled estimates of the measurements have been obtained.

The computational effort has been minimized because a reduced set of balances has been obtained, such that the reduced set involves no unmeasured variables. Algebraic expressions that accomplish the same effect have been presented in the literature (Crowe *et al.*, 1983; Swartz, 1989). These methods describe the construction of a projection matrix which effectively blanks out the unmeasured variables, producing a reduced set of balances.

2.1.4. Projection matrices to remove unmeasured variables from the objective function

The approach by Crowe *et al.* (1983) considers linear balance equations, where H is the matrix of material balance equations equivalent to the incidence matrix of the process network. As in the approach by Mah *et al.* (1976), the balance equations are formulated as

two incidence matrices for the measured and unmeasured variables: $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$ respectively. A matrix $\boldsymbol{P}$ is constructed, whose columns span the null space of $\boldsymbol{H_2}^T$:

$$H_2^T P = 0 \tag{2.14}$$

The balance equations are then premultiplied by $\boldsymbol{P^T}$, effectively removing the unmeasured variables.

$$P^T(H_1 x + H_2 u) = P^T H_1 x = Cx = 0 \tag{2.15}$$

The standard objective function is then minimized subject to the new balance equations in Eq. (2.15) and is solved using the approach of Kuehn and Davidson (1961), see Eq. (2.5), with $\boldsymbol{H}$ replaced by $\boldsymbol{C}$. Once the reconciled measurements have been calculated, the unmeasured variables can be obtained from the solution for u from the original balance equations.

The projection matrix $\boldsymbol{P}$ is calculated by reducing $\boldsymbol{H_2}$ to obtain $\boldsymbol{H_{2r}}$ with linearly independent columns. Partition $\boldsymbol{H_{2r}}$ so that

$$H_{2r} = \begin{bmatrix} H_{2a} \\ H_{2b} \end{bmatrix} \tag{2.16}$$

with $\boldsymbol{H_{2a}}$ square and nonsingular. Then a matrix $\boldsymbol{P}$ is given by

$$P^T = [-H_{2b}H_{2a}^{-1} \mid I] \tag{2.17}$$

The projection method is an algebraic equivalent of the graphical approach by Mah *et al.* (1976) and the reduced balance scheme of Vaclavek (1969). This method was later extended to include bilinear balance equations (Crowe, 1985), where some concentrations are measured in streams where the total flow rate is unmeasured.

The method presented by Swartz (1989) extends the theory of a projection matrix into the realm of nonlinear balance equations. This approach uses Gauss-Newton iteration coupled with orthogonal decomposition to eliminate the m vector of unmeasured variables from the balance equations. The data reconciliation problem may be stated as:

$$\begin{aligned} &\text{Min}_{x,u}\ (y - x)^T \Sigma^{-1} (y - x) \\ &\text{Subject to} \\ &\qquad H(x,u) = 0 \end{aligned} \tag{2.18}$$

Since the model equations $\boldsymbol{H}$ can consist of z linear and / or nonlinear equations, the problem is solved by using a strategy that successively linearizes the balance equations and is similar to the approach of Britt and Luecke (1971).

The linearized constraint equations may be separated into two parts corresponding to the measured and unmeasured variables. Linearization of the balance equations around a point (x^k, u^k) gives

$$H(x^k,u^k) + \left.\frac{\partial H}{\partial x}\right|_{(x^k,u^k)} (x - x^k) + \left.\frac{\partial H}{\partial u}\right|_{(x^k,u^k)} (u - u^k) = 0 \tag{2.19}$$

which may be written as:

$$Ax + Bu = c$$

with

$$\begin{aligned} A &= \left.\frac{\partial H}{\partial x}\right|_{(x^k,u^k)} \\ B &= \left.\frac{\partial H}{\partial u}\right|_{(x^k,u^k)} \\ c &= Ax^k + Bu^k - H(x^k,u^k) \end{aligned} \tag{2.20}$$

where (x^k,u^k) is the solution at the previous iterate.

In general, there will be a subset of u that cannot be uniquely determined, in which case u needs to be eliminated from the constraint equations in order to obtain a solution. The decoupling of the unmeasured variables from the linearized constraint equations may be conveniently achieved by QR decomposition of $\boldsymbol{B}$ (Swartz, 1989).

As an introduction to the subject of QR decomposition, some properties of orthogonal vectors and matrices will be introduced (Watkins, 1991).

Orthogonality: Two or more vectors that are perpendicular are said to be orthogonal, i.e. x and y are orthogonal if $x^{\mathrm{T}}y = 0$.

Orthogonal Matrix: A matrix $\boldsymbol{Q} \in \mathbb{R}^{z \times z}$, is said to be orthogonal if $\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{Q} = I$ (Watkins, 1991).

The QR decomposition is presented with the help of the following theorem (Watkins, 1991):

Theorem: Let $\boldsymbol{B} \in \mathbb{R}^{z \times m}$ with rank $(\boldsymbol{B}) = r > 0$. Then there exist matrices $\boldsymbol{B}^*$, $\boldsymbol{Q}$ and $\boldsymbol{R}$, such that $\boldsymbol{B}^* = \boldsymbol{QR}$ where $\boldsymbol{B}^*$ is obtained by permuting the columns of $\boldsymbol{B}$, $\boldsymbol{Q} \in \mathbb{R}^{z \times z}$ is orthogonal, and

$$R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{z \times m} \qquad 2.21$$

where $\boldsymbol{R}_{11} \in \mathbb{R}^{r \times r}$ is nonsingular and upper triangular.

Now let the components of u be ordered such that the first r columns of $\boldsymbol{B}$ are linearly independent. Then the matrix $\boldsymbol{B}$ can be rewritten as:

$$B = Q \begin{bmatrix} R \\ 0 \end{bmatrix}$$
$$= [Q_1 | Q_2] \left[\begin{array}{c|c} R_{11} & R_{12} \\ \hline 0 & 0 \end{array}\right] \qquad 2.22$$

where $\boldsymbol{Q}$ is an orthogonal matrix and $\boldsymbol{R}$ is an upper triangular matrix of constants. The column vectors of $\boldsymbol{B}$ are a linear combination of the orthogonal column vectors of $\boldsymbol{Q}$. $\boldsymbol{Q}$ is further partitioned into $\boldsymbol{Q_1}$ and $\boldsymbol{Q_2}$ to correspond with the partitioning of $\boldsymbol{R}$ into $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$.

The QR decomposition can now be used to decouple the unmeasured variables from the optimization problem. By premultiplying the modified constraint equation, Eq. (2.20), by $\mathbf{Q}^T$ ($= \boldsymbol{Q}^{-1}$) the decoupling is achieved.

$$Q^T(Ax + Bu) = Q^T c$$
$$\begin{bmatrix} Q_1^T A & R_{11} & R_{12} \\ Q_2^T A & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ u_r \\ u_{m-r} \end{bmatrix} = \begin{bmatrix} Q_1{}^T c \\ Q_2{}^T c \end{bmatrix} \qquad 2.23$$

The vector of unmeasured variables is subdivided into u_r , representing the unmeasured variables corresponding to the first r independent columns of $\boldsymbol{B}$, and u_{m-r} , the remaining unmeasured variables which correspond to the linearly dependent columns of $\boldsymbol{B}$. A more formal classification of the unmeasured variables will be dealt with in subsection 2.3. Expanding the matrix form of Eq. (2.23) reveals:

$$Q_1^T Ax + R_{11}u_r + R_{12}u_{m-r} = Q_1^T c$$
$$Q_2^T Ax = Q_2^T c \qquad 2.24$$

Equations that have been obtained from the above expansion allow the calculation of the unmeasured variables and the solution of the optimization problem in terms of the measured

variables only. The first r equations allow u_r to be written explicitly in terms of the other variables.

$$u_r = R_{11}^{-1}(Q_1^T c - Q_1^T A x - R_{12} u_{m-r}) \tag{2.25}$$

Since u appears neither in the remaining equations of Eq. (2.24) nor in the objective function of Eq. (2.26), the decoupling of the problem is complete. The problem can now be restated as follows:

$$\begin{aligned} &\text{Min}_x \quad (y - x)^T \Sigma^{-1} (y - x) \\ &\text{subject to} \quad Cx = d \end{aligned} \tag{2.26}$$

where: $C = Q_2^T A$
$d = Q_2^T c$

From Eq. (2.26) it can been seen that the unmeasured variables have been completely removed from the optimization problem. The reconciliation problem may now be solved in terms of the measured variables, and Eq. (2.25) is used to solve for the unmeasured variables, u_r. The components of u_{m-r} may be arbitrary, and are therefore nonunique. This nonuniqueness of u is related to the system observability (to be discussed later). The optimization problem described by Eq. (2.26) can be treated as a restricted least squares problem.

The problem is classified as being restricted because of the restrictions imposed by the balance equations. The solution of Eq. (2.26) is given by Swartz (1989):

$$x^* = y - \Sigma C^T \left[C \Sigma C^T \right]^{-1} (Cy - d) \tag{2.27}$$

A new estimate for the measured values can now be calculated. The unmeasured variables, u_r, can be calculated using the set of values u_{m-r} and the new estimates for the measured variables in Eq. (2.25).

Starting with estimates of the unmeasured variables and measurement data, the following iterative scheme is followed:

1) Set $k=0$ and x_k = initial measurements.
2) Linearize the constraint (model) equations.
3) QR decompose $\boldsymbol{B}_k$.
4) Decouple the unmeasured variables to yield a restricted least squares optimisation problem.
5) Solve the restricted least squares optimisation problem for x_{k+1}
6) Evaluate the unmeasured variables u_{k+1} from solutions obtained in 5).
7) Check convergence of x_{k+1} and u_{k+1} ; if converged then stop.
8) Set $k=k+1$ and iterate from 2) with the new values for x and u.

2.1.5. Inclusion of mathematical programming to treat bounds on variables and strongly nonlinear models

Previous data reconciliation techniques were developed using linear (or linearized) model equations; however many chemical engineering processes operate in strongly nonlinear regions. For this reason Liebman and Edgar (1988) suggested nonlinear programming as an alternative method for data reconciliation. They found that the data estimates obtained from a nonlinear reconciliation procedure are strongly dependent on the system under investigation and the magnitude of the measurement noise.

The nonlinear data reconciliation problem can be formulated as follows:

$$\text{Min } (y - x)^T \Sigma^{-1} (y - x)$$

Subject to

$$\begin{aligned} Ax &= b \\ \psi(x) &= 0 \\ l < x &< u \end{aligned} \qquad 2.28$$

Where $\boldsymbol{A}$ represents the linear constraints, $\psi(x)$ are the nonlinear constraints and u and l are upper and lower bounds for x. Liebman and Edgar (1988) suggest that this formulation be

solved using an established nonlinear programming (NLP) algorithm: for example, the generalized reduced gradient algorithm (Lasdon and Waren, 1986). This algorithm has the following advantages: it explicitly handles nonlinear constraints, bounds on the variables and allows a general nonlinear objective function. In their study only the case where all the variables are measured was considered. However, the main disadvantage is the relatively large amount of computational power required.

Standard data reconciliation techniques ensure that estimates of the process variables satisfy the material and energy balances in either their linear or nonlinear form. Additional constraints, as for example non-negativity restrictions on the flow rates or known upper and lower limits on the process variables, are not taken into account. As a result thereof, the reconciliation procedure can give rise to spurious results such as negative flowrates (Tamhane and Mah, 1985). Traditionally the arguments for not imposing inequality constraints were: 1) the solution can no longer be obtained analytically, and 2) statistical analysis of the measurement residuals for the purpose gross error detection is difficult (Narasimhan and Harikumar, 1993a).

An algorithm for the bounded data reconciliation problem is presented in the literature by Narasimhan and Harikumar (1993a). A quadratic programming algorithm is used to obtain estimates that satisfy the bounds. The simplest model of a steady state process with all variables measured is considered.

$$\begin{aligned} &\text{Min}_x \quad (y - x)^T \Sigma^{-1} (y - x) \\ &\text{Subject to} \\ &\qquad Hx = 0 \\ &\qquad x + w = u \\ &\qquad x - v = l \\ &\qquad w_i, v_i \geq 0 \quad i = 1,...,m \end{aligned} \tag{2.29}$$

This is the standard form of a Quadratic Programming (QP) problem with slack variables (w,v) and bounds (u,l) on the variables x.

The form in Eq. (2.29) has $3m$ variables and $z + 2m$ constraints. Narasimhan and Harikumar propose a smaller QP with $2m$ variables and $z + m$ constraints by formulating the bounded data reconciliation problem as follows:

$$\begin{aligned} &\text{Min}_w \quad w^T\Sigma^{-1}w + 2c^Tw \\ &\text{Subject to} \\ &\qquad Hw = b_1 \\ &\qquad v - w = b_2 \\ &\qquad w,v \geq 0 \end{aligned} \tag{2.30}$$

where $b_1 = Hu$

$b_2 = u - l$

$c = \Sigma^{-1}(y - u)$

In the above formulation the bounds have been treated explicitly as additional constraints. Several methods are available for solving QP problems. Narasimhan and Harikumar, (1993a) chose Wolfe's algorithm modified by Dantzig (1963), because it directly returns those variables that are at their bounds in the optimal solution. This property is useful in the statistical analysis of the estimates, which will be described in a later section.

2.2. Dynamic Data Reconciliation

Keuhn and Davidson (1961) pioneered the work on data reconciliation for steady state chemical engineering processes. For dynamic process data, Kalman filtering was introduced (Gleb, 1974) to recursively smooth measurement data and estimate parameters. Both these techniques were developed for linear systems and weighted least squares objective functions.

Modifications to Kalman filtering, to handle nonlinear systems have been developed. These modifications typically involve replacing the nonlinear equations with first order Taylor series approximations. Jang *et al.* (1986) compared the extended Kalman filter to nonlinear parameter estimation and concluded that nonlinear programming was superior in terms of

response to changes in parameters and robustness in the presence of modelling errors and strong nonlinearities. In addition, Kalman filtering approaches are restricted to data reconciliation problems for which a weighted least squares objective function is appropriate and do not support the inclusion of variable bounds or inequality constraints.

Almasy (1990) presented a method for dynamic data reconciliation which is based on using linear balance equations to reconcile the measured states. Other modelling equations are neglected due to claims that dynamic filtering cannot be performed sufficiently quickly, unless the model is linear.

Three methods for dynamic data reconciliation will be introduced in this section; the nonlinear dynamic data reconciliation approach (Liebman *et al.*, 1992); the recursive dynamic data reconciliation technique (Darouach and Zasadzinski, 1991); and the unbiased estimation in dynamic data reconciliation (Rollins and Devanathan, 1993).

2.2.1. Nonlinear dynamic data reconciliation

The nonlinear dynamic data reconciliation (NDDR) approach of Liebman *et al.* (1992) can efficiently perform computations for both linear and nonlinear models.

The general NDDR formulation can be presented as follows:

$$\begin{aligned} &\text{Min}_{\hat{y}(t)}\ \Phi[y,\hat{y}(t);\sigma] \\ &\text{Subject to} \\ &f_d\left[\frac{d\hat{y}(t)}{dt},\hat{y}(t)\right] = 0 \\ &h_d[\hat{y}(t)] = 0 \\ &g_d[\hat{y}(t)] \leq 0 \end{aligned} \qquad 2.31$$

where $\hat{y}(t)$ = estimate functions

y = discrete measurements

σ = measurement noise standard deviations

f_d = differential equation constraints

h_d = algebraic equality constraints

g_d= inequality constraints including upper and lower bounds

For most applications the objective function is simply a weighted least squares

$$\Phi[y,\hat{y}(t);\sigma] = \sum_{j=0}^{c} \frac{1}{2}[\hat{y}(t_j) - y_j]^T \Sigma^{-1}[\hat{y}(t_j) - y_j] \qquad 2.32$$

where $\hat{y}(t_j)$ = the true value of the estimates at time t_j

y_j = the value of the measurements at time t_j

Σ = the measurement error covariance matrix

Given a perfect model, an ideal data reconciliation scheme would use all the information from startup until the present time. This would lead to an optimization problem of an ever-increasing dimension. Liebman *et al.* (1992) proposed a moving time window, $n\Delta t$ to reduce the optimization problem to one of manageable dimensions. An additional advantage to the NDDR solution strategy is that the length of the time window, n, provides a means for tuning the performance of the data reconciliation scheme. In general the length of this history

horizon is the only tuning parameter needed to adequately tune the performance of the NDDR algorithm (Liebman *et al.*, 1992).

The NDDR algorithm:

1) Obtain process measurements.

2) Optimise Φ for all $\hat{y}(t)$ over $t_c - n\Delta t \leq t \leq t_c$.

3) Save only one $\hat{y}(t)$:

- save $\hat{y}$ at time t_c for on-line control purposes
- save $\hat{y}$ at intermediate time t_j for off-line data smoothing calculations.

4) Repeat at next time step $t_c + 1$.

The discretization of $\hat{y}(t)$ in step 2 can be accomplished using orthogonal collocation on finite elements (Villadsen and Michelsen, 1978) or by numerical integration techniques; such as Runga Kutta methods (Figueroa and Romagnoli, 1994). The NDDR approach has been successfully applied to a nonlinear dynamic data reconciliation on a simulated adiabatic continuous stirred tank reactor with a first order exothermic reaction (Figueroa and Romagnoli 1994).

2.2.2. Recursive dynamic data reconciliation

Darouach and Zasadzinski (1991) propose an on-line estimation algorithm for systems of dynamic material balance equations. The model considered is linear and deterministic with all the variables measured and can be written in discrete difference form as follows:

$$EY_{k+1} = B_d Y_k \tag{2.33}$$

where Y_k is the vector of all unknown variables at time instant k and E and B_d are constraint matrices in a generalised dynamic system. The model contains more variables than constraints and therefore cannot be written in a standard state equation form. The model is called singular because E is singular and therefore the standard Kalman filter cannot be applied to estimate Y_k. A recursive optimal solution in a weighted least squares sense is

proposed to estimate Y_k.

Consider a process network formed by z nodes and n streams with the following discrete material balance equations.

$$W_{i+1}^{*} = W_i^{*} + MD_{i+1}^{*} \tag{2.34}$$

where D^{*}_{i+1} = the true vector of flowrates (n x 1) at time instant $i+1$

W^{*}_i = the true vector of volumes (z x 1) at time instant i

M = the (z x n) incidence matrix of full row rank

The vector of volumes W^{*}_i is defined as the true and unknown total mass in the z nodes at time instant i.

The measurements are modelled as

$$\begin{aligned} D_i &= D_i^{*} + \epsilon_{Di} \\ W_i &= W_i^{*} + \epsilon_{Wi} \end{aligned} \tag{2.35}$$

where ϵ_{Di} is a (n x 1) vector of normally distributed random measurement noise with zero mean and known covariance $\Sigma_D > 0$, and ϵ_{Wi} is a (z x 1) vector of normally distributed random measurement noise with zero mean and known covariance $\Sigma_W > 0$. The material balance equations can now be written as:

$$-EY_{i+1}^{*} + B_d Y_i^{*} = 0$$

where

$$Y_i^{*} = \begin{bmatrix} W_i^{*} \\ D_i^{*} \end{bmatrix}, \quad E = [I \,|\, -M], \quad B_d = [I \,|\, 0] \tag{2.36}$$

The model for the measurements can also be rewritten as:

$$Z_i = Y_i^* + \epsilon_i$$

Where

$$Z_i = \begin{bmatrix} W_i \\ D_i \end{bmatrix}, \quad \epsilon_i = \begin{bmatrix} \epsilon_{Wi} \\ \epsilon_{Di} \end{bmatrix} \qquad 2.37$$

Y^* is a vector of the true values for all the measurements and Z is the vector of measurements. The vector ϵ_i is the normally distributed random noise with zero mean and known covariance Σ.

$$\Sigma = \begin{bmatrix} \Sigma_W & 0 \\ 0 & \Sigma_D \end{bmatrix} \qquad 2.38$$

The problem considered is the approximation of Y_i based on the model and measurement equations. From the model and measurement equations collect the $k+1$ measurements and the k constraints as follows:

$$Z = Y^* + \epsilon$$
$$\Phi_k Y^* = 0 \qquad 2.39$$

where $Z = [\, Z_1^T, \ldots, Z_{k+1}^T \,]^T$, $Y^* = [\, Y_1^{*T}, \ldots, Y_{k+1}^{*T} \,]^T$, $\epsilon = [\, \epsilon_1^T, \ldots, \epsilon_{k+1}^T \,]^T$ and

$$\Phi_k = \begin{bmatrix} B_d & -E & 0 & \cdots & 0 \\ 0 & B_d & -E & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & B_d & -E \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_k \end{bmatrix} \qquad 2.40$$

Now the dynamic data reconciliation problem can be formulated as in the steady state case.

$$\text{Min } J = \frac{1}{2}(\hat{Y} - Z)^T \Sigma^{-1}(\hat{Y} - Z)$$

$$\text{Subject to} \qquad \Phi_k \hat{Y} = 0 \qquad 2.41$$

$\hat{Y}$ is the vector of calculated estimates for the measurements. The solution is the same as for the steady state data reconciliation.

$$\hat{Y} = Z - \Sigma\Phi_k^T(\Phi_k\Sigma\Phi_k^T)^{-1}\Phi_k Z \qquad 2.42$$

From the solution it can be seen that the computational effort increases with the number of observations, leading to numerical problems such as round off errors and singularities. To avoid these, Darouach and Zasadzinski (1991) propose a recursive solution procedure.

2.2.3. Unbiased estimation in dynamic data reconciliation

The essence of this technique proposed by Rollins and Devanathan (1993) is a backward difference approximation. A constrained least squares approach is used to obtain an optimal solution that guarantees improved values for process variables. The authors claim that this approach is computationally simpler than the method of Darouach and Zasadzinski (1991).

Rather than solve for estimates over all the time intervals, apply Eqs. (2.39 to 2.42) over two successive time intervals. For each time instant i ($i = 2, \ldots, k$) an estimate for the value of each variable at time instant i and i - 1 is determined. Thus, for $i = 1$ and k one estimate is determined for each variable at time instants 1 and k, while two estimates are generated for each variable at time instants $i = 2, \ldots, k - 1$.

In the present formulation of the optimization problem two estimates for the accumulation variables are obtained for all the time instants, except for the first and the last. It was proposed that the estimation accuracy is further improved by averaging, when two estimates

exist for one variable at one time instant. Further improvement is achieved by re-estimating these process variables using the averaged estimates as if they were the original measurements. This is necessary because the averaged estimates do not satisfy the material balance constraints. However the non-averaged estimates at each iteration satisfy the material balance constraints and can be used if these properties are needed. The process variable estimators are unbiased and have known distributions. This technique was motivated by the recursive dynamic data reconciliation of Darouach and Zasadzinski (1991), which obtained very accurate estimators.

A comparison of the techniques of Rollins and Devanathan, and Darouach and Zasadzinski (performed by Rollins and Devanathan) showed that the formers' method is computationally faster, but not as accurate when the variances of the process measurements are large. However the accuracy of the estimators, proposed by Rollins and Devanathan, is shown to approach that of the recursive technique of Darouach and Zasadzinski by iteratively recalculating the estimates and as the measurements variances decreased.

2.3. Variable Classification

The classification of variables is related to the subject of data reconciliation. Prior to, or during data reconciliation, it is important to know which unmeasured variables can be uniquely calculated from the measurements.

The variables can be classified as follows:

unobservable	Unmeasured variables that can undergo a feasible change in their values without being detected by the measurement instruments.
observable	Unmeasured variables that are not unobservable.
nonredundant	Measured variables that, upon deletion of the corresponding measurements, become unobservable.
redundant	The remaining measured variables that are not nonredundant.

Detailed treatment of these properties may be found in Stanley and Mah (1981).

2.3.1 Observability

From Eq. (2.25) it is clear that every u_i in the set of unmeasured variables u_{m-r} is unobservable, purely because it is an arbitrary set. Furthermore, any unmeasured variable in the set u_r , that is linearly dependant on u_{m-r} , is also unobservable by association. This translates to any element of u_r being classified as unobservable if and only if the corresponding row of $\boldsymbol{R}_{11}^{-1}\boldsymbol{R}_{12}$ in Eq. (2.25) has at least one nonzero entry (Swartz, 1989). The observable variables correspond to those elements of u_r that can be uniquely determined. This is consistent with Crowe *et al.* (1983), who state that the unobservable variables correspond to the linearly dependent columns in $\boldsymbol{B}$. To see this, we have from Eq. (2.22):

$$\left[B_1 \middle| B_2\right] = \left[Q_1 \middle| Q_2\right]\begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}$$

$$= \left[Q_1 R_{11} \mid Q_1 R_{12}\right]$$

$$B_1 = Q_1 R_{11} \qquad 2.43$$

$$B_2 = Q_1 R_{12}$$

$$\therefore B_2 = B_1 R_{11}^{-1} R_{12}$$

B_2 is linearly dependent on the remainder of the B matrix, B_1, if a row of $R_{11}^{-1}R_{12}$ has one or more nonzero elements.

A similar approach, using sparse LU decomposition, is used by Albuquerque and Biegler (1995) to evaluate observability in dynamic systems.

2.3.2. Redundancy

By definition, a nonredundant variable becomes unobservable upon deletion of the corresponding measurement. It follows that the column representing a nonredundant measured variable in C is zero. This becomes apparent when one takes a closer look at the QR decomposition of B.

$$\left[\boldsymbol{B}_1 \middle| \boldsymbol{B}_2\right] = \left[\boldsymbol{Q}_1 \middle| \boldsymbol{Q}_2\right] \left[\begin{array}{c|c} \boldsymbol{R}_{11} & \boldsymbol{R}_{12} \\ \hline \boldsymbol{0} & \boldsymbol{0} \end{array}\right]$$

$$\left[\begin{array}{ccc|ccc} b_{1,1} & \cdots & b_{1,r} & b_{1,(m-r)} & \cdots & b_{1,m} \\ \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\ b_{z,1} & \cdots & b_{z,r} & b_{z,(m-r)} & \cdots & b_{z,m} \end{array}\right] =$$

$$\left[\begin{array}{ccc|ccc} q_{1,1} & \cdots & q_{1,r} & q_{1,(z-r)} & \cdots & q_{1,z} \\ \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\ q_{z,1} & \cdots & q_{z,r} & q_{z,(z-r)} & \cdots & q_{z,z} \end{array}\right] \left[\begin{array}{ccc|ccc} r_{1,1} & \cdots & r_{1,r} & r_{1,(m-r)} & \cdots & r_{1,m} \\ 0 & \cdots & \cdot & \cdot & \cdots & \cdot \\ 0 & \cdots & r_{r,r} & r_{r,(m-r)} & \cdots & r_{r,m} \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array}\right] \qquad 2.44$$

Now multiplying both sides of Eq. (2.44) by $\boldsymbol{Q}^{\mathrm{T}}$, ie. $\boldsymbol{Q}^{-1}$, gives

$$\left[\begin{array}{ccc} q_{1,1} & \cdots & q_{1,z} \\ \cdot & \cdots & \cdot \\ q_{r,1} & \cdots & q_{r,z} \\ \hline q_{(z-r),1} & \cdots & q_{(z-r),z} \\ \cdot & \cdots & \cdot \\ q_{z,1} & \cdots & q_{z,z} \end{array}\right] \left[\begin{array}{ccc|ccc} b_{1,1} & \cdots & b_{1,r} & b_{1,(m-r)} & \cdots & b_{1,m} \\ \cdot & \cdots & \cdot & \cdot & \cdots & \cdot \\ b_{z,1} & \cdots & b_{z,m} & b_{z,(m-r)} & \cdots & b_{z,m} \end{array}\right] =$$

$$\left[\begin{array}{ccc|ccc} r_{1,1} & \cdots & r_{1,r} & r_{1,(m-r)} & \cdots & r_{1,m} \\ 0 & \cdots & \cdot & \cdot & \cdots & \cdot \\ 0 & \cdots & r_{r,r} & r_{r,(m-r)} & \cdots & r_{r,m} \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{array}\right] \qquad 2.45$$

$\boldsymbol{Q}_2^T$ forms the last z - r to z rows of $\boldsymbol{Q}^T$. From Eq. (2.45) it can been seen that $\boldsymbol{Q}_2^T$ multiplied with any column vector of $\boldsymbol{B}$ of produces a column of zeros; these being the zeros in each column of $\boldsymbol{R}_{11}$ and $\boldsymbol{R}_{12}$ starting after the r^{th} row. Now since a nonredundant variable becomes unobservable upon deletion, it would be placed in $\boldsymbol{B}_2$ and be a linear combination of one or more of the variables in $\boldsymbol{B}_1$. The column vector from a_i , corresponding to a nonredundant variable in $\boldsymbol{A}$, could thus be represented as:

$$a_i = \gamma_1 b_1 + \gamma_2 b_2 + \ldots + \gamma_r b_r$$

$$\text{and} \quad Q_2^T a_i = \gamma_1 Q_2^T b_1 + \gamma_2 Q_2^T b_2 + \ldots + \gamma_r Q_2^T b_r$$

where

$$Q_2^T b_r = 0 \quad \forall\, r \qquad 2.46$$

$$\therefore\ Q_2^T a_i = 0$$

Since $\boldsymbol{C} = \boldsymbol{Q}_2^T\boldsymbol{A}$, the development above proves that a nonredundant measurement will have a column of zeros in $\boldsymbol{C}$. The converse may also be shown to be true, namely that a column of zeros in $\boldsymbol{C}$ implies that the corresponding variable is nonredundant (Swartz, 1989).

The deletion of a nonredundant variable causes it to become unobservable. The column of zeros in $\boldsymbol{C}$, corresponding to a nonredundant measured variable, will cause the reconciliation procedure to return a zero adjustment for that variable. Therefore a nonredundant variable passes through the reconciliation process unchanged. The effect is that the statistical tests cannot be applied to nonredundant variables as these tests are based on the adjustments or residuals generated by the reconciliation procedure.

The condition of a column of zeros in the $\boldsymbol{C}$ matrix defining a nonredundant measurement is sufficient but not necessary for dynamic systems where a specific variable is a linear combination of several time instants, or even for steady state systems with several data sets. As reported by Albuquerque and Biegler (1995), the column associated with a certain measured variable (column of $\boldsymbol{A}$) does not have to lie in the null space of the coefficient

matrix of the unmeasured variables (***B***).

The linear combination of all the columns associated with this variable across all data sets or time instants must lie in this space. For example, consider the following process with one measured variable z_i and one parameter θ where the constraint is $z_{i+1} = z_i + \theta$. If the data is collected across data sets $i = 1,2,3$, then the constraints for the regression problem will be:

$$\begin{aligned} z_3 &= z_2 + \theta \\ z_2 &= z_1 + \theta \end{aligned} \qquad 2.47$$

Rearranging the constraints in matrix form.

$$Az + B\theta = 0$$

$$\begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \end{bmatrix} \theta = 0 \qquad 2.48$$

A projection matrix orthogonal to ***B*** would be

$$Q = \begin{bmatrix} -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \; R = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix} \therefore Q_2 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \qquad 2.49$$

The constraint equations are premultiplied by Q_2^T, and the result is shown below.

$$Cz = Q_2^T Az = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\sqrt{2} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = 0 \qquad 2.50$$

There are no columns of zeros in $Q_2^T A$; however the variable can still be nonredundant if (z_1, z_2, z_3) lies in the null space of C.

2.4. Gross Error Detection

High reliability of process measurement data is extremely important for good monitoring, control and optimization of chemical processes. Measurements corrupted by undetected gross errors result in false control and optimization. In the normal treatment of process data reconciliation, it is assumed that the only errors present in the data are normally distributed measurement errors with zero means and known variances. Process data may contain other types of errors caused by non-random events such as instrument bias or malfunction. These errors are characteristically so large that the measurement bears little relation to the underlying physical quantity (Ripps, 1965). The presence of these gross errors invalidates the statistical basis of the data reconciliation. It is therefore necessary to test the validity of the above assumptions before reconciling the data.

The popular treatment in the literature combines statistical tests on the measurement and constraint residuals with serial elimination search techniques. A serial elimination scheme was first proposed by Ripps (1965) and later extended by Nogita (1972), who defined a statistical test function which was criticized by Mah *et al.* (1976). Mah *et al.* (1976) developed the constraint test and presented an algorithm for the identification of gross errors based on graph-theoretical results. Since then a number of tests have found favour in the literature: the Global Test (Reilly and Carpani, 1963; Almasy and Sztano, 1975; Madron *et al.*, 1977), the Measurement Test (Mah and Tamhane, 1982; Rosenberg et al., 1987; Serth and Heenen, 1986), the Generalized Likelihood Ratio Method (Narasimhan and Mah, 1987), the Recursive Identification of Gross Errors (Crowe, 1988) and the Unbounded / Bounded test (Narasimhan and Harikumar, 1993b).

Ripps (1965) proposed a serial elimination algorithm to detect gross errors based on analysis after data reconciliation had been performed using the approach of Kheun and Davidson

(1961). The Kheun and Davidson (1961) approach had been found to work well for data containing only small errors. However, when gross errors were present, the least squares procedure is inadequate because the individual adjustments appear as the square and a high penalty is imposed on making any single large correction. Thus, a gross error present in one measurement is attributed to a series of small errors in each measurement. This distribution of the gross errors has two disadvantages: it makes the data as a whole highly inaccurate and it does not indicate which measurement contains a gross error to guide instrument repair or similar corrective action (Ripps, 1965).

Ripps' algorithm involved removing measurements and re-evaluating the least squares minimization on the resultant measurement set. The measurement set which caused the minimum in the least squares objective function to be reported was considered to contain no gross errors. Often the minimum in the least squares objective function coincides with the most correct measurement set, but this need not always be the case. The correctness of any possible classification and resultant data reconciliation is best determined by subsequent tests, for example: other parameters computed from the data, consistency of parameters from run to run and recalibration of suspect instruments. If only a few measurements are removed one can usually use the redundancy present in the balance equations to estimate the lost data.

Although Ripps' technique offered a strong tool to the process analyst, it required some knowledge about the process in selecting the suspect measurements. In addition, there is no criterion to determine the number of measurements containing gross errors.

2.4.1. Global test

The global test states that, under the hypothesis that no gross errors exist in the measured variables, the quantity:

$$\tau_{GT} = w_g^T \left[C \Sigma C^T \right]^{-1} w_g$$
$$\text{where} \quad w_g = Cy - d \qquad 2.51$$

follows a χ^2_v distribution with v degrees of freedom, where v = rank (C). This may be tested

against tabulated values of χ^2_v at a given significance level to check for the presence of gross errors (Reilly and Carpani, 1963; Almasy and Sztano, 1975; Madron *et al.*, 1977). The shortcoming of the global test is that it does not give an indication of which measurement contains a gross error. Nevertheless, the global test can still be used together with a serial elimination technique to identify an error free set of measurements. In serial elimination techniques measured variables are removed from the measurement set and the statistical tests applied to the reduced data set. This procedure eventually returns a set of measurements that pass the global test and which could be considered to be free of any gross errors. The serial elimination technique using the global test is computationally intensive and has been surpassed by the measurement test as a method for detecting gross errors. However, the global test without any serial elimination techniques is still a good test of the measurement set under consideration. For completeness, an example of a global test serial elimination technique is presented (Rosenberg *et al.*, 1987). Three sets are used in the serial elimination. Let S be the set of all measured stream flows, L be a temporary set which contains the measurements being deleted in a particular step, and F be the current set of measurements suspected of containing gross errors.

Global Test - Algorithm:

1) Determine t_{max} [1]. Set $t = 0$ and F = 0.
2) Determine the degrees of freedom v for the network of measurements in set S. Calculate τ_{GT}. IF $\tau_{GT} < \chi^2_v$, declare no gross errors in set S and STOP.
3) Set $t = t + 1$. IF $t > t_{max}$ declare all measurements in set S suspect and STOP. ELSE set $P_{min} = 1 - \alpha$. [2] FOR each possible combination L of t measurements DO:

a) Delete the set of t measurements, L, from the network.
b) Obtain the projected process constraints $\boldsymbol{C}$, to reflect the reduced set of measurements, S - L. IF a column of $\boldsymbol{C}$ contains zeros, disregard the set L and choose next. [3]
c) Determine the degree of freedom of the reconciled set of data, L, and calculate τ_{GT}.
d) IF $p\{\chi^2_v < \tau_{GT}\} < P_{min}$ replace F by L and reset $P_{min} = p\{\chi^2_v < \tau_{GT}\}$, ELSE choose next set of t measurements in 3a).

4) IF $P_{min} < 1 - \alpha$ declare all measurements in $\boldsymbol{F}$ suspect and stop, ELSE GOTO 3).

[1] t_{max} is the maximum number of measurements that can be deleted from the original network to leave the degree of freedom of $\boldsymbol{C} = 1$, i.e. when $\boldsymbol{C}$ consists of a single row.

[2] P_{min} is the minimum cumulative probability

[3] Variables with a column of zeros in $\boldsymbol{C}$ will not effect the calculated residuals or the global test statistic. For this reason gross error in these measurements can not be detected.

2.4.2. Measurement test

The measurement test states that, under the hypothesis that no gross errors are present, the measurement residual vector

$$r = y - x^* = \Sigma C^T \left[C \Sigma C^T \right]^{-1} (Cy - d) \qquad 2.52$$

follows a normal distribution. This can be deduced from the assumption that if only random errors are present in the measurements, then the residuals of these adjustments should follow the same statistical distribution as the random errors, namely a normal distribution. The test statistic suggested has the following form (Mah and Tamhane, 1982):

$$r^* = \Sigma^{-1} r$$

$$z_i = \frac{|r_i^*|}{\sqrt{\left[C^T \left[C \Sigma C^T\right]^{-1} C\right]_{ii}}} \qquad 2.53$$

The i^{th} measurement is suspected to contain a gross error if z_i exceeds a critical z_c. One can conclude that the i^{th} measurement contains a gross error if $z_i > z_{\alpha/2}$. The level of significance equals α and $z_{\alpha/2}$ is obtained from statistical tables of the normal variate (Miller and Freund, 1985). The above test can be adjusted to take the effect of carrying out multiple tests, $\{i=1,\ldots,k\}$ into account. For multiple tests the probability of predicting a gross error when in fact there is none (a Type I error), can be much larger than α. It approaches unity as k tends to infinity. To account for this effect and to guarantee that the probability of a Type I error is controlled at α, Mah and Tamhane (1982) proposed that the i^{th} measurement be rejected if

$$z_i > z_{\frac{\beta}{2}}$$

$$\text{where} \quad \beta = 1 - (1 - \alpha)^{\frac{1}{k}} \qquad 2.54$$

since $\beta < \alpha$ for $k > 1$, $z_{\beta/2} < z_{\alpha/2}$ making it more difficult to reject the hypothesis that no gross error is present (Mah and Tamhane, 1982). The critical statistic value is then $z_c = z_{\beta/2}$. The basic algorithm for the measurement test for a system containing gross errors given by Serth and Heenen (1986) is:

The Measurement Test - Algorithm:

1) Apply data reconciliation to the full system.
2) Calculate each z_i.
3) Compare each z_i with z_c. IF $z_i > z_c$ denote measurement i as suspect and place in F.
4) IF F is empty GOTO 7) ELSE remove from the general system the measurements contained in F. Denote the set of measurements under consideration as S-F.
5) Calculate the reconciled estimates of the measurements in S-F.
6) Calculate the corrected values for the measurements in F.
7) If F is empty $y = x$ (no gross errors in the measurements). If F is not empty then y = least square estimates from 5) and the unmeasured variables are calculated in 6).

A defect of the measurement test is that the least squares procedure tends to spread the gross errors over all the measurements, thereby creating large residuals for good measurements. When these residuals fail the test, the corresponding measurements are incorrectly suspected of containing gross errors. One suggested solution to this problem is that the measurement test be applied in an iterative manner. Only the measurement with the largest normalized residual is discarded at each iteration. The iterations end when all the remaining measurements satisfy the measurement test. The Iterative Measurement Test developed by Serth and Heenen (1986) employs this technique. Another problem with successive elimination schemes is that if there are no bounds on the variables being returned by data reconciliation, unrealistic values in the variables could result, for example negative flow rates. Two examples of algorithms that apply checks for bound violation in the reconciled measurements and calculated unmeasured variables are: the Modified Iterative Measurement Test (MIMT) (Searth and Heenan, 1986) and the Dynamic Measurement Test, DMT proposed by Rosenberg *et al.* (1987). The Modified Iterative Measurement Test is presented as it will be applied to the industrial case study in Section 6.

The Modified Iterative Measurement Test - Algorithm:

1) Set iteration counter $k=0$, and F=0.
2) Perform data reconciliation on the system.
3) Calculate each z_i.
4) Select the stream with the largest $|z_i| > z_c$ and add it to F. IF $|z_i| \leq z_c$ for all i stream GOTO 9).
5) Remove the streams in F from S and obtain the projected matrix $\boldsymbol{C}$ for the reduced balance system.
6) Calculate the reconciled estimates, x^* in S - F, $x_k^* = x^*$.
7) Calculate the unmeasured values in F.
8) IF all variables x_k^* satisfy their bounds, increment k and GOTO 3), ELSE remove the last entry in F and replace it with the next largest value of $|z_i| > z_c$ and GOTO 5). IF $|z_i| \leq z_c$ for all the remaining measurements, delete the last entry in F and GOTO 9).
9) The measurements in F are considered to contain gross errors.

The MIMT was tested against the IMT and other statistical tests and found to represent the best combination of speed and effectiveness (Serth and Heenan, 1986).

The Dynamic Measurement Test is similar to the MIMT; however it also includes bound checking on the unmeasured variables. The DMT attempts to enlarge the set of suspect measurements at each step until all the suspect measures are included. Note that in evaluating the z_i values, occasions may occur where two or more measurements have the same z_i value. In these cases all the measurements currently under consideration are considered suspect and placed in F.

2.4.2.1. Uniqueness of the measurement test statistic

With reference to the measurement test statistic presented above, it is possible to have the same z_i value for different variables. If this is the case then the measurements are indistinguishable for gross error detection using the measurement test. The above

phenomenon occurs when there are linear dependencies in the columns of $\boldsymbol{C}$, as will be shown below. It is shown that $z_i = z_j$, corresponding to the i^{th} and j^{th} measurements, if and only if there exists a nonzero constant γ such that $c_i = \gamma\, c_j$, where c_i and c_j are the corresponding columns of $\boldsymbol{C}$. Recall that the residual vector has the following form

$$\begin{aligned} r &= \Sigma C^T\left[C\Sigma C^T\right]^{-1}(Cy - d) \\ &= \Sigma C^T\left[C\Sigma C^T\right]^{-1}C(y - x) \end{aligned} \qquad 2.55$$

with the measurement test statistic calculated from

$$z_i = \frac{\left[C^T V C(y - x)\right]_i}{\sqrt{\left[C^T V C\right]_{ii}}} \qquad 2.56$$

where $V = \left[C\Sigma C^T\right]^{-1}$

Now $z_i = \pm\, z_j$ if and only if

$$\begin{aligned} & C^T V c_i = \gamma C^T V c_j \\ \text{or } & C^T V(c_i - \gamma c_j) = 0 \\ \therefore\ & c_i = \gamma c_j \end{aligned} \qquad 2.57$$

The necessary and sufficient condition for $z_i = \pm\, z_j$ may be restated as

$$Q_2^T(a_i - \gamma a_j) = 0 \qquad 2.58$$

Note however that the condition $a_i = \gamma\, a_j$ is a sufficient but not necessary condition for Eq. (2.58) to be satisfied. If ($a_i - \gamma\, a_j$) is in the null space of $\boldsymbol{Q_2^T}$ it is also a solution for Eq. (2.58), (Iordache *et al.* 1985).

2.4.3. Measurement credibility

Usually the number of measurements suspected of containing gross errors may be extensive in real problems. Madron (1985) formulated a new method based on measurement credibility.

The method defines a maximum value for the gross error that may occur in individual measurements. This method complements other statistical tests and makes it possible to reduce the number of measured quantities which have to be examined during the final phase of gross error elimination.

The data reconciliation problem is formulated as follows:

$$\begin{aligned} &\text{Min}_x\ J = (y - x)^T \Sigma^{-1} (y - x) \\ &\text{Subject to} \\ &\qquad Cx = 0 \end{aligned} \tag{2.59}$$

In the absence of any gross errors $J_{\min}$ has a central chi-square distribution with v degrees of freedom, where v = rank (C) (Madron, 1985).

Assume the j^{th} measurement is corrupted by a constant gross error b_j which will be expressed as a multiple of the standard deviation σ_j. The mathematical expectation of the error ϵ_j is then equal to the size of the gross error b_j.

$$\begin{aligned} &E(\epsilon_j) = b_j = p_j \sigma_j \\ &E(\epsilon_i) = 0 \quad \text{for } i \neq j \end{aligned} \tag{2.60}$$

The expectation of the adjustments is no longer equal to zero.

$$E(x^* - y) = -\Sigma C^T (C \Sigma C^T)^{-1} C E(e) = \chi(\nu_c, \delta) \tag{2.61}$$

In this case J_{min} has a non-central chi-square distribution with v degrees of freedom and parameter of non-centrality δ. The estimation of the parameter of non-centrality is described in Madron (1985). Using this estimate of δ, it is possible to estimate the size of the gross error in individual measurements.

When a serial elimination scheme is applied for identifying the source of gross errors, one may meet the following phenomenon. After a measurement has been deleted, its value is determined as a parameter of the mathematical model. If the calculated values differ significantly from the measurement values, then reservations about the measurements in question may be raised. In most cases *a priori* information is available about the maximum possible error that may occur during that measurement. For example, accumulation in a storage tank must be more than zero and less than the tank capacity. The presence of gross errors corrupts the results of the data reconciliation and may cause unrealistic gross errors to be incorrectly reported in certain measurements. Continuing the above example, if the estimated gross error is larger that the tank capacity, then this measurement may be removed from suspicion because the estimated size of the gross error is physically impossible. The above outlined considerations are the basis for a new method of identifying gross errors (Madron, 1985). The procedure can be divided into four steps:

1) The maximum value of a gross error, which cannot physically be exceeded by an actually occurring gross error, is assessed for each of the measured quantities. This value is denoted $b_{i,max}$.
2) Based on the value of the test criterion J_{min} and δ, the magnitude of the respective gross errors corresponding to J_{min} are estimated for the individual measurements, $| b_i |$.
3) From the measurement set, a subset exists for which the following holds.

$$|b_i^*| > b_{i,\max} \qquad 2.62$$

In this way a subset of suspect measurements are removed, which are not considered sources of gross errors since the estimate of their gross errors is greater than the assumed value $b_{i,max}$.

4) Now the set of measurements suspected as sources of a gross error is found as the difference between the set of suspect measurements assessed by conventional methods and the set of measurements satisfying the inequality above.

2.4.4. Generalized likelihood ratio (GLR)

The GLR method is based on the likelihood ratio statistical test and provides a general framework for identifying any type of gross errors that can be modelled (Narasimhan and Mah 1987). The GLR approach was developed by Willsky and Jones (1974) to identify abrupt failures in dynamic systems.

The serial elimination strategies proposed (Searth and Heenan, 1986; Rosenberg, 1985) exploit the association of a gross error with a measurement, but are not applicable to gross errors that are not directly associated with measurements such as leaks, since the serial elimination strategy assumes that only the eliminated measurements are corrupted by gross errors.

Narasimhan and Mah (1987) propose an alternative strategy which incorporates a serial compensation of gross errors. In this strategy the gross errors are identified and compensated for (using an estimation of the gross errors' magnitude) before attempting to identify any more. The serial compensation strategy can be used to identify multiple gross errors of any type. They show serial compensation to be computationally more efficient than serial elimination.

Process Model

The steady state model of a chemical process in absence of gross errors can be described by

$$\begin{aligned} y &= x + \epsilon \\ Hx &= 0 \end{aligned} \qquad 2.63$$

where $y \equiv$ (n x 1) vector of measurement errors

$x \equiv$ (n x 1) vector of true values for the measurements

$\epsilon \equiv$ (n x 1) vector of measurement errors - normally distributed, zero mean with known covariance Σ

$H \equiv$ (z x n) matrix of constraints (Mass / Energy Balances)

Measurement Bias Model

The model for a bias of unknown magnitude in measurement i is given by

$$y = x + \epsilon + be_i \tag{2.64}$$

where $b \equiv$ unknown measurement bias

$e_i \equiv$ vector with unity in position i, and zero elsewhere

Process Leak Model

The presence of a leak affects the balance constraints

$$Hx - be_j = 0 \tag{2.65}$$

Equation (2.65) represents a mass flow leak in process unit j of unknown magnitude b. For physical leaks the value of b is allowed to have nonnegative values. Any negative values are interpreted as a reflection of unsteady state behaviour.

The GLR Method

It is assumed that one gross error is present. The balance residuals are

$$r_b = Hy \tag{2.66}$$

Since r_b is a linear transformation of y it has a multivariate normal distribution. Therefore, if gross errors are present due to bias or process leaks, the mathematical expectation of the residuals equals:

$$\mathrm{E}(r_b) = bf_i$$

$$\text{where } f_i = \begin{cases} He_i & \text{for bias in measurement } i \\ e_j & \text{for leak from node } j \end{cases} \tag{2.67}$$

If μ is defined as the unknown expected value of r, a hypothesis test for gross error can be formulated.

H_o: $\mu = 0$

H_1: $\mu = bf_i$

The parameter b can be any real number while f_i is any vector from **F**, $\mathbf{F} \equiv \{ He_i, e_j : i = 1, \ldots, n; j = 1, \ldots, z\}$.

The GLR test statistic, T_i is calculated as follows:

$$T_i = \frac{d_i^2}{V_i}$$

where

$$d_i = f_i^T V^{-1} r_b \qquad 2.68$$

$$V_i = f_i^T V^{-1} f_i$$

$$V = H\Sigma H^T$$

This calculation is performed for every vector f_i in F and the test statistic is therefore obtained as:

$$T = \sup \; T_i \qquad 2.69$$

Let f^* be the vector that led to the supremum in Eq. (2.69). The test statistic is compared with a threshold T_c and a gross error is detected if T exceeds T_c. The gross error corresponding to f^* is identified and its magnitude estimated from Eq.(2.70) (Narasimhan and Mah 1987).

$$b' = \frac{f_i^T V^{-1} r_b}{f_i^T V^{-1} f_i} \qquad 2.70$$

T_i has a central chi-square distribution with one degree of freedom. Using arguments similar

to Mah and Tamhane (1982), Narasimhan and Mah (1987) chose the upper 1 - β quantile of the chi-square distribution, $T_C = \chi^2_{1,1-\beta}$. For a given significance α the value of β is computed by:

$$\beta = 1 - (1 - \alpha)^{\frac{1}{p}} \qquad 2.71$$

where p is the number of gross errors hypothesized (cardinality of set F).

2.4.4.1. Equivalence of the GLR test and the measurement test (MT)

The gross error vector, f_i , for bias in sensor i is equal to He_i. Substituting for f_i , in Eq. (2.68) the GLR for a bias in sensor i obtained

$$T_i = \frac{(e_i^T H^T V^{-1} r_b)^2}{e_i^T H^T V^{-1} H e_i} \qquad 2.72$$

performing the following change of variables:

$$z = y$$
$$A = C$$
$$V = H\Sigma H^T = C\Sigma C^T$$
$$r_d = Hz = Cz = Cy$$

Equation (2.72) can be written as

$$T_i = \frac{(e_i^T C^T (C\Sigma C^T)^{-1} Cy)^2}{e_i^T C^T (C\Sigma C^T)^{-1} C e_i} = \frac{(r_i^*)^2}{[C^T (C\Sigma C^T)^{-1} C]_{ii}} \qquad 2.73$$

If the same level of significance is used for both methods, then the GLR test criterion is equal to the square of the measurement test statistic (Narasimhan and Mah, 1987).

2.4.4.2. The unbounded GLR method

In this method proposed by Narasimhan and Harikumar (1993b), gross error detection using the GLR test is performed. After all the suspected gross errors have been detected and compensated for, the bounded data reconciliation problem is solved using a quadratic programming algorithm. The fundamentally different idea is that it is possible to perform gross error detection prior to data reconciliation using the GLR test. The test is efficient in the sense that data reconciliation is performed only once at the final step. In this method the bounds on process variables have no effect on the performance of the gross error detection, hence this test is named the unbounded GLR (UGLR) method.

2.4.4.3. The bounded GLR method

This method proposed by Narasimhan and Harikumar (1993b) makes use of bound information on the values of the process variables in gross error detection. To include bound information, the bounded data reconciliation problem is solved as the first step. The variables that are at their bounds in the solution (restricted variables) and the remaining variables (unrestricted variables) are then tested separately for gross errors. Gross errors in the restricted variables are detected by applying the GLR test to their measurement residuals, and gross errors in the unrestricted variables are detected by applying the GLR test using the constraint residuals due to the unrestricted variables.

In tests presented in Narasimhan and Harikumar (1993b), the bounded GLR (BGLR) was found to detect more gross errors than the UGLR at the expense of a greater number of mispredictions. Both these methods are especially useful when tight bounds on process variables are specified.

2.4.5. Recursive identification of gross errors

A method for the recursive prediction of changes in the statistical tests was formulated by Crowe (1988). The inverses of large matrices are not required and the data reconciliation can easily be recalculated after the deletion of any set of measurements. It has been shown that the decrease in the reconciliation objective function, caused by the deletion of a single measurement, equals the square of the corresponding maximum power measurement statistic

calculated prior to that deletion (Crowe 1988).

The purpose of this method is to predict the effect of deleting any given set of measurements on the statistical tests without carrying out the full reconciliation.

Global test

The typical reconciliation problem can be defined as follows:

$$\begin{aligned} &\text{Min}_x\ J = (y - x)^T \Sigma^{-1} (y - x) \\ &\text{Subject to} \\ &\qquad Hy = 0 \end{aligned} \tag{2.74}$$

The solution, following Crowe *et al.* (1983) can be written as:

$$\begin{aligned} &\epsilon = -\Sigma H^T V^{-1} r_b \\ &\text{where} \\ &\qquad r_b = Hy \\ &\qquad V = H\Sigma H^T \end{aligned} \tag{2.75}$$

From the formulae for the variance of a linear combination of jointly distributed random variables, the mathematical expectation of r_b is zero, and if the variance of the measurements is Σ, then the variance of r_b is V. By substitution of Eq. (2.75) the value of the objective function, J at the solution is

$$J = r_b^T V^{-1} r_b \tag{2.76}$$

The objective function has a chi-square distribution. This is the basis for the global test.

To assess the effect of deleting a particular set of l measurements on the objective function, evaluate the following function, (Crowe 1988):

$$\Delta J = -(r_b^T V^{-1} B) G_l (B^T V^{-1} r_b)$$

where

$$G_l = B^T V^{-1} B \qquad 2.77$$

$\boldsymbol{B}$ is composed of the columns of $\boldsymbol{H}$ corresponding to the deleted measurements while r_b and $\boldsymbol{V}$ are obtained from the original system, as calculated in Eq. (2.75). Equation (2.77) allows the prediction of the reduction in the objective function as a result of each of a sequence of trial deletions, singly, two at a time, and so on. Any set of measurements whose deletion does not reduce the objective function enough to satisfy the chi-square test cannot contain all of the gross errors (Crowe 1988).

The measurements that upon deletion satisfy the chi-square test are then examined further to establish whether each set also satisfies the measurement statistical test. Finally, the unmeasured variables are computed. These results are checked for reasonableness concentrating on cases that return negative results. A set of measurements that, when deleted leads to negative flows, indicates a serious flaw and it is unlikely that all the gross errors have been identified. Negative flows could also arise if the variances assigned (especially for small measurements) are too large (Crowe, 1988).

Measurement test

Under the assumption that the residuals are normally distributed with zero mean and variance Σ_r , Mah and Tamhane (1982) defined the measurement test:

$$z_i = -\frac{e_j^T \Sigma_r^{-1} r}{\sqrt{e_i^T \Sigma_r^{-1} e_i}} \qquad 2.78$$

The change in the statistic of a remaining undeleted measurement, due to the deletion of the measurement corresponding to the j^{th} column in $\boldsymbol{H}$, can be calculated by:

$$\Delta = \frac{h_j M r}{-h_j^T M h_j}$$

Where 2.79

$$M = V^{-1} B G_1^{-1} B^T V^{-1}$$

The measurement test statistic can now be calculated without performing the data reconciliation (Crowe, 1988).

Formulae have been developed to predict the effect of deleting any set of measurements on the objective function and on the measurement statistics. These formulae can be used without having to compute the reconciliation for each case.

2.4.6. Simultaneous data reconciliation and gross error detection

In normal data reconciliation the least square objective function is based on the assumption that the measurements contain no gross errors. Tjoa and Biegler (1991) propose a new objective function that is constructed using a combined distribution. The function takes contributions from random and gross errors into account. This distribution function is also known as a contaminated Gaussian distribution.

The advantages of minimizing a bivariate objective function are that the presence of any gross errors is taken into account in the process of data reconciliation. Thus less bias is introduced in the estimation and simultaneously a gross error detection test can be constructed based on the distribution function (Tjoa and Biegler 1991).

The standard least squares objective function is obtained from the frequency of a Gaussian distribution for an error ϵ with standard deviation σ, which can be written as follows:

$$f = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{1}{2}\epsilon^2\sigma^{-2}) \qquad 2.80$$

Here ϵ represents the difference between the measured variable y and the estimation x^*. In general, the above function is used to describe the frequency of purely random errors and is used to develop the standard least squares criterion. However, if there are gross errors, then a linear combination of distribution functions based on their likelihood is preferred (Tjoa and Biegler, 1991). If the probability of gross error measurements is p_b ($p_b < 0.5$), and the ratio of the standard deviation of the outliers to that of random errors is b_o ($b_o > 1$), then the frequency function for the bivariate distribution can be written as follows:

$$f = \frac{1}{\sigma\sqrt{2\pi}}\left[(1 - p_b)\exp(\frac{1}{2}\epsilon^2\sigma^{-2}) + \frac{p_b}{b_o}\exp(\frac{1}{2}\epsilon^2\sigma^{-2}b_o^{-2})\right] \qquad 2.81$$

The objective function can now be constructed based on the contaminated Gaussian distribution instead of the Gaussian distribution. Tjoa and Biegler developed a hybrid Successive Quadratic Programming (SQP) method tailored to the new objective function. The nonlinear objective function can now be reformulated as the log reciprocal of the error likelihood for each measurement:

$$\min_z \Phi_1(x) = -\sum_{i=1}^{r} \ln\left[(1 - p_b)\exp(-\frac{1}{2}\epsilon_i^2\sigma_i^{-2}) + \frac{p_b}{b_o}\exp(-\frac{1}{2}\epsilon_i^2\sigma_i^{-2}b_o^{-2})\right]$$

Subject to

$$H(z) = 0$$
$$z_l \le z \le z_u \qquad 2.82$$

where

$$\epsilon = (y - x^*) \quad , \quad z = [x^*, u]^T$$

The SQP method is known as an efficient optimization strategy for solving small to moderate

sized NLP problems; it usually takes fewer iterations to converge than other methods. The basic idea of the SQP method is to solve a quadratic programming (QP) problem at every iteration to find a new search direction. This QP subproblem is formed by making a quadratic approximation of a Lagrange function subject to linearized constraints.

At convergence each measurement error can be tested against the combined distribution. If the probability associated with an error (that is suspected to be an outlier) is greater than that of random errors, then the measurement can be identified as an outlier, i.e. as containing a gross error. Thus, the distribution function that is used for the objective function can also be used as a rational basis for a gross error detection test. If the residual ϵ_i satisfies the following inequality, then the corresponding measurement i is identified as containing a gross error.

$$|\epsilon_i| > \sigma_i \sqrt{\frac{2b_o^2}{b_o^2 - 1} \ln\left[\frac{b_o(1 - p_b)}{p_b}\right]} \qquad 2.83$$

Furthermore, Tjoa and Biegler (1991) report that the effectiveness of the gross error detection test is not very sensitive to the values of p_b and b_o. The effectiveness of the test increases slightly as the values of b increases or p decreases. In cases where p_b and b_o are unknowns, values of 0.05 and 10 respectively, give reasonably good results. This result is encouraging because if the standard deviation, p_b and b_o can be specified *a priori,* then the optimization task can be made much easier.

2.4.7. Gross error detection when measurement variances are unknown

Rollins and Davis (1993) have addressed the estimation of an unknown variance - covariance matrix by adapting their model for the system, the Unbiased Estimation Technique (UBET), for an unknown Σ. The UBET consists of an α-level global test. The global test is used to test for the existence of one or more biased measurements or process leaks. In their treatment they also introduce a method for evaluating S, an approximation of Σ, in the presence of changing process conditions.

The variance of the mean residuals takes the following form:

$$V_N = \frac{V}{N} = \frac{\Sigma_\tau + H\Sigma H^T}{N} \qquad 2.84$$

where N represents the number of measurement sets. V has two components; one representing measurement variability $H\Sigma H^T$ and the other representing process variability, Σ_τ. If no significant changes occur in the measured variables while the data is being taken, it is reasonable to neglect the process variability component.

Alternatively, the mean residuals can be calculated from the data matrix Y and the constraints matrix H, as follows.

$$r = YH^T = \begin{bmatrix} r_{11} & \cdots & r_{1z} \\ \vdots & \vdots & \vdots \\ r_{N1} & \cdots & r_{Nz} \end{bmatrix} = [r_1,\cdots,r_z]$$

where

$$Y = \begin{bmatrix} y_{11} & \cdots & y_{1m} \\ \vdots & \vdots & \vdots \\ y_{N1} & \cdots & y_{Nm} \end{bmatrix} \qquad 2.85$$

The data matrix Y has N entries for each variable. The sampling time (each value of N) is the time taken to obtain a complete column of Y. This time can be very large if any of the measured variables are determined by laboratory analysis. The larger the sampling time, the more likely it is that there will be process variations during sampling. Hence, it may not be reasonable to estimate V_N by HSH^T since this calculation involves measurement variability only. Rollins and Davis (1993) recommend that V_N be estimated by S_r since this takes both process and measurement variability into account.

S_r can be determined by using the residual constraint matrix. The sample variance for the j^{th} measured variable and the covariance between the j^{th} and k^{th} measured variables can be

evaluated as follows.

$$s_{rjj} = \frac{1}{N-1}\sum_{i=1}^{N}(r_{ij} - \bar{r}_j)^2$$

$$s_{rjk} = \frac{1}{N-1}\sum_{i=1}^{N}(r_{ij} - \bar{r}_j)(r_{ik} - \bar{r}_k) \qquad 2.86$$

where

$$\bar{r}_j = \frac{1}{N}\sum_{i=1}^{N} r_{ij}$$

When V_N is unknown, Rollins and Davis (1993) propose the following α-level global test to test for gross errors. The presence of gross errors is suspected if the following inequality holds true.

$$N\bar{r}^T S_r^{-1}\bar{r} \geq \frac{(N-1)z}{N-z}F_{z,N-z,\alpha} \qquad 2.87$$

This statistic is known as the Hotelling T^2 statistic. Rollins and Davis (1993) also reported component statistical tests and showed their test to be effective in identifying gross errors when the measurement covariance is not known.

2.4.8. Effect of the choice of regression variables in nonlinear gross error detection

In the capture of data, total flow rates and compositions are the process variables that are usually measured. However, it is often convenient to formulate the balance / constraint equations in terms of total and component flows. These two sets of variables will be referred to here respectively as primary and secondary variables. Likewise, it is possible to formulate the data reconciliation problem either in terms of primary or secondary variables. The fact that the material balance equations are linear in the secondary variables is exploited to develop an efficient data reconciliation procedure (Crowe *et al.*, 1989).

The drawback of using the secondary variables for data reconciliation is that this procedure does not produce maximum likelihood estimates for the true values of the primary measured

variables. The question was raised by Crowe *et al.* (1989) as to what effect the choice of regression variables might have on the performance of a gross error detection algorithm. Since the two sets of regression variables will yield different sets of residuals, the choice of variables could affect the gross error detection procedure.

In an investigation by Crowe *et al.* (1989) the problems of the data reconciliation and gross error detection were formulated using both the primary and secondary variables. In the case where primary variables were used the data reconciliation procedure was solved using the method of Britt and Luecke (1973) and when the secondary variables were used the matrix projection method of Crowe (1986) was implemented to find the solution. In both cases the modified iterative measurement test, MIMT (Searth and Heenan, 1986) was used for gross error detection.

The results of this investigation showed that the use of secondary variables as regression variables for nonlinear data reconciliation adversely affected the performance of the gross error detection algorithm. The effect was an increased tendency to make Type I errors; that is, incorrectly identifying a measurement as containing a gross error. In other aspects the algorithm's performance did not differ greatly from that obtained by regressing on the primary variables. The results showed that for systems having small numbers of variables and constraint equations for which the computational effort is small, data reconciliation should be performed by regressing on the primary variables. For large systems where computational efficiency is important, the best approach may be to regress on the secondary variables and use the primary variables for gross error detection.

2.5. Further Developments

2.5.1. Updating the Jacobian matrices using Broyden's method

In the iterative scheme described in subsection 2.1.4., $\boldsymbol{A}$ and $\boldsymbol{B}$ need to be re-evaluated for each iteration in step 1. In most chemical process applications the derivatives are estimated by finite differences of implicit functions which are very time consuming to evaluate.

Therefore, the computing time for step 1 is likely to be large. This prompted Knepper and Gorman (1980) to use old values of derivatives until the constraints were satisfied. However, this approach leads to slow convergence.

Broyden (1965) developed a method of updating the Jacobian matrix in order to overcome the disadvantages of Newton's method for solving systems of equations. The method uses the function residuals at the new point to generate from the old Jacobian an approximation of the Jacobian at the new point. Broyden's method has been extended to the case of nonlinear data reconciliation to update the $\boldsymbol{A}$ and $\boldsymbol{B}$ matrices (Pai and Fisher, 1988).

Pai and Fisher (1988) concluded that in most cases one initial evaluation of $[\ \boldsymbol{A}\ |\ \boldsymbol{B}\ ]$, is sufficient for the algorithm to reach a good solution and that the convergence rate was faster than when the constant direction approach of Knepper and Gorman (1980) is used.

2.5.2. Estimation of an unknown variance - covariance matrix

In all the theories presented in this report for data reconciliation and gross error detection, the variance - covariance matrix is an important feature. This matrix serves as a weighting function between the various measurements. Throughout the theory it has been assumed that this variance - covariance matrix has been available, but no indication of it's origin has been presented. It is possible that from familiarity with the process, an estimate of this matrix may be obtained. Another suggestion has been that one can follow a statistical approach based on sample sets of measurements over a period of time to estimate a variance - covariance $\boldsymbol{S}$ from this sample variance. The ability of $\boldsymbol{S}$ to accurately estimate $\boldsymbol{\Sigma}$ increases as N (the number of samples) increases. However as N increases the time over which the samples are taken increases. The probability of a departure from steady state increases with an increasing sample time. Departure from steady state has to be accounted for.

If the process is truly at steady state, the estimates using the sample variance and covariances should prove adequate and simple to use from the direct method (Almasy and Mah, 1984):

$$s_{ii} = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ik} - \bar{x}_i)^2$$

$$s_{ij} = \frac{1}{N-1}\sum_{k=1}^{N}(x_{ik} - \bar{x}_i)(x_{jk} - \bar{x}_j) \qquad 2.88$$

where

$$\bar{x}_i = \frac{1}{N}\sum_{k=1}^{N} x_{ik}$$

where N represents the number of measurement sets.

Almasy and Mah (1984) considered the steady state case and have presented a method for estimating the covariance matrix by using the constraint residuals calculated from available process data. They minimize the sum of squares of the off-diagonal elements of a covariance matrix subject to the relation deduced from statistical properties of residual constraints. This method gives an analytical solution which is very sensitive to the correlated measurements.

Darouach *et al.* (1989) have proposed a method for estimating a diagonal covariance matrix based on the maximum likelihood estimator, the material balance constraints and the statistical properties of the residuals. Their algorithm couples the two problems of variance estimation and data reconciliation. They resolve a nonlinear optimization problem and obtain an iterative solution for variance estimation which results in data reconciliation being hindered by computational and convergence problems.

Keller *et al.* (1992) have proposed a method, similar to that of Almasy and Mah (1984), of estimating the covariance matrix, based on the relation deduced from the statistical properties of material balance constraints. The authors claim that their method is more robust with regard to the correlated measurements and more suitable in practical situations than the above mentioned algorithms.

The method of estimating the covariance matrix in the presence of changing process conditions (Rollins and Davis, 1993) has already been presented in subsection 2.4.7.

2.6. The Development of Data Reconciliation in the Minerals Processing Arena

There has been a parallel development of data reconciliation techniques in the minerals processing industry, with little reference to the chemical engineering literature or use of chemical engineering examples. Mineral processing flowsheets are similar to chemical flowsheets and may be defined as a set of process streams connected by nodes. These nodes can be pieces of process equipment, a flotation cell for example, or a point where streams join.

The principle aim of a material balance on a mineral processing flowsheet, is to calculate the mass flowrates in the circuit. Measurements can be of two types: 1) mass flow measurements or 2) assay measurements. Examples of assay measurements are: percent iron, percent minus 100 mesh and percent solids in any stream. Assay measurements are analogous to composition measurements in the standard chemical engineering approach.

Much of the theory developed is based on the two product rule, which performs the material balance around a node with one input (w_1) and two outputs (w_2 and w_3). For each component that has been assayed the mass balance equations are:

$$w_1 - w_2 - w_3 = 0$$
$$w_1 x_1^k - w_2 x_2^k - w_3 x_3^k = 0 \qquad 2.89$$

where x_i^k is the measured assay value of the k^{th} component in stream i. If any one of the mass flows is known, it is a simple task to calculate the remaining flows by using the two product formula.

$$\frac{w_2}{w_1} = \frac{x_1^k - x_3^k}{x_2^k - x_3^k} \qquad 2.90$$

This two product formula can easily be expanded to higher orders.

In general, the problem will be over defined: for example two assays per stream will allow two different values to be calculated for the flow rate. In addition, the presence of measurement errors will also add to the confusion.

The common way to describe a mineral processing flowsheet is to construct an incidence matrix, $\boldsymbol{H}$. The mass balance equations for the mass flow rates at each node i can be written in summation form.

$$\sum_{j=1}^{m} H_{ij} w_j^* = 0 \qquad 2.91$$

If the solids consist of K various assayed components, the mass balance at each node i for component k can be written in summation form.

$$\sum_{j=1}^{m} H_{ij} x_j^{k*} w_j^* = 0 \qquad 2.92$$

In a mathematical sense minimizing the adjustments made to the measurements is equivalent to minimizing the weighted sum of squares:

$$\mathrm{Min}_{x_j^{k*}, w_j^*} J = \mathrm{Min}_{x_j^{k*}, w_j^*} \sum_{k=1}^{K} \sum_{j=1}^{m} \frac{(x_j^{k*} - x_j^k)^2}{(\sigma_j^k)^2} + \sum_{j=1}^{m} \frac{(w_j^* - w_j)^2}{(\sigma_j)^2} \qquad 2.93$$

such that w_j^* and x_j^{k*} satisfy the mass balance equations.

2.6.1. Solution techniques

Essentially two techniques for the solution of the material balance problem (defined above) have been presented: 1) Lagrangian multipliers or 2) Direct search techniques.

2.6.1.1. Lagrangian multipliers

The basis of the Lagrangian multipliers technique is the combination of J and the mass balance constraints into a single function,

$$L_\lambda = J + \sum_{i=1}^{n} \lambda_i \sum_{j=1}^{m} H_{ij} w_j^* + \sum_{k=1}^{K} \sum_{i=1}^{n} \lambda_i^k \sum_{j=1}^{m} H_{ij} x_j^{k*} w_j^* \qquad 2.94$$

where λ_i and λ_j^k are unknown Lagrange multipliers. This approach is similar to the one followed by the chemical engineering literature. The material balance problem then reduces to minimizing L_λ with respect to λ_i, λ_j^k, w_j^* and x_j^{k*}. Wiegal (1972) developed the MATBAL computer program for the solution of this problem using a gradient method. This involves a partial differentiation of L for each of the unknowns and then setting the derivatives to zero. The result is a set of nonlinear equations, one for each unknown, hence an iterative solution is required.

Cutting (1976) solves the material balance problem along similar lines to that of Wiegal (1972); however the iterative method employed by Cutting to solve the resulting system of nonlinear equations follows the direct solution (Keuhn and Davidson, 1969).

In problems where none or only one of the mass flows is known, working with the relative mass flows in the circuit leads to a simplified problem. This following method proposed by Smith and Ichiyen (1973) presents a problem which is again formulated as a least squares optimization in the assay variables.

$$J = \sum_{i=1}^{m} (x_i^* - x_i)^T \Sigma (x_i^* - x_i)$$

Subject to 2.95

$$H(x_i^*, w^*)$$

In a step similar to the QR decomposition the mass balances are rearranged in the following form as functions of the assay variables only.

$$Cx_i = 0 \qquad 2.96$$

For some initial estimate of w^* an adjustment to the assay vector which minimizes J is computed:

$$x_i^* = x_i - \Sigma C^T [C \Sigma C^T]^{-1} C x_i \qquad 2.97$$

J is evaluated at w^* and the new estimate x_i^*. A direct search algorithm is then used to find the minimum over w^* of J. The previous two steps are repeated at every iteration until a pre-specified level of convergence is reached. Smith and Ichiyen (1973) recommend the method of Rosenbrock (1960) which consists of a one dimensional minimization for each element of w^*.

2.6.1.2. Direct search technique

The alternative to using Lagrangian multipliers is the use of direct search techniques. The principle in these methods is to define search and non-search variables. The search variables must be an independent set from which all other variables can be calculated using the material balance constraints. In this case, only the J function needs to be minimized. This is achieved by moving over the response surface J, defined by the search variables. White *et al.* (1977) use a one dimensional iterative technique to perform this direct search.

The problem is formulated as a normalized weighted sum of squares containing all the variables (search and non-search) in the objective function J.

$$J = \sum_{i=1}^{m} \frac{[\sigma_i(x_{ig} - x_{ia})]^2}{x_{ig}} \qquad 2.98$$

where x_{ig} = is the given i^{th} variable value

x_{ia} = is the adjusted i^{th} variable data value

The direct search is perfomed as follows. For each search variable two successive steps equal to Δ (a fixed percentage of the given data value) are taken and the corresponding Js are evaluated. If the second J is smaller than the first, a step of Δ from point 1 is taken in the opposite direction and J is evaluated. Additional steps of 2Δ, 4Δ, etc. are taken until J again begins to increase. At this point a quadratic is fitted to these three points and the quadratic is minimum computed. The corresponding x_{ia} replaces x_{ig} in J and the procedure is repeated for the next search variable. It has been shown that this technique is more powerful than a Fibonacci or Golden Section searches due to the use of *a priori* knowledge of the nature of the J response surface, (White *et al.*, 1977).

2.6.1.3. Alternative solution methods

A few researchers have proposed new methods based on the theory developed above. Hodouin and Everell (1980) employ an approach in which the function J is decomposed into two parts; one which contains the assay data, J^a, and the other which contains the flowrate data, J^d. The functions J^a and J^d are then minimized under the material balance constraints. The procedures may be analytical, that is involving Lagrangian multipliers, or may involve an additional set of search variables. This method may be viewed as a combination of the Lagrangian and direct search techniques.

Kimpel (1983) developed equations for the calculation of stream flow rates for individual splitter and mixer units for different types of experimental error models. Traditionally, the

error is assumed to be normally distributed, but Kimpel developed equations based on the two product formulae that take double exponential and rectangular error distributions into account.

3. SENSITIVITY ANALYSIS

This section is dedicated to evaluating the sensitivity of the reconciliation procedure and statistical tests to the measurement covariance matrix. Formulae were developed for the derivative of the reconciliation equation with respect to individual measurement variances. The predictions of the derivative matrix were then tested by manually changing a specific measurement variance and evaluating the effective change in the reconciled measured variables.

3.1. Development of the Derivative of the Reconciliation Equations

The solution to the data reconciliation problem, as formulated in subsection 2.1.4, has the following form.

$$x^* = y - \Sigma C^T \left[C \Sigma C^T \right]^{-1} (Cy - d) \tag{3.1}$$

The following development is based on the assumption that the covariance matrix is diagonal, which corresponds to the measurements being uncorrelated. The proposed method is to take the derivative of Eq. (3.1) with respect to the individual measurement variances σ_i, contained in Σ. Σ appears as a product in Eq. (3.1), therefore the product law for matrices (Eitelberg, 1991) will be used in the derivation of the derivative of Eq. (3.1). Taking the derivative of Eq. (3.1) with respect to individual measurement variances yields the following:

$$\begin{aligned} \frac{\partial x}{\partial \sigma_k} &= - \frac{\partial}{\partial \sigma_k} \left\{ \Sigma C^T \left[C \Sigma C^T \right]^{-1} (Cy - d) \right\} \\ &= - \left\{ \left[\frac{\partial}{\partial \sigma_k} \Sigma \right] C^T \left[C \Sigma C^T \right]^{-1} + \Sigma C^T \left[\frac{\partial}{\partial \sigma_k} \left[C \Sigma C^T \right]^{-1} \right] \right\} (Cy - d) \end{aligned} \tag{3.2}$$

The individual derivatives are now evaluated separately:

$$\frac{\partial}{\partial \sigma_k}\Sigma = \begin{bmatrix} \frac{\partial}{\partial \sigma_k}\sigma_1 & \cdots & 0 \\ 0 & \frac{\partial}{\partial \sigma_k}\sigma_k & 0 \\ 0 & \cdots & \frac{\partial}{\partial \sigma_k}\sigma_n \end{bmatrix} \qquad 3.3$$

This is a matrix whose k^{th} diagonal element is unity, and zero everywhere else. The derivative with respect to σ_k when it is embedded in a matrix inversion, has a convenient formula derived in Appendix A.

$$\frac{\partial}{\partial \sigma_k}\left[C\Sigma C^T\right]^{-1} = -\left[C\Sigma C^T\right]^{-1}\left\{\frac{\partial}{\partial \sigma_k}\left[C\Sigma C^T\right]\right\}\left[C\Sigma C^T\right]^{-1} \qquad 3.4$$

The three product terms of Eq. (3.4) are evaluated as follows:

$$\left[C\Sigma C^T\right] = \begin{bmatrix} \sum_{k=1}^{n} C_{1,k}C_{1,k}\sigma_k & \cdots & \sum_{k=1}^{n} C_{1,k}C_{z,k}\sigma_k \\ \vdots & \cdots & \vdots \\ \sum_{k=1}^{n} C_{z,k}C_{1,k}\sigma_k & \cdots & \sum_{k=1}^{n} C_{z,k}C_{z,k}\sigma_k \end{bmatrix} \qquad 3.5$$

The derivative of Eq. (3.5) with respect to σ_k equals the k^{th} column of $\boldsymbol{C}$ multiplied by its transpose:

$$\frac{\partial}{\partial \sigma_k}\left[C\Sigma C^T\right] = \begin{bmatrix} C_{1,k}C_{1,k} & \cdots & C_{1,k}C_{z,k} \\ \vdots & \cdots & \vdots \\ C_{z,k}C_{1,k} & \cdots & C_{z,k}C_{z,k} \end{bmatrix} = c_k c_k^T \qquad 3.6$$

where c_k is the k^{th} column of $\boldsymbol{C}$. The derivative of the data reconciliation equation with

respect to the individual measurement variances can now be stated.

$$\frac{\partial x}{\partial \sigma_k} = -\left\{\left[\frac{\partial}{\partial \sigma_k}\Sigma\right] C^T\left[C\,\Sigma\,C^T\right]^{-1} + \Sigma\,C^T\left[C\,\Sigma\,C^T\right]^{-1}\left[c_k\,c_k^T\right]\left[C\,\Sigma\,C^T\right]^{-1}\right\}\left(C\,y - d\right) \tag{3.7}$$

The entries in the derivative matrix, Eq. (3.7), may vary greatly in orders of magnitude due to the different sizes of the measurements and variances. Therefore each element is scaled by its measured value and variance so that direct comparison is made possible. Each element is divided by its associated measured value and multiplied by its associated variance, corresponding to a relative ratio, as shown in Eq. (3.8). Comparisons can now be drawn about the sensitivity of any measurement because the results are reported as relative changes and therefore directly comparable.

$$\frac{\partial x_i}{\partial \sigma_j}\cdot\frac{\sigma_j}{x_i} = \lim_{\Delta\sigma_j \to 0}\frac{\Delta x_i / x_i}{\Delta\sigma_j / \sigma_j} \tag{3.8}$$

From the equation for the calculation of the unmeasured variables, Eq. (2.25), the extension of the derivative matrix to include the effect of the measurement variances on the unmeasured variables is straightforward.

$$u_r = R_{11}^{-1}(Q_1^T c - Q_1^T A x - R_{12}u_{m-r})$$

$$\frac{\partial u_r}{\partial \sigma_k} = -R_{11}^{-1}Q_1^T A\left[\frac{\partial x}{\partial \sigma_k}\right] \tag{3.9}$$

The sensitivity analysis of all the examples in this chapter was performed using MATLAB. MATLAB is a mathematical script language that uses input m-files to execute a list of commands.

3.2. Investigation of a Simple Network's Sensitivity to the Variance Matrix.

To investigate the sensitivity of a system with respect to measurement variances, a simple flow network was selected, as shown in Fig 3.1. The system's flow rates were arbitrarily chosen and then distorted with random noise to achieve the characteristics of plant measurements. The standard deviation of the measurements was assumed to be five percent of the measured flow rates. The model was further simplified in that only a single component was present in the network. Appendix B presents the balance equations and m-files describing the sensitivity analysis of this simple network.

Table 3.1 Measured Variables and Variances for the Simple Flow Network.

	A	B	C	D	E	F	G
Flow	6.0632	17.0443	17.0886	6.0327	11.5453	6.0479	6.0431
σ	0.0919	0.7301	0.7263	0.0910	0.3332	0.0914	0.0913

Analysis of Table 3.1 reveals measurements that have approximately the same flow rate, and hence variance, differing only in their position in the network.

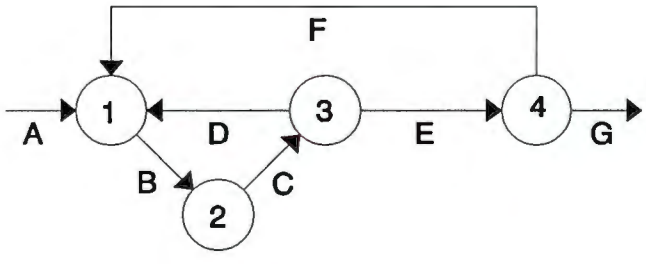


Figure 3.1 Diagram of the simple flow network.

The sensitivity matrix for this network is shown in Table 3.2. The entry corresponding to position (x_A , σ_B) in Table 3.2 represents the sensitivity of reconciled measurement A to the

variance of measurement B. In order to rank the sensitivity of all the reconciled measurements to a specific variance, the 1-norm of each of the columns in Table 3.2 is calculated. The 1-norm is defined as the sum of the absolute values of the column entries.

Table 3.2 Sensitivity Matrix of the Simple Network Example.

	σ_A	σ_B	σ_C	σ_D	σ_E	σ_F	σ_G
x_A	-0.0083	0.0028	0.003	0.0014	0.0031	0.0049	-0.0069
x_B	-0.0016	0.0057	0.0062	-0.0057	0.0026	-0.0059	-0.0013
x_C	-0.0016	0.0057	0.0062	-0.0058	0.0026	-0.0059	-0.0013
x_D	0.0011	0.0078	0.0085	-0.0206	-0.0019	0.0041	0.0009
x_E	-0.003	0.0044	0.0047	0.0023	0.0049	-0.0108	-0.0025
x_F	0.0027	0.0055	0.006	0.0029	0.0062	-0.0256	0.0022
x_G	-0.0083	0.0028	0.003	0.0014	0.0031	0.0049	-0.0069
1 Norm	0.0266	0.0347	0.0376	0.0402	0.0244	0.0621	0.0222
Rank	5	4	3	2	6	1	7

Table 3.2 predicts that the system is more sensitive to the variances of measurements D and F than A and G. These variables are highlighted because they all have approximately the same flowrate and variance. Measurements A, D, F and G differ only in their position in the network, and yet they have varying sensitivities.

In order to demonstrate visually the sensitivity of the system to the measurement variances, individual measurement standard deviations were adjusted from five to one hundred percent of the respective flow rate. The resulting reconciled measured variables were compared to the base case (standard deviation equals five percent of the flow rate), and a percentage change in the reconciled variable was calculated. Figures 3.2 to 3.8 represent graphs of the percentage change in reconciled flow rate versus the increase in measurement standard deviation and hence variance, plotted for each measurement variance. For the variation of each variance, a graph is produced plotting the individual percentage changes in the reconciled flow rates for each measurement.

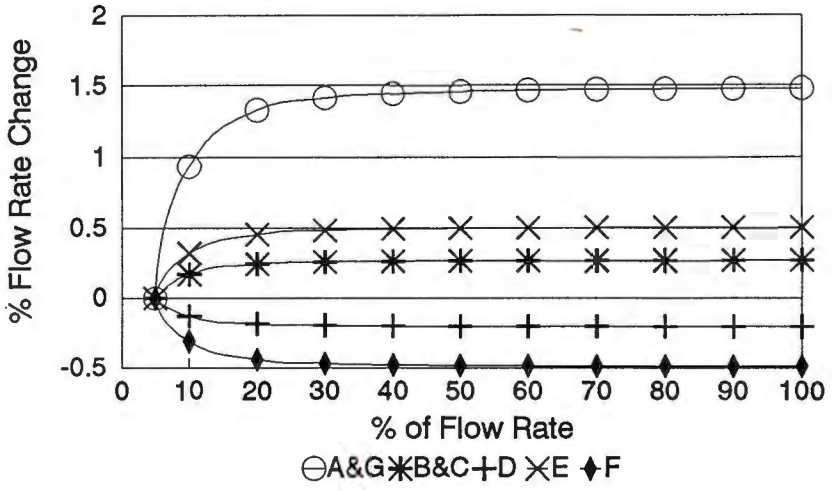


Figure 3.2 Percentage change in the system flowrates due to changes in variance A.

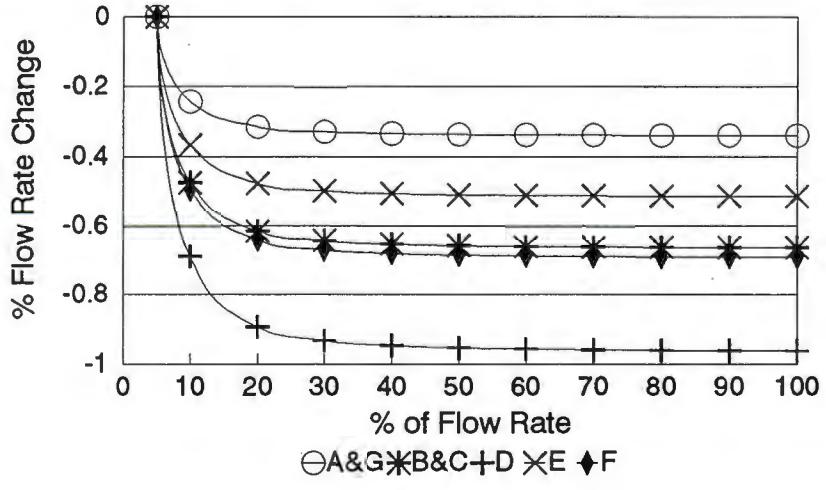


Figure 3.3 Percentage change in the system flowrates due to changes in variance B.

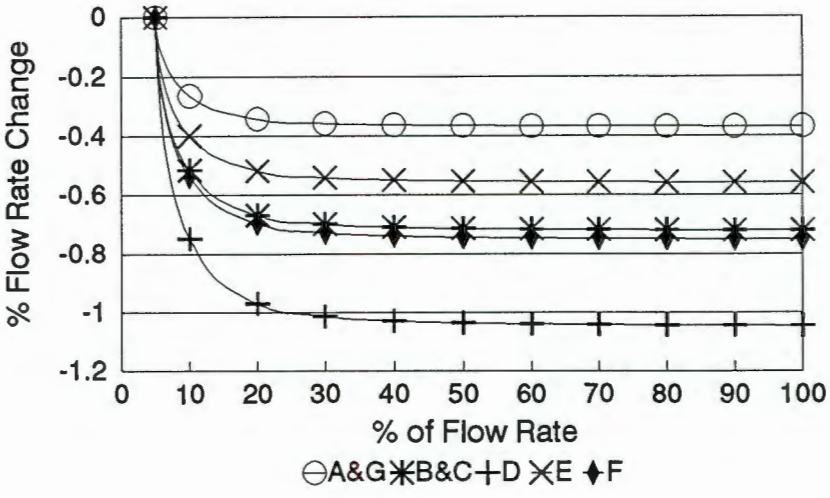


Figure 3.4 Percentage change in the system flowrates due to changes in variance C.

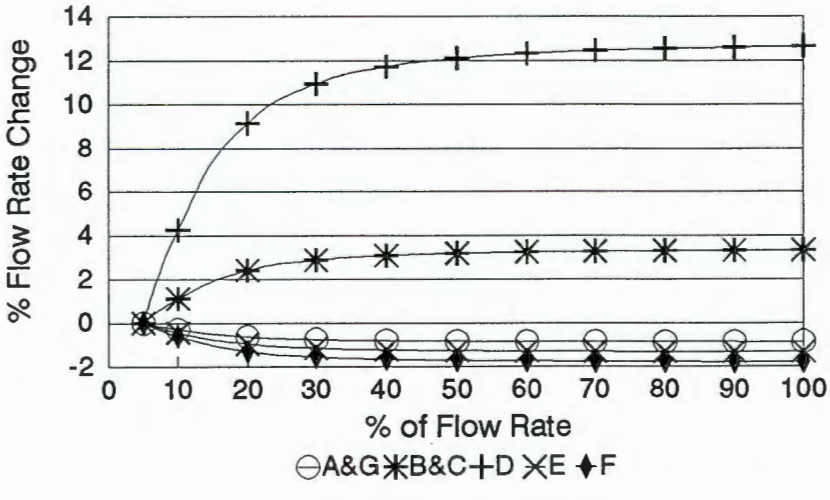


Figure 3.5 Percentage change in the system flowrates due to changes in variance D.

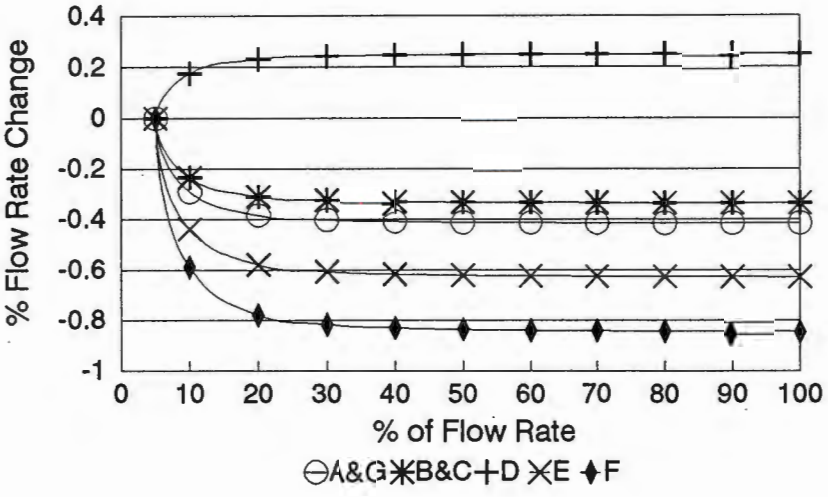


Figure 3.6 Percentage change in the system flowrates due to changes in variance E.

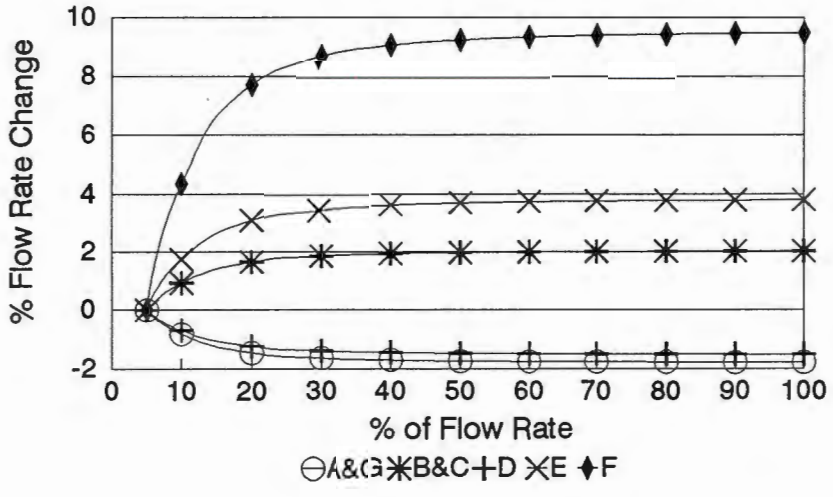


Figure 3.7 Percentage change in the system flowrates due to changes in variance F.

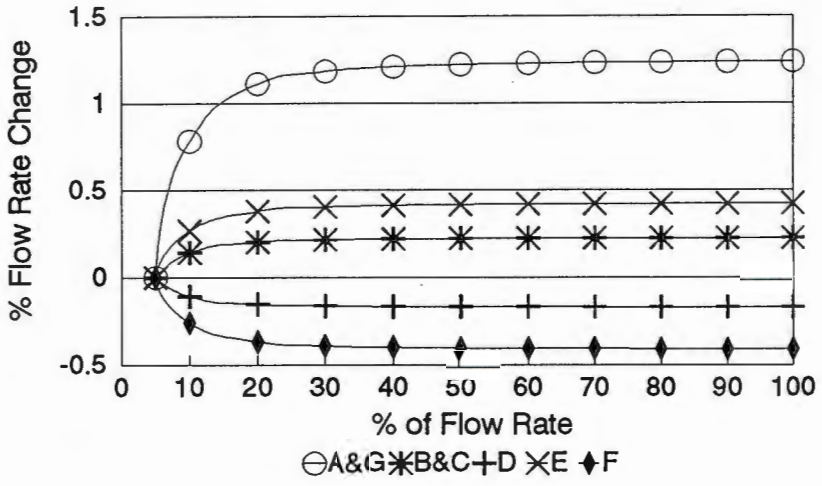


Figure 3.8 Percentage change in the system flowrates due to changes in variance G.

Figures 3.2, 3.5, 3.7 and 3.8 show that the system is more sensitive to changes in the measurement variance of measurements D and F than A and G. Changes in the variance of the D and F measurements almost a nine to thirteen percent change in some of the variables, as seen in Fig. 3.5 and 3.7. Changes in the A and G variances produce less than one and a half percent change in all of the variables, as seen in Fig. 3.2 and 3.8.

The sensitivity matrix successfully predicted the sensitivity of the reconciliation procedure to individual measurement variances. In addition, it highlighted differences in sensitivities to seemingly similar measurements. The reconciliation procedure exhibits large differences in its sensitivity to different measurement variances that are similar in all aspects except for their position in the network. The prediction of this varying sensitivity would be impossible without applying the sensitivity analysis.

3.3. Sensitivity Analysis of a Steam-Metering System

Steam-metering systems of industrial chemical processes constitute an important area of application for data reconciliation techniques. These systems comprise mass-flow networks characterized by moderate size, relatively complex topography, and a range of flowrates that typically covers two orders of magnitude. The network under consideration is the process steam system for a methanol synthesis unit used by Searth and Heenan (1986) to evaluate a number of gross error detection techniques.

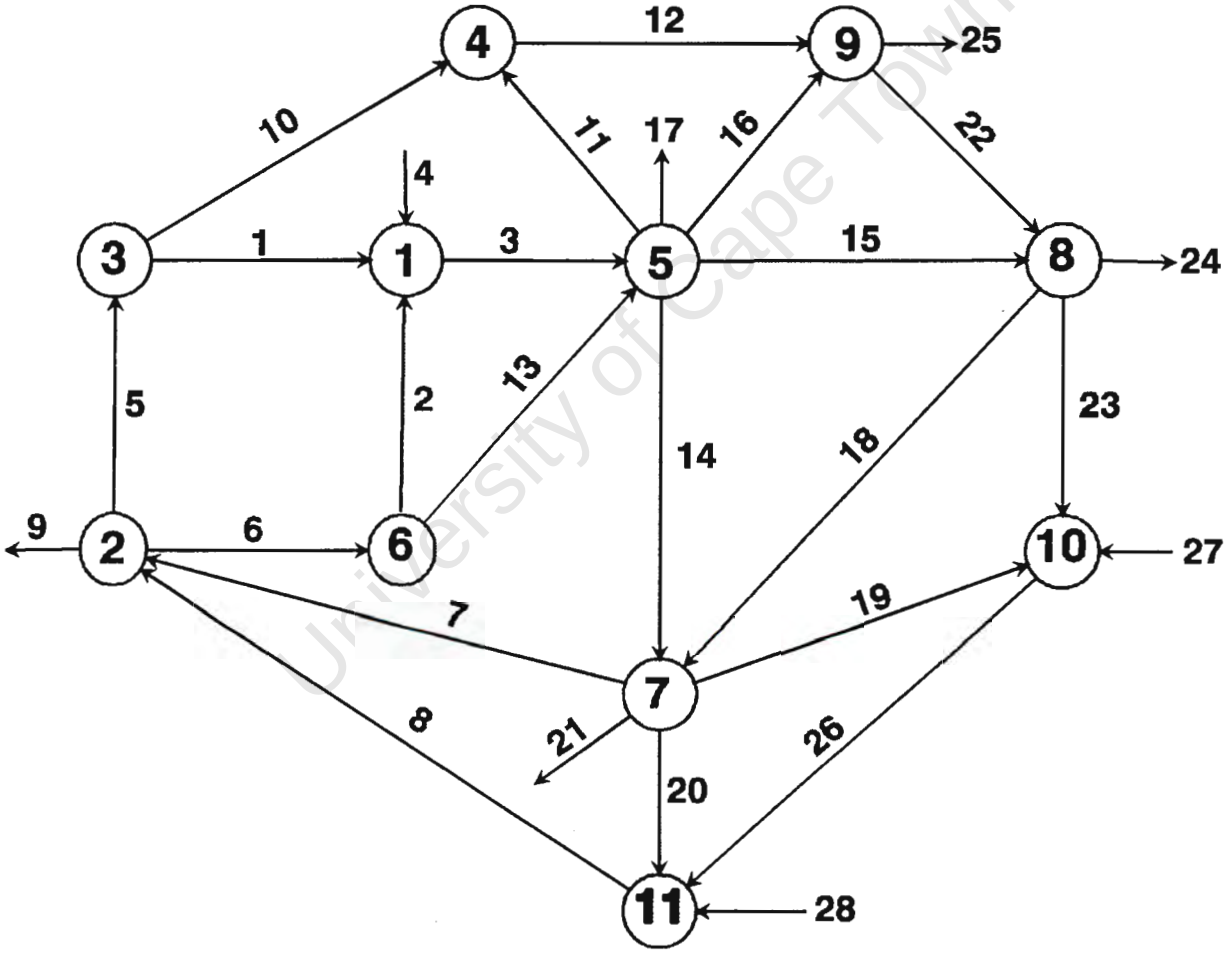


Figure 3.9 Process steam network for a methanol synthesis example.

The network in Fig. 3.9 consists of twenty eight measured streams and eleven nodes (or process units). Searth and Heenan (1986) assumed the standard deviation σ_j of the

measurement error for stream j was 2.5% of the corresponding true flowrate. A table of these true flowrates, together with the rest of the information describing the steam flow network and the m-files for this example are presented in Appendix C. The covariance matrix was then given by

$$\Sigma = \text{Diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_{28}^2) \tag{3.10}$$

The sensitivity matrix of the steam metering system was calculated using Eq. (3.8) and can be found in Appendix C. The column sum (1-norm) was used to rank the effect of the individual variances, as shown in Table 3.3.

Table 3.3 Rank of the effect of the individual measurement variances.

Measurement	23	22	19	24	18	1	20
1-Norm	4.1671	2.0827	1.8035	1.5471	1.4455	0.8519	0.8315
Measurement	17	27	15	5	16	7	11
1-Norm	0.7156	0.6310	0.6207	0.4589	0.4509	0.4337	0.3527
Measurement	26	28	21	3	10	14	13
1-Norm	0.3516	0.2216	0.2117	0.2117	0.2084	0.1698	0.1660
Measurement	6	25	4	2	8	9	12
1-Norm	0.1102	0.1057	0.1056	0.1012	0.0626	0.0589	0.0463

From Table 3.3 it may be inferred that the system is most sensitive to the variance of measurement 23 and least sensitive to the variance of measurement 12. Measurements 7 and 23 have approximately the same value and hence assumed variance. While measurement 23 leads the table of sensitivities, the 7th measurement is situated just past the mid point in the rank of sensitivities. As a check, the variances of the measurements were over-estimated covering a range of percentages (5 -> 85%) and the percentage change in the reconciled flowrates evaluated.

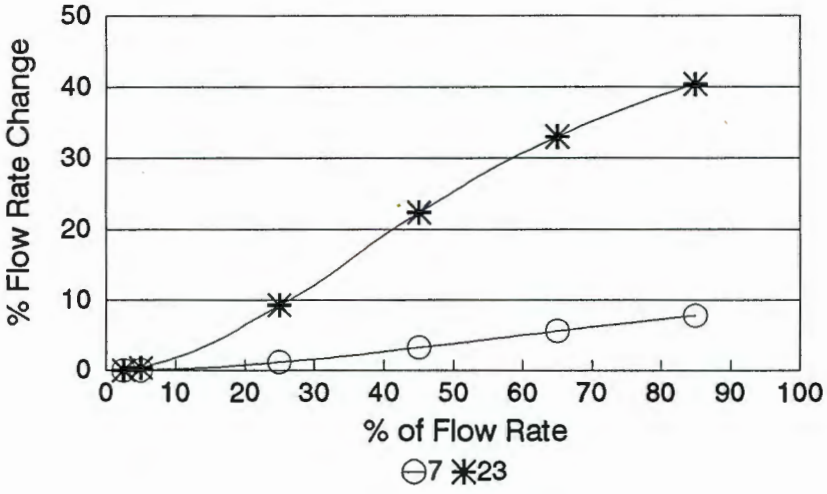


Figure 3.10 Effect of varying the 7th and 23rd variances in the reconciliation of the 7th and 23rd measurements.

Figure 3.10 displays two results for gauging the sensitivity of the system to variations in the variance of the 7th and 23rd measurements. The y axis displays the percentage change in the reconciled values of the 7th and 23rd measurements, while the x axis displays the range over which the standard deviation of the measurements was considered. The standard deviation was varied from five to eighty-five percent of the respective measurement value.

From Figure 3.10 it can be seen that although both measurements have approximately the same size and hence variance, their sensitivities vary greatly. The reconciled value of the 7th measurement varies up to eight percent compared to forty percent for the 23rd measurement. This reaffirms the observation of the previous section that two measurements of approximately the same size may have varying sensitivities based only on their position in the network.

3.4. Sensitivity Analysis of the NH_3 Synthesis Loop

A NH_3 synthesis loop presented by Crowe *et al.* (1993) will be used to demonstrate the sensitivity of gross error detection with respect to the assumed measurement variances.

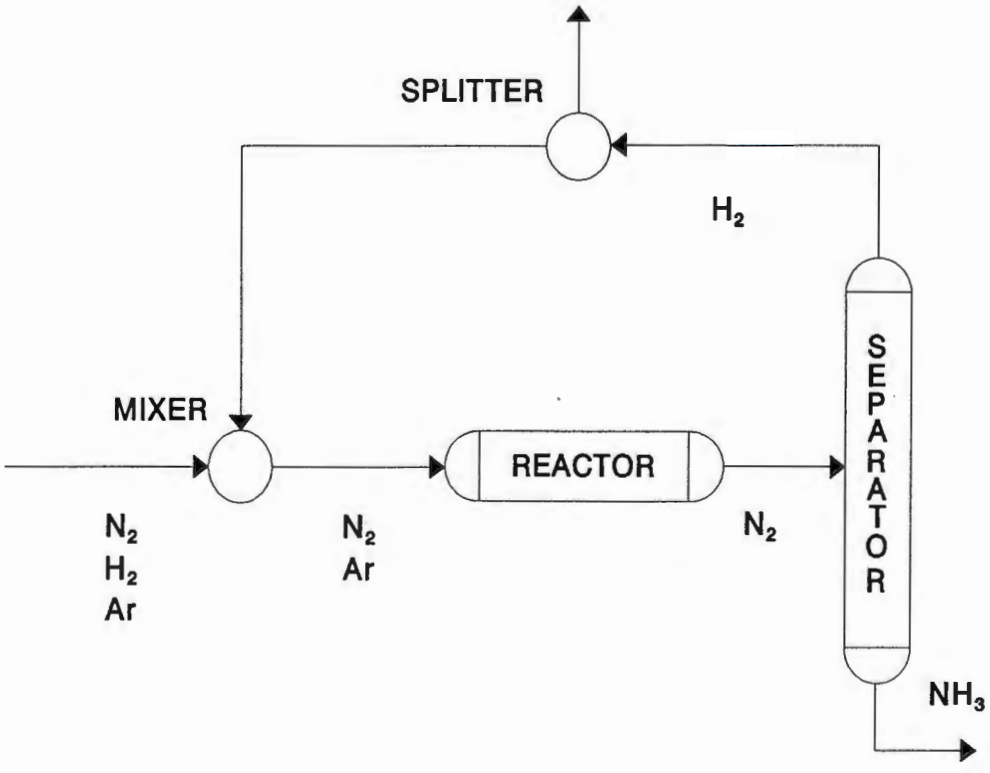


Figure 3.11 NH_3 synthesis flowsheet.

The flowsheet in Fig. 3.11 contains 8 measured variables and 13 unmeasured variables. Only the measured variables are shown in Fig. 3.11; more detailed information about this network and the m-files for this example are presented in Appendix D. In the original formulation of this flowsheet, correlations between the measurement variances were present (off diagonal elements in the variance matrix). In this presentation the variance matrix has been modified to contain only diagonal entries.

Table 3.4 Measured, reconciled and variance data for the NH_3 synthesis flowsheet.

	N_2^1	H_2^1	Ar^1	N_2^2	Ar^2	N_2^3	NH_3^4	H_2^5
y	33	89	0.4	101	20.2	69	62	205
x^*	31.79	95.29	0.402	99.98	20.08	69.59	60.795	204.4914
σ	0.82	6.34	1e-04	8.16	0.326	3.81	3.08	32

In Tables 3.4 and 3.5 the measurements are labelled with a superscript indicating their position in the network; for example, the nitrogen measurement in the first stream is labelled as N_2^1. Table 3.5 presents the sensitivity matrix for this network.

Table 3.5 Sensitivity matrix for the NH_3 synthesis example.

	σN_2^1	σH_2^1	σAr^1	σN_2^2	σAr^2	σN_2^3	σNH_3^4	σH_2^5
$x^* N_2^1$	-0.0129	0.0216	0	-0.0009	0	-0.0002	-0.0076	0
$x^* H_2^1$	-0.0129	0.0219	0	-0.0007	0	-0.0005	-0.0077	-0.0001
$x^* Ar^1$	0	0	0.0029	0	-0.0029	0	0	0
$x^* N_2^2$	-0.0034	0.0043	0	-0.0037	0	0.0043	-0.0015	0
$x^* Ar^2$	0	0	0.0029	0	-0.0029	0	0	0
$x^* N_2^3$	0.001	-0.0038	0	-0.0049	0	0.0064	0.0013	0
$x^* NH_3^4$	-0.0136	0.0228	0	-0.0007	0	-0.0005	-0.0081	0
$x^* H_2^5$	0.0006	0.0019	0	0	0	0	0.0004	-0.0029
1-Norm	0.0444	0.0762	0.0057	0.0109	0.0057	0.012	0.0266	0.0031
Rank	2	1	7	5	6	4	3	8

Analysis of Table 3.5 reveals that the system is most sensitive to the variance of hydrogen in the first stream and least sensitive to the variance of hydrogen in the fifth stream. To verify this result, the variances of the hydrogen measurements in the first and fifth streams were overestimated and the percentage change in the reconciled flowrates was calculated. The assumption was that the maximum error that one could make in assigning a variance would be $\pm$ 25 %. For this reason the measurement variances of the hydrogen measurements in the first and fifth where ranged over $\pm$ 25 % of their original values and the resulting percentage change in the reconciled values calculated.

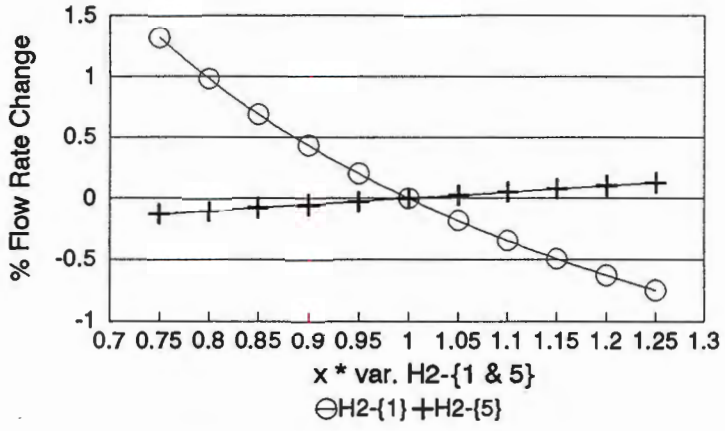


Figure 3.12 Effect of variations in the sensitivity of H_2^1 and H_2^5 on the system.

Figure 3.12 clearly shows a greater sensitivity to changes in the variance of the hydrogen measurement in the first stream than the fifth.

3.4.1. The Effect of Measurement Variances on the Measurement Test.

Statistical tests are applied to the reconciled data in order to identify measurements containing gross errors. Following the development in subsection 2.4.2. the measurement test statistic can be calculated as follows (Mah and Tamhane, 1982):

$$r^* = \Sigma^{-1} r$$

$$z_i = \frac{|r_i^*|}{\sqrt{\left[C^T \left[C \Sigma C^T\right]^{-1} C\right]_{ii}}} \qquad 3.11$$

The i^{th} measurement is suspected to contain a gross error if z_i exceeds a critical z_c.

$$z_i > z_c = z_{\frac{\beta}{2}}$$

where

$$\beta = 1 - (1 - \alpha)^{\frac{1}{n}} \qquad 3.12$$

Table 3.6 The Results of Varying the Measurement Variances of Hydrogen Measurement in the First and Fifth Streams.

H_2^1	z_i										
x * σ->	0.75	0.80	0.85	0.90	0.95	1.00	1.05	1.10	1.15	1.20	1.25
N_2^1	2.02	1.97	1.92	1.87	1.83	1.79	1.75	1.72	1.69	1.65	1.62
H_2^1	3.16	3.10	3.04	2.99	2.94	2.89	2.84	2.80	2.76	2.72	2.68
Ar^1	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
N_2^2	0.53	0.52	0.50	0.49	0.48	0.47	0.46	0.46	0.45	0.44	0.43
Ar^2	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
N_2^3	0.66	0.65	0.63	0.62	0.60	0.59	0.58	0.57	0.56	0.55	0.54
NH_3^4	1.29	1.24	1.18	1.13	1.09	1.04	1.00	0.96	0.92	0.89	0.85
H_2^5	3.16	3.10	3.04	2.99	2.94	2.89	2.84	2.80	2.76	2.72	2.68
H_2^5	z_i										
x * σ->	0.75	0.80	0.85	0.90	0.95	1.00	1.05	1.10	1.15	1.20	1.25
N_2^1	1.79	1.79	1.79	1.79	1.79	1.79	1.79	1.79	1.79	1.79	1.79
H_2^1	2.89	2.89	2.89	2.89	2.89	2.89	2.89	2.89	2.89	2.89	2.89
Ar^1	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
N_2^2	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47	0.47
Ar^2	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29	0.29
N_2^3	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59	0.59
NH_3^4	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04	1.04
H_2^5	2.89	2.89	2.89	2.89	2.89	2.89	2.89	2.89	2.89	2.89	2.89

From Eq. (3.11) it can be seen that the variance matrix features strongly in the gross error detection criterion. The following test was devised to evaluate the measurement test's sensitivity to measurement variances. The measurement test was applied to the NH_3 synthesis flow network. The measurement variances were varied over $\pm$ 25% for the most and the least sensitive variances as predicted by the sensitivity matrix, which corresponded to the hydrogen

measurements in the first and fifth streams respectively. The resulting z_i values of the measured variables are presented in Table 3.6.

Table 3.6 shows that changes in the variance of the measurement in the first stream cause a greater change in the z_i values than in the fifth stream. The critical value, $z_c = 2.73$ for this example. The middle column of Table 3.6 corresponds to the original variance data. It can be seen that initial measurements fail the measurement test for both the hydrogen measurements in streams one and five.

The measurements that failed the measurement test initially, pass when the variance of the hydrogen in the first stream is over-estimated by more that 20 %. By comparison, the values of the measurement test statistics remain unchanged when the variance of the hydrogen measurement in the fifth stream is adjusted by $\pm$ 25 %.

It is thus important to note that the errors in assigning a measurement variance can cause errors in the reconciled measurements, which may in turn be large enough to cause significant changes in the values of the measurement test statistic.

4. MEASUREMENT STRUCTURE SYNTHESIS

Measurement structure synthesis involves formulating the 'best' measurement structure for a process flowsheet subject to constraints imposed by a user. These constraints can take the form of naming certain variables as measured, controlling the amount of redundancy in the flow network and ensuring that specified unmeasured variables can be calculated. The aim is to produce measurement structures that have the properties of redundancy in the measurements and observability in the unmeasured variables, as set by the user. In this way optimum measurement structures can be developed upon which data reconciliation and gross error detection can readily be applied.

The proposed approach is in the form of two search algorithms. The first is a Global Search Algorithm which returns the global optimal solution of the measurement structure synthesis problem. The second is the Local Search Algorithm, whose solution is not guaranteed to be the global optimum; however a local solution is obtained in a fraction of the time required for the Global Search Algorithm to obtain a solution.

4.1. Global Search Algorithm

The Global Search Algorithm (GSA) is designed to search for the minimum set of measurements that satisfy the variable classification constraints. The search is initiated with a superstructure of measurements. All the variables are considered measured in this initial superstructure. The search algorithm then proceeds to remove all combinations of one,two, ..., measurements and tests the resulting measurement structure for constraint violations. The search is terminated when no more measurements can be removed without causing the resulting measurement structure to fail the variable classification constraints. The optimum measurement structure is defined as the minimum number of measurements that satisfy the variable classification constraints.

The term 'generation' is defined as the set of tests performed for the removal of every

combination of *n* measurements. For example: the second generation would contain the calculations performed in assessing the deletion of every possible combination of two measurements from the superstructure. The algorithm returns a set of measurement structures that represent the minimum number of measurements that satisfy the constraints. An exhaustive search is performed, and results in the global optimum being obtained, in terms of measurement structures that satisfy the constraints.

The algorithm starts with the removal of each measurement individually. If the removal of one measurement causes a violation of the constraints, it is removed from consideration and will not take part in the next generation. This is repeated until the effect of individually removing all the measurements has been evaluated. In the next generation every combination of two measurements (based upon the successful removal of measurements in the previous generation) are removed and the resulting measurement structures tested for constraint violations. The present generation bases its search on the preceding generation's results, and so on. This process is continued until no more measurement structures that satisfy the constraints can be found.

For example, if the measurement superstructure contains twenty-six measurements (*a,... z)* and the removal of measurements *a* and *b* cause the constraints to be violated, then when one tests the removal of every combination of three measurements, *a* and *b* are omitted from the search. This is due to the fact that if the removal of measurements *a* and *b* caused a violation of the constraints, then the measurement structures formed by the deletion of measurements *a, b* and any one in the set *c -> z* will also violate the constraints. In this way a number of measurement structures that have to be tested can be reduced.

The introduction of a cost function can also be used to target the search towards a specific measurement structure. In the examples considered the cost function was defined as unity for each measurement. In this way all possible measurement structures representing the minimum number of measurements that satisfied the constraints, are returned.

4.1.1. Restricting the search

This section will illustrate by means of an example how the previous 'good' measurement structures are used to define the search in the current generation.

Consider a measurement superstructure consisting of five measurements. Assume that the first generation failed to produce any 'bad' measurement structures. The second generation tests the removal of all combinations of two measurements. Table 4.1 presents all the possible combinations of removing two from five measurements.

Table 4.1 All possible combinations of removing two measurements.

Every combination of two out of five measurements			
1,2	{1,3}	1,4	1,5
	{2,3}	2,4	2,5
		3,4	3,5
			4,5

A matrix Γ is used to keep a record of 'bad' measurement structures. If the removal of the measurements indexed by {1,3} and {2,3} causes the constraints to be violated, then the Γ matrix will have the following form:

$$\Gamma = \begin{bmatrix} 1 & 2 \\ 3 & 3 \end{bmatrix} \qquad 4.1$$

In the third generation the removal of all combinations of three measurements will be tested, as shown Table 4.2. The measurement structures involved in the third generation are generated from the 'good' measurement structures found in the second generation. In the third generation the algorithm will try and remove one more measurement from each of the 'good' measurement structures of the second generation.

Consider the general case where (i,j) was evaluated as a 'good' measurement structure. The third generation will test all combinations of (i,j,k) where $k = j + 1$ to 5. In this manner the

algorithm builds up a set of measurement structures based on the 'good' results of the second generation. For example: from (1,2), the following measurement structures will be tested: (1,2,3), (1,2,4) and (1,2,5). Entries in the last column of Table 4.1 are disregarded because they are already accounted for by preceding entries. Consider the following combinations that could be formed from (1,5):

(1,5,2) already formed from (1,2) as (1,2,5), see above.

(1,5,3) would have been formed from (1,3), if (1,3) was not a 'bad' measurement structure.

(1,5,4) already formed from (1,4) as (1,4,5), see Table 4.2.

Table 4.2 shows all the possible measurement structures that can be evaluated in the third generation. The values in braces are not tested, as they contain subsets of 'bad' measurement structures. Theses entries would not normally be present, but are presented for accounting purposes.

Table 4.2 Measurement structures considered in third generation.

Measurement structures that could be tested in the third generation.					
1,2,3	1,2,4	1,2,5	{1,3,4}	{1,3,5}	1,4,5
			{2,3,4}	{2,3,5}	2,4,5
					3,4,5

A new set of measurement structures has been created, based on the results of the previous generation. In so doing it has been possible to reduce the size of the search set by four elements, (the braced measurement structures in Table 4.2). In general, entries that contain subsets of bad' measurement structures will still exist. In this example, (1,2,3) contains the subset (1,3), which has previously been specified as 'bad'. Integer cuts (Duran and Grossmann, 1986) were used to remove this measurement structure and others like it from consideration.

4.1.2. Integer cuts

Duran and Grossmann used integer cuts to help reduce the enumeration in a search for promising system configurations in process systems synthesis.

In the present study, integer cuts are used to avoid testing proposed measurement structures that contain subsets of measurements that have previously been labelled as 'bad'. The measurement structure being tested is described by an n vector, y_j, where n is the number of elements in the superstructure. If the measurement structure resulting from the removal of measurements i_1 and i_2 is being tested, then the characteristic y_j will consist of a column of zeros with unity in positions i_1 and i_2.

A record is kept of all previous 'bad' measurement structures. The indices of the measurements whose removal caused 'bad' measurement structures are placed in a column of the matrix Γ, with each newly found 'bad' measurement structure appending a column to Γ. As measurement structures with greater numbers of measurements are placed in Γ, zeros are appended to the existing columns of Γ. K and L will be used to denote the number of rows and columns in Γ.

Before a measurement structure is tested for constraint violation, it is first checked to see if it contains any subsets of previously found 'bad' measurement structures. The measurement structure under consideration is tested against all the previously found 'bad' structures and fails the test if the following criterion is not met.

$$\sum_{k=1}^{K} y_j(\Gamma(k,l)) \leq |\Gamma_l| - 1 \quad \forall\, l \in L \tag{4.2}$$

where Γ is a matrix of dimension (K,L), containing L previously evaluated 'bad' measurement structures and $|\Gamma_l|$ is the number of nonzero elements in the l^{th} column of Γ. Equation (4.2) sums the elements of y_j in the positions dictated by the column entries in Γ. Using this approach it can be evaluated if the measurement structure described in y_j contains a subset of 'bad' measurements. This test is applied to each column of Γ until it fails, in which case the present measurement structure is excluded from further analysis. If

the measurement structure passes this test, it is then further evaluated to see if all the constraints are satisfied.

In Table 4.2 there is one undetected measurement structure that contains a subset of 'bad' measurements, namely (1,2,3) containing subset (1,3). Integer cuts are used to remove this measurement structure from consideration. For the case of (1,2,3) the y_j vector has the following form: y_j = [1 1 1 0 0]. Using the integer criterion, Eq (4.2), it can be shown how the decision is made to remove this measurement structure from consideration:

$$\sum_{k=1}^{2} y_j(\Gamma(k,l)) = y_j(1) + y_j(3) = 1 + 1 = 2 \not\leq |\Gamma_l| - 1 = 2 - 1 = 1 \quad \text{for } l = 1 \qquad 4.3$$

The measurement structure fails to meet the integer cut criterion and is therefore removed from consideration. By comparison, the measurement structure (3,4,5), with y_j = [0 0 1 1 1], passes the integer cut criterion and moves on for further analysis:

$$\sum_{k=1}^{2} y_j(\Gamma(k,l)) = y_j(1) + y_j(3) = 0 + 1 = 1 \leq |\Gamma_l| - 1 = 2 - 1 = 1 \quad \text{for } l = 1$$

$$\sum_{k=1}^{2} y_j(\Gamma(k,l)) = y_j(2) + y_j(3) = 0 + 1 = 1 \leq |\Gamma_l| - 1 = 2 - 1 = 1 \quad \text{for } l = 2 \qquad 4.4$$

After applying the integer constraints to values in Table 4.2, the set of measurement structures to be evaluated in the third generation is obtained, shown in Table 4.3.

Table 4.3 The measurement structures to be considered in the third generation.

Measurement structures in the third generation.					
{1,2,3}	1,2,4	1,2,5	{1,3,4}	{1,3,5}	1,4,5
			{2,3,4}	{2,3,5}	2,4,5
					3,4,5

The measurement structures in braces in Table 4.3 represent those which have been removed from consideration. By using the 'good' measurement structures from the second generation

and integer cuts to guide the search, the number of measurement structures that need to be evaluated has been reduced from ten to five.

4.1.3. Variable constraints

A standard unconstrained search of all the possible measurement structures leads to a great number of calculations, even for flowsheets of medium size. The existence of a number of constraints placed by a user serves to focus the search and restricts the number of calculations performed.

The following variable classification constraints will be allowed:

1) A variable must be measured.
2) A measured variable must have a degree of redundancy of one. A measurement with a degree of redundancy equal to one retains its redundancy if any other one measurement is deleted.
3) A measured variable may be nonredundant.
4) A variable must be observable.

The implementation of constraints on specific variables is performed through an $(n+m,4)$ matrix *VAR*, where n and m are the number of measured and unmeasured variables respectively. This matrix consists mostly of zeros with ones indicating the placement of a specific constraints on the variables. Table 4.4 shows the typical structure of a *VAR* matrix.

Table 4.4 The variable constraint matrix, *VAR*.

Variable	Definite measurement	Degree of redundancy=1	Allow non-redundancy	Observable variable
1	1	1	0	0
2	1	0	0	0
3	1	0	1	0
4	0	0	0	1
5	0	0	0	0

Table 4.4, shows how the constraints can be combined. Variable one must be measured, and have a degree of redundancy of one. Variable two must be measured and redundant. Variable three must be measured but may be nonredundant. Variable four does not need to be measured but has to be observable. Variable five may be unmeasured and unobservable. These constraints provide total flexibility with regard to which variables need to be measured / unmeasured, redundant / nonredundant, or observable / unobservable. The implementation of these constraints, as they appear in the algorithm, will be presented in the following section.

4.1.3.1. Definition of a variable as measured

Variables that the user sets as definite measurements are simply excluded from the search. The GSA is prevented from removing these measurements in its search for the optimum measurement structure.

4.1.3.2. Redundant / nonredundant measurements

The redundancy of measurements can be evaluated from examination of the corresponding columns of the $\boldsymbol{C}$ matrix, formed during the reconciliation procedure using the QR decomposition of the matrix $\boldsymbol{B}$, and the $\boldsymbol{A}$ matrix, as discussed in subsection 2.1.4.

If a measurement is set to be redundant then its corresponding column in $\boldsymbol{C}$ is checked for a column of zeros. If a column of zeros is found it implies that the measurement in question is nonredundant and therefore the current measurement structure violates this constraint. This check is applied to every measurement except those which have been set, in the third column of $\boldsymbol{VAR}$, to be nonredundant. The result of this test is that a measurement structure that passes this test will not contain any nonredundant measurements other than those specified by the user.

4.1.3.3. Ensuring the observability of variables

The QR decomposition of the $\boldsymbol{B}$ matrix produces the two matrices $\boldsymbol{R_{11}}$ and $\boldsymbol{R_{12}}$; and in the calculation of the unmeasured variables, a variable is unobservable if it contains nonzero elements in the corresponding row of $(\boldsymbol{R_{11}})^{-1}\boldsymbol{R_{12}}$ or if they form part of the u_{m-r} vector, as

discussed in subsection 2.3.1. All variables constrained to be observable in ***VAR*** are tested for these properties.

4.1.3.4. Ensuring a specified degree of redundancy in a measurement

The degree of redundancy was previously defined as the ability of a measurement to retain its redundancy in the event of any measurement failures.

The test for a specified degree of redundancy uses the criterion established for the tests of redundancy. For the proposed measurement structure, failures in every other measurement are simulated. The corresponding columns of the new $\boldsymbol{C}$ are then tested for columns of zeros for all those measurements constrained to have a degree of redundancy of one.

This test is easily expanded to impose higher degrees of redundancy. For example if a degree of redundancy of two is specified, then failures would be simulated for every combination of two measurements and the resulting measurement structure tested for nonredundancies in the specified measurements.

4.1.4. Updating the Γ matrix and the terminating criterion

If a proposed measurement structure fails in any one of the constraints described in the previous section, then the measurement structure is labelled as 'bad' and is excluded from future consideration. This is achieved by integer cuts, as explained in a subsection 4.1.2. The indices of the unmeasured variables are appended in column form to Γ, and in so doing a record of failed measurement structures is generated. This prevents the future testing of measurement structures containing subsets of previously evaluated ones as being 'bad'.

Continuing the example of subsection 4.1.1: If during the third generation, the measurement structure defined by the removal of measurements (3,4,5) violates the constraints, its indices are appended in column form to Γ.

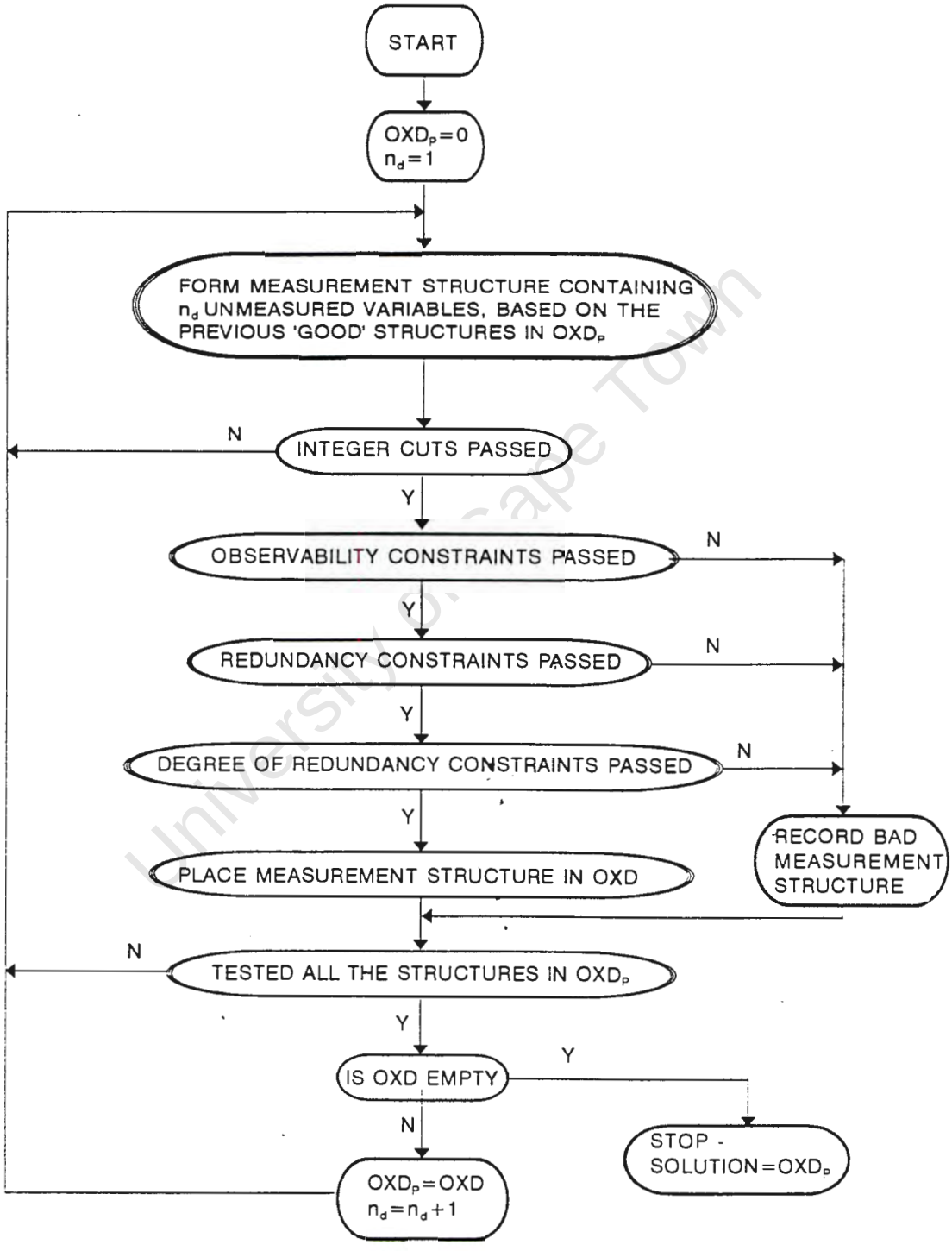
$$\Gamma = \begin{bmatrix} 1 & 2 & 3 \\ 3 & 3 & 4 \\ 0 & 0 & 5 \end{bmatrix} \qquad 4.5$$

The search is terminated when no more 'good' measurement structures are reported. The solution set is then the set of 'good' measurement structures in the previous generation. If a cost function for the measurements was employed it would be possible to choose a minimum cost measurement structure.

The cost function is not directly involved in the search algorithm because it may cause possible measurement structures to be overlooked. If one measurement is particularly expensive it may be removed at an early stage, but the result may be that two or more measurements, whose combined cost may be more expensive, may be needed to replace that measurement at a later stage of the search.

4.1.5. Flowsheet of the GSA

The algorithm upon which this flowsheet is based appears in Appendix E.



OXD_p=MATRIX OF PREVIOUSLY EVALUATED GOOD MEASUREMENT STRUCTURES

OXD=MATRIX OF CURRENT GOOD MEASUREMENT STRUCTURES

4.2. Local Search Algorithm

The Local Search Algorithm (LSA) presents a method to obtain a quick solution to the measurement structure synthesis problem since the GSA may become intractable for large problems due to the exhaustive combinatorial nature of the search. However, the LSA is not guaranteed to find the global optimum.

The LSA method was inspired by a procedure for measurement structure selection presented by Madron and Veverka (1992). They presented a method for the optimal selection of measurement points in process networks modelled by linear balance equations. This method is based on multiple Gauss-Jordan elimination of the system of linear model equations. Two objective functions that may be used in the optimization are proposed: the first is based on the overall cost of the measurements, while the second deals with the precision results.

Madron and Veverka presented the following classification of variables: measured and required; measured and not required; unmeasured and required (observable); and unmeasured and not required (unobservable).

After Gauss-Jordan elimination has been applied to a system of model equations the following results can be concluded:

1) Whether the system contains dependent equations.
2) If the system contains contradictory equations.
3) If all the required unmeasured variables are observable, and which unmeasured variables need to be measured to eliminate these unobservabilities.
4) Which measurements are redundant and nonredundant.

The optimization procedure starts with a suboptimal base structure. All measurement structures of *distance* one, are be found by manipulating the columns of the Gauss-Jordan reduced model equations matrix. The *distance* is defined as the difference between the number of distinct measurements of two measurement structures (Madron and Veverka,

1992). The objective function is evaluated for all the measurement structures of *distance* one, and the one with the minimum objective function value is exchanged with the base design. The optimization is continued until a local minimum is found. This approach finds a measurement structure that ensures that the observability constraints, on specified unmeasured variables, are met with the minimum number of measurements. However, it does not guarantee finding the global optimum measurement structure, nor does it address the question of degree of redundancy or enforce redundancy in the measurements.

The main difference between the LSA and the GSA is that instead of starting with a superstructure (containing all variables) the LSA begins with a random set of unmeasured variables. It then proceeds to convert some of these unmeasured variables into measurements to satisfy redundancy and degree of redundancy constraints. No adjustments for the observability constraints are made at this stage because the initial set of random unmeasured variables is chosen such that no unobservable variables are present. The LSA then proceeds in a similar fashion to the GSA in searching for the optimum measurement structure (subject to the variable classification constraints) by removing measurements. The measurements that were added to satisfy the constraints of redundancy and degree of redundancy are not considered for removal in the search for the optimum measurement structure, because their removal will cause the constraints (that caused their placement in the first place) to be violated. Only the initial measurements that were specified by the initial selection of unmeasured variables are operated upon.

The implementation of the LSA causes a reduction in the size of the measurement superstructure that needs to be explored. Instead of the measurement superstructure containing all the measurements it now contains only those measurements specified by the initial choice of unmeasured variables.

The solution is not necessarily the global optimum because the initial choice of unmeasured variables may restrict the LSA. This effect has been countered by multiple applications of the LSA with different starting points. As will be shown in later sections, the global solution is often found by multiple applications of the LSA.

4.2.1. Initial choice of unmeasured variables

The initial choice of the unmeasured variables is obtained from a random selection of variables that ensures that no unobservable variables are present. The basis for the selection stems from the discussion on variable classification in subsection 2.3.

There are two points that govern the choice of unmeasured variables:

1) Data reconciliation must be able to be performed on the resulting measurement structure. This is important because the purpose of developing measurement structure synthesis algorithms is the development measurement structures upon which data reconciliation can readily be applied.
2) The resulting set of unmeasured variables must contain no unobservable variables. This is used to ensure that the data reconciliation returns a unique result.

The first point implies that the resulting $\boldsymbol{C}$ matrix, formed from the $\boldsymbol{A}$ and the $\boldsymbol{Q_2}$ obtained from the QR decomposition of $\boldsymbol{B}$, must contain a minimum of one row. If this $\boldsymbol{C}$ matrix is empty the data reconciliation procedure yields the trivial result that the reconciled measurement values equal the measurement values. $\boldsymbol{C}$ is obtained from the premultiplication of the $\boldsymbol{A}$ by the $\boldsymbol{Q_2}^{\mathrm{T}}$. In order for $\boldsymbol{C}$ to contain one row $\boldsymbol{Q_2}$ must contain one column. Since $\boldsymbol{Q_2}$ is obtained from the QR decomposition of $\boldsymbol{B}$, the selection of unmeasured variables must be such that the correct form of $\boldsymbol{Q_2}$ is obtained.

The second point that no unobservable variables be present implies that there should be zero entries in the u_{m-r} vector or equivalently, the matrix $\boldsymbol{R_{12}}$ should not exist. Since $\boldsymbol{R_{11}}$ and $\boldsymbol{R_{12}}$ are formed from the QR decomposition of $\boldsymbol{B}$ and the column entries in $\boldsymbol{R_{12}}$ correspond to dependent columns in $\boldsymbol{B}$, the following rule is applied: the rank of the $\boldsymbol{B}$ matrix must equal the number of columns present. This prevents the presence of dependencies between the columns of $\boldsymbol{B}$ and ensures that the matrix $\boldsymbol{R_{12}}$ will be absent from the QR decomposition of $\boldsymbol{B}$.

The combination of these two points ensures that for each problem considered there exists a maximum number of unmeasured variables that can be assigned at this stage. The

implementation of these two points is further explained by revisiting the QR decomposition of $\boldsymbol{B}$.

$$B = QR = [Q_1 | Q_2] \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix} \qquad 4.6$$

$\boldsymbol{B}$ has dimension (z x m), where z and m are the number of balance equations and unmeasured variables respectively. $\boldsymbol{Q}$ therefore has dimension (z x z) while the dimension of $\boldsymbol{R}$ is (z x m). The first point specifies $\boldsymbol{Q_2}$ to have only one column and hence dimension of (z x 1). The resulting dimension of $\boldsymbol{Q_1}$ is thus (z x z - 1). $\boldsymbol{Q_1}$ operates on $\boldsymbol{R_{11}}$, therefore the dimension of R_{11} must be (z-1 x z-1). The second point specifies that $\boldsymbol{R_{12}}$ does not exist, therefore the number of columns in $\boldsymbol{R_{11}}$ must equal the number of unmeasured variables, m. Therefore the number of unmeasured variables initially selected, m, now equals z - 1.

The z - 1 variables are selected in the following manner: A random variable is selected and if it is not constrained to be measured and its addition to $\boldsymbol{B}$ does not cause the rank of $\boldsymbol{B}$ to be less than the number of columns in $\boldsymbol{B}$, it is placed in the initial set of unmeasured variables.

The absence of unobservable variables and the restriction of $\boldsymbol{C}$ to contain only one row ensures that data reconciliation can be applied to the initial measurement structure. The reconciliation will return a unique result as there are no unobservable measurements in the measurement structure.

4.2.2. Removal of nonredundant measurements

The measurement structure specified by the initial choice of unmeasured variables can now be analysed to identify nonredundant measurements. If these measurements are specified through variable constraints as being redundant, then some of the unmeasured variables can be converted to measurements to remove these nonredundancies.

If any nonredundant variable classification constraints are violated, the first unmeasured variable is converted into a measurement and the variable classification repeated. If the nonredundancies in any of the specified measurements are removed, the measurement is included in the measurement structure. If the addition of the measurement does not affect the nonredundancy of the specified measurements, it is reclassified as unmeasured and the effect of measuring the next unmeasured variable is evaluated. This process is continued until all the nonredundancies are removed or the list of unmeasured variables is exhausted. If after exhausting the list of unmeasured variables there are still nonredundancies present in the system, the initial choice of measurements can be classified as 'bad' and the LSA restarted. Although this is a possible scenario, it has never been encountered in the examples considered.

4.2.3. Enforcing the degree of redundancy constraints

After all the unwanted nonredundancies have been removed, the constraints that specify a degree of redundancy in certain measurements can, if they exist, be enforced.

The measurement structure is tested so see if any of the degree of redundancy constraints are violated and a list of all offending measurements made. The first unmeasured variable is converted into a measurement, and the resulting measurement structures tested for this constraint violation. The update to the measurement structure is accepted if the number of measurements that fail their degree of redundancy constraints diminishes. This is continued until no more measurements fail this constraint or all the unmeasured variables have been converted to measurements. In this way it is possible that all the unmeasured variables will be converted into measurements in order to meet these constraints.

4.2.4. Searching for the optimum measurement structure

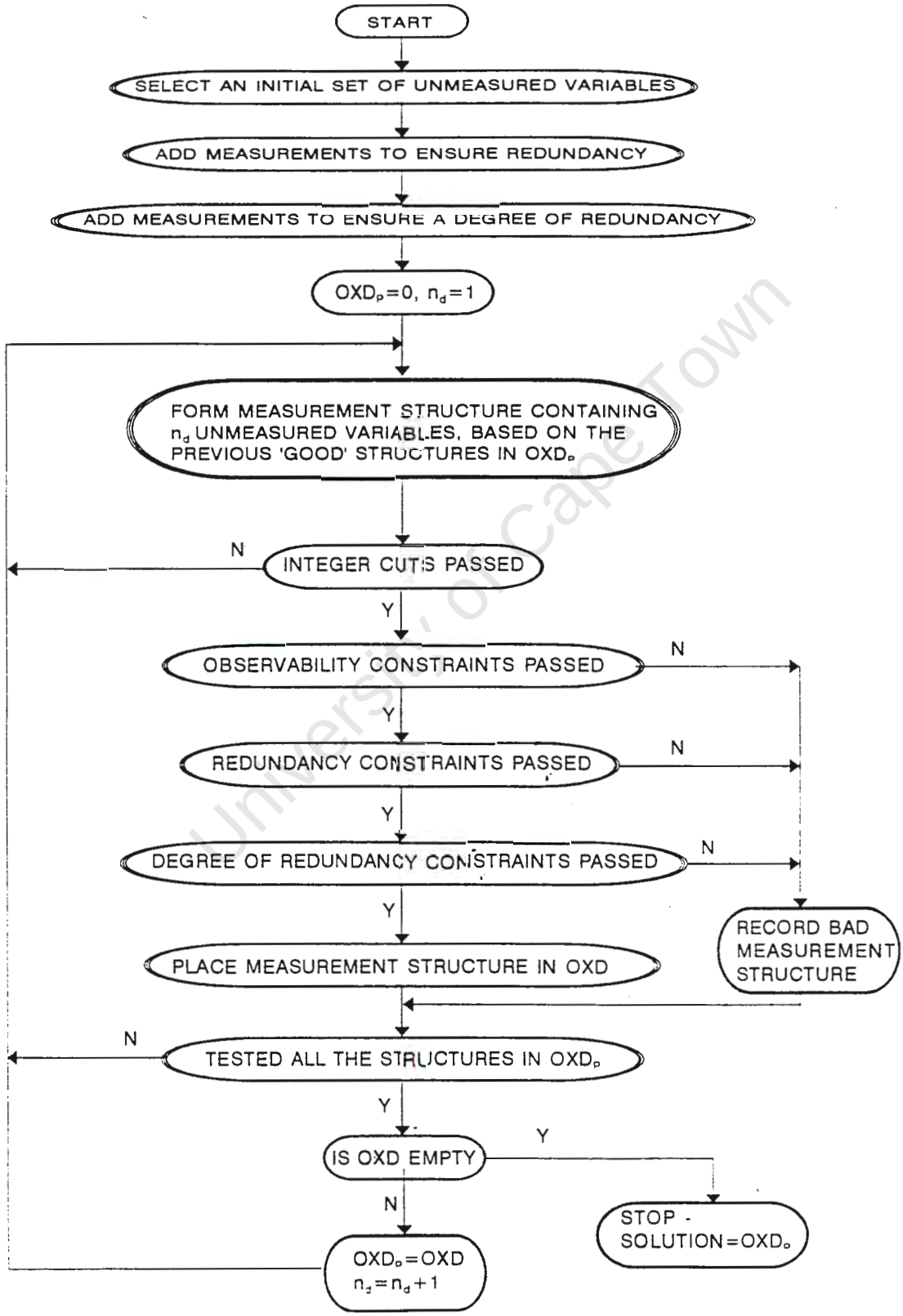
Now that all the variable classification constraints have been met the LSA can perform a search for the optimal measurement structure. This search is exactly the same as the GSA except the superstructure from which it can remove measurements has been reduced in size. This superstructure contains only the measurements present in the initial measurement structure. In effect the LSA is a modified version of the GSA starting possibly closer to the optimal solution.

From this point onwards the search employs all the devices of the GSA, with the next generation based on 'good' solutions in the previous generation and integer cuts used to restrict the search.

The LSA is based on the presumption that starting the search closer to the solution will return a reasonable 'good' local solution, if not the global solution, in a fraction of the time needed by the GSA. As will be shown in the examples, the GSA's execution time can be restrictive and quick good solutions can be obtained using the LSA.

4.2.5. Flowsheet of the LSA

The algorithm upon which this flowsheet is based appears in Appendix E.



4.3. Measurement Structure Synthesis Examples

The measurement structure synthesis algorithms have been applied to two different flowsheets; a steam flow network (Searth and Heenan, 1986), and a flow network from Madron and Veverka (1992). For measurement structure synthesis, the MATLAB m-files were developed on a PC and later converted, using MATCOM, to the 'C' source code and compiled on a SUN SPARCstation 20 using the GNU 'C' compiler. In this way the simplicity of the MATLAB script language was combined with the power and speed of a SUN SPARCstation 20.

4.3.1. Steam flow network

This flow network was introduced in subsection 3.3, as shown in Fig. 3.9. The variable constraint matrix ***VAR*** is shown in Table 4.5. From a possible twenty-eight measured variables, measurements four, seven, eight, nine, fifteen, seventeen, twenty-one, twenty-four, twenty-five, twenty-seven and twenty-eight are constrained to be measured with a degree of redundancy equal to one. The remaining variables may be measured or unmeasured. In addition, all measured variables are constrained to be redundant and all unmeasured variables are constrained to be observable.

Table 4.5 ***VAR*** for the steam flow network

	Definite measurement	Degree of redundancy = 1	Allow non-redundancy	Observable variable
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	1	1	0	0
5	0	0	0	0
6	0	0	0	0
7	1	1	0	0
8	1	1	0	0
9	1	1	0	0
10	0	0	0	0
11	0	0	0	0
12	0	0	0	0
13	0	0	0	0
14	0	0	0	0
15	1	1	0	0
16	0	0	0	0
17	1	1	0	0
18	0	0	0	0
19	0	0	0	0
20	0	0	0	0
21	1	1	0	0
22	0	0	0	0
23	0	0	0	0
24	1	1	0	0
25	1	1	0	0
26	0	0	0	0
27	1	1	0	0
28	1	1	0	0

4.3.1.1. Global Search Algorithm Solution

The GSA reported a solution time of 96.7 seconds. Twenty-four equivalent (in terms of the number of measurements) solutions were found and are shown in Table 4.6. An arbitrary selection of one the solutions is displayed in Fig. 4.1 which displays the resulting measurement structure together with all the redundant measurements, measurements with a degree of redundancy equal to one and unmeasured variables.

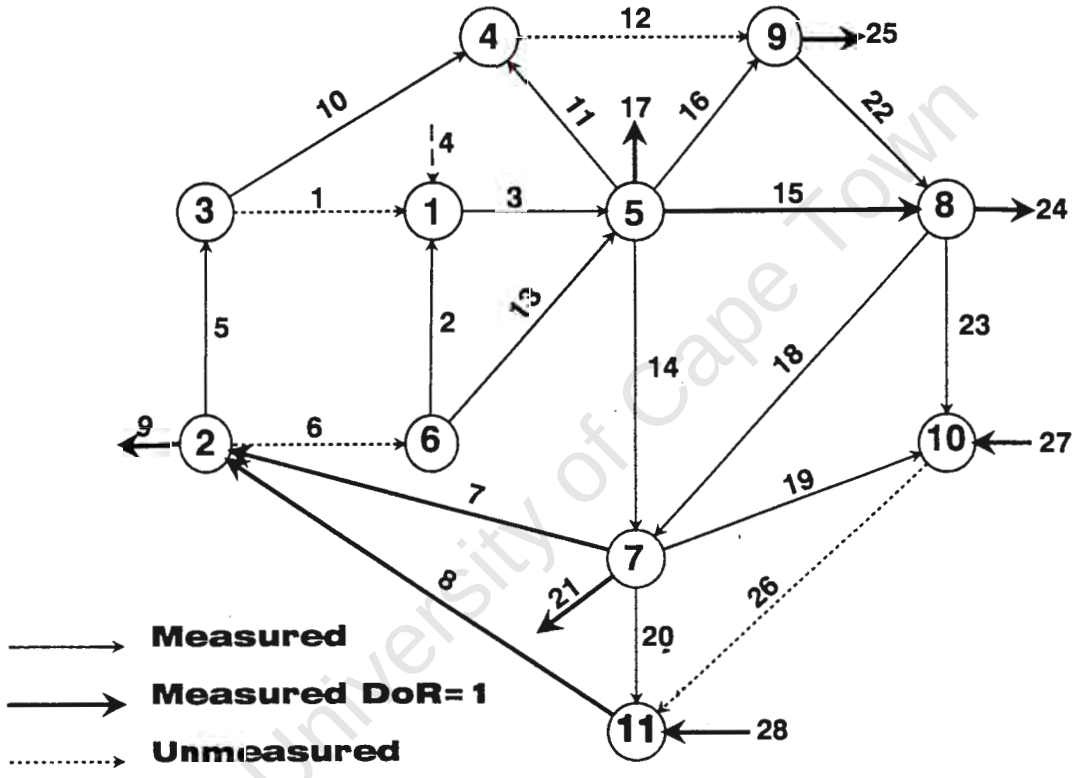


Figure 4.1 Optimal measurement structure for the network

From Fig. 4.1, it can be seen that all the variable constraints have been met. Using measurement seven as an example (constrained to be measured with a degree of redundancy of one), if measurement eighteen fails, measurement seven will not become nonredundant because eighteen is redundant, and can be calculated and its value used to ensure redundancy in measurement seven. This example may be repeated by removing any one of the other measurements and measurement seven will always remain redundant, and therefore has a degree of redundancy equal to one.

Table 4.6 GSA solution set for the steam flow network

Unmeasured Variables			
1	2	10	26
1	2	11	26
1	2	12	26
1	6	10	26
1	6	11	26
1	6	12	26
1	10	13	26
1	11	13	26
1	12	13	26
2	5	10	26
2	5	11	26
2	5	12	26
2	10	11	26
2	10	12	26
5	6	10	26
5	6	11	26
5	6	12	26
5	10	13	26
5	11	13	26
5	12	13	26
6	10	11	26
6	10	12	26
10	11	13	26
10	12	13	26

It should be noted that it is the existence of the constraints advocating redundancy that cause the greatest reduction in the number of searches performed. In the case where there are very

few redundancy constraints, the number of searches and hence the time taken increases significantly, as will be shown in subsection 4.3.2.

4.3.1.2. Local Search Algorithm Solution

The LSA was also applied to the steam metering problem. In order to find the global solutions (shown in Table 4.6) the LSA was applied to the problem ten times with each run starting the LSA with a different set of initial unmeasured variables. A summary of the results appears in Table 4.7. The ten different starting points are presented in a table in Appendix E.

Table 4.7 is arranged in the following way: the ten rows represent ten runs started with different initial sets of unmeasured variables; the first four columns represent the unmeasured variables of the ten solutions; the fifth column displays the number of equivalent solution sets that where found for each run and the last column reports the time taken to reach a solution.

As was shown in Table 4.6, there are twenty-four equivalent solutions. The LSA can find more than one of the solutions during each run, but for simplicity only the last of these is presented.

Table 4.7 Summary of the LSA results

Run #	Unmeasured Variables				Soln. #	TIME [s]
1	6	10	12	26	1	12.8
2	12	13			2	6.5
3	1	13			2	7.4
4	5	10	26		1	8.1
5	1	2	12		2	9.9
6	6	10	11	26	2	16.6
7	10	12	13	26	3	19.8
8	1	2	10		1	8.1
9	10	13	26		2	11.0
10	10	11	26		3	11.0

The LSA was successful in finding some of the twenty-four equivalent solutions in Table 4.7 in three out of the ten runs. The time per run is reduced from 96.7 seconds for the GSA to 19.8 seconds for the longest LSA solution. While it may be argued that the combined CPU time for ten runs of the LSA exceeds that for the GSA, it will be shown in the following example and in the industrial case study, that this is not always the case.

4.3.2. Madron flow network

A flowsheet used by Madron and Veverka (1992) to illustrate their approach for the optimal selection of measuring points in plants governed by linear models is displayed in Fig. 4.2. The network is defined in Appendix F.

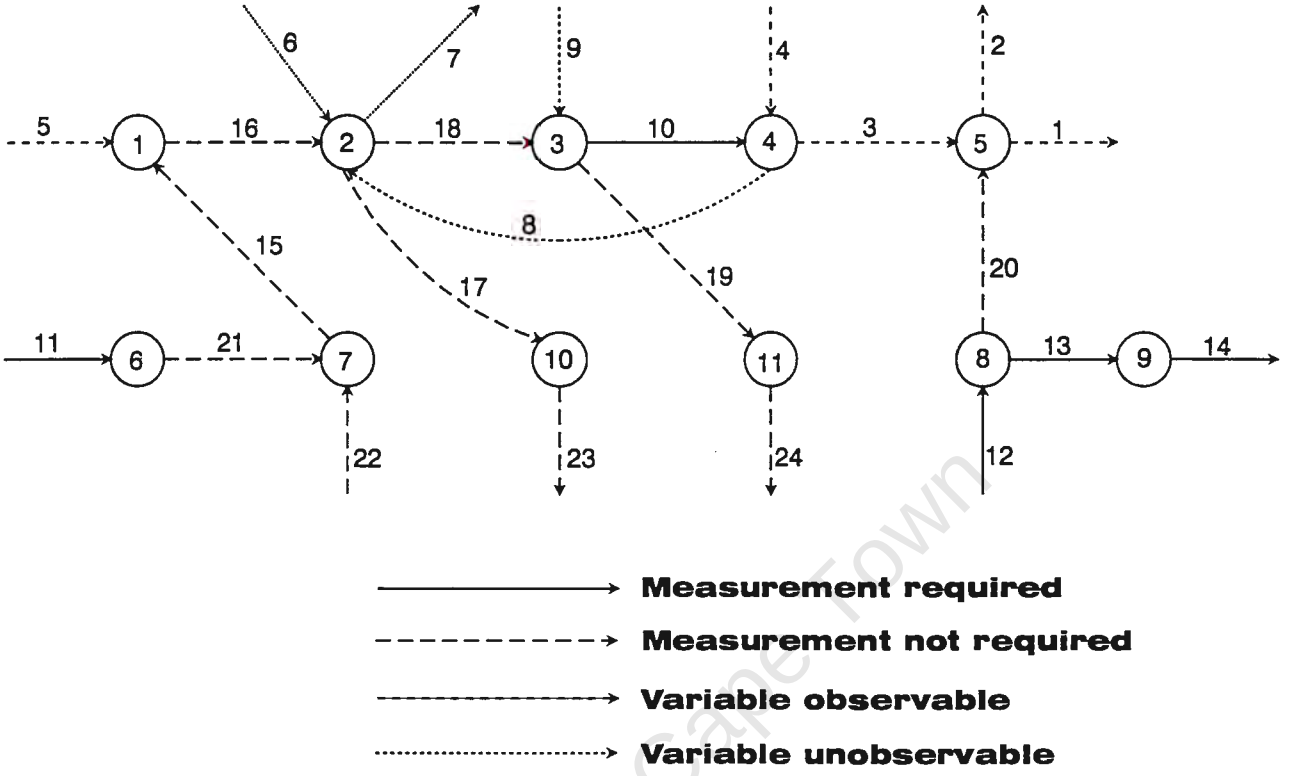


Figure 4.2 Madron flow network with all variable constraints displayed

The variable constraints specified in the original paper were duplicated. In addition it was specified that all the required measurements (ten to fourteen) be redundant, with the remaining measurements allowed to be nonredundant. The variable constraint matrix for the Madron network is shown in Table 4.8. The solution obtained by Madron does not represent a global optimum, but only a good, feasible solution (Madron and Veverka, 1992).

Table 4.8 ***VAR*** matrix for the Madron network

Measurement	Definite measurement	Degree of redundancy = 1	Allow non-redundancy	Observable variable
1	0	0	1	1
2	0	0	1	1
3	0	0	1	1
4	0	0	1	1
5	0	0	1	1
6	0	0	1	0
7	0	0	1	0
8	0	0	1	0
9	0	0	1	0
10	1	0	0	0
11	1	0	0	0
12	1	0	0	0
13	1	0	0	0
14	1	0	0	0
15	0	0	1	0
16	0	0	1	0
17	0	0	1	0
18	0	0	1	0
19	0	0	1	0
20	0	0	1	0
21	0	0	1	0
22	0	0	1	0
23	0	0	1	0
24	0	0	1	0

4.3.2.1. Global Search Algorithm Solution

The GSA returned one solution after 31 minutes. The solution consisted of removing thirteen from the original twenty-four measurements, as shown in Table 4.9. The solution reported by Madron and Veverka (1992) is shown in Table 4.10.

Table 4.9 GSA solution for the Madron example

Measurements to be removed from the superstructure												
3	6	7	9	15	16	17	18	19	20	22	23	24

Table 4.10 Madron solution set

Measurements to be removed from the superstructure										
1	3	5	6	7	9	17	18	19	23	24

The C matrix, for the GSA's solution, is shown in Table 4.11. From this, it can be seen that only one column of zeros exists, therefore only measurement five is non-redundant, while measurements ten to fourteen are redundant, as specified in the variable constraints.

Table 4.11 C matrix showing positions of non-redundant measurement (5)

1	2	4	5	8	10	11	12	13	14	21
0.11	0.11	-0.11	0	0.11	-0.11	-0.98	0.11	-0.11	0	0.98
0.57	0.57	-0.57	0	0.57	-0.57	0.19	0.57	-0.56	0	-0.18
0	0	0	0	0	0	0	0	1	-1	0

The matrix that determines the observability / unobservability of the unmeasured variables is shown for the algorithm's solution in Table 4.12. It can be seen that the row that relates to variable three contains no non-zero entries, and therefore is observable. Therefore, the GSA's solution satisfies all the variable classification constraints.

Table 4.12 Table of $(R_{11})^{-1}(R_{12})$

Variables	22	9	23	6	24
3	0	0	0	0	0
15	-1	0	0	0	0
17	0	0	-1	0	0
19	0	0	0	0	-1
20	0	0	0	0	0
16	-1	0	0	0	0
18	0	1	0	0	-1
7	-1	-1	1	-1	1

The GSA returns a superior solution (in terms of a smaller measurement set), while meeting the variable classification constraints.

In reaching a solution, the GSA finds the minimum measurement set by removing measurements from the superstructure. Essentially there are three groups of variables from which elements can be removed. They are: variables that are not required to be measured, variables that must be observable and variables that may be unobservable. Table 4.13 displays how many measurements in each of these classes are removed by both solution techniques.

Table 4.13 Breakdown of.the various variable classifications in the solution sets for both the Madron and GSA approaches.

Variable classification	Madron	GSA
Measured not required (10)	5	9
Unmeasured observable (5)	3	1
Unmeasured unobservable (5)	3	3
Total measurements removed (20)	11	13

The number in brackets in the first column of Table 4.13 indicates the number of measurements or unmeasured variables in each class of the variable classification constraints. The GSA removed more of the non-required measurements, and although it removed fewer of the measurements corresponding to the unmeasured observable variables, it still improved on Madron's solution. In order to meet the variable classification constraints, some of the nonrequired measurements were retained as measurements, and a few of the unmeasured variables were converted into measurements.

As was mentioned in the previous section, the existence of redundancy constraints contributes greatly to the reduction of the search. The Madron network has fewer streams than the Searth and Heenan example and it would be expected that with fewer streams the solution time would decrease. However, the solution time increases by twenty fold due to the virtual removal, save five measurements, of the constraints that enforce redundancy. Redundancy is a good property in a flow network, especially if gross error detection is going to be applied. The serial elimination techniques used in combination with statistical tests remove combinations of measurements from the measurement structure and test the remaining measurements. The statistical tests are based on the residuals of the reconciliation procedure. If a measurement is nonredundant, or becomes nonredundant due to the elimination of any other measurement being tested, it passes through the reconciliation procedure unaffected. Nonredundant measurements have zero values for their test statistics and hence are positions at which gross errors can exist undetected.

4.3.2.2. Local Search Algorithm Solution

The LSA was applied to the Madron problem and Table 4.14 presents the results obtained. As in the steam metering problem, ten runs were performed. The initial sets of unmeasured variables may be found in Appendix E.

Table 4.14 Summary of the LSA results for the Madron problem

Run #	Unmeasured Variables													Time [s]
1	3	15	16	18	19	20	21	23						0.4
2	3	5	6	7	9	17	18	19	20	22	23	24		0.7
3	3	6	7	9	15	16	17	18	19	20	22	23	24	1.6
4	3	6	7	9	15	16	17	18	19	20	22	23	24	1.2
5	3	5	6	7	9	17	18	19	20	22	23	24		1.0
6	3	6	7	9	15	17	18	19	20	21	23	24		0.9
7	3	6	7	9	15	16	17	18	19	20	22	23	24	1.6
8	3	6	7	9	15	17	18	19	20	21	23	24		1.2
9	4	15	16	18	19	20	21	23						0.3
10	8	9	15	16	19	20	21	23	24					0.4

The results in Table 4.14 show that the LSA obtained the global solution (found using the GSA) in three out of the ten runs. In addition seven of the ten solutions are superior to the solution reported by Madron and Veverka (1992). The inability of the LSA to guarantee a global solution is also displayed in the results. The most remarkable result is the difference in time taken to reach a solution; the GSA required 31 minutes while an equivalent solution is found by the LSA in 1.6 seconds. The results illustrate that it is possible to obtain a 'good' solution close (and even sometimes equal) to the global optimum in a fraction of the time using the LSA.

5. GENETIC ALGORITHMS AND GROSS ERROR DETECTION

This section presents the results of an investigation into the use of Genetic Algorithms (GAs) in gross error detection. Present methods combine statistical tests with serial elimination search algorithms for gross error detection (Searth and Heenan, 1986; Rosenberg *et al.*, 1987). The proposal is to replace the serial elimination techniques with an optimization approach. Since the statistical tests that form the objective function of the optimization are expected to be nonconvex, a global optimization strategy will need to be implemented.

The power of GAs comes from the fact that the technique is robust, and can deal successfully with a wide range of problem areas. Genetic Algorithms are not guaranteed to find the global optimum, but they are generally good at finding 'acceptably good' solutions to problems 'quickly'. Where specialized techniques exist for solving particular problems, they are likely to out-perform Genetic Algorithms in both speed and accuracy. The main ground for Genetic Algorithms, then, is in difficult areas where no such techniques exist. Even where existing techniques work well, improvements have been made by hybridizing them with a Genetic Algorithm (Beasley *et al.*, 1993). This section presents the results of an investigation into the effect of using Genetic Algorithms in gross error detection.

A simple GA dealing with a minimization on an irregular surface will be presented as an introduction, followed by a performance comparison of GAs using the recursive gross error detection method (Crowe, 1988), the Global test (Reilly and Carpani, 1963; Madron *et al.*, 1977; Almasy and Szatno, 1982) and a simple random search for gross errors.

The Genetic Algorithm TOOLBOX for MATLAB provided the computational tools for this investigation. It was developed by Andrew Chipperfield, Peter Fleming, Hartmut Pohlheim and Carlos Fonseca of the Department of Automatic Control and Systems Engineering at the University of Sheffield. This TOOLBOX is in the development stage and is intended to form part of the suite of TOOLBOXES distributed by MATHWORKS, the producers of MATLAB. The manual accompanying the TOOLBOX provides an excellent reference to

Genetic Algorithms.

5.1. An Overview of Genetic Algorithms

A Genetic Algorithm (GA) is a stochastic global search method that is based on natural biological evolution. The basic principles of GAs were first laid down rigorously by Holland, (1975) and are well described in many texts, of which an excellent review article is given by Beasley *et al.* (1993).

GAs operate on a population of potential solutions by applying the principle of survival of the fittest. The fitness of individuals in a population is evaluated by comparing the value of the corresponding objective function with those of the remainder of the population. At each generation, individuals from the population are selected based on their fitness, and bred together using operators borrowed from natural genetics. The offspring are hoped to contain the best parts of their parents, resulting in even fitter individuals. This process leads to the evolution of populations of individuals that are better suited to their environment than their parents.

Individuals are encoded as chromosomes (Holland, 1975) composed over some alphabet such that the genotypes (chromosome values) are uniquely mapped onto decision variables, the phenotypes. The most commonly used representation in GAs is that of binary numbers. A chromosome containing two variables represented to the same levels of accuracy may have the following structure.

$$\begin{array}{c|c} 101011 & 101110 \\ x_1 & x_2 \end{array}$$

In the above representation, x_1 and x_2 are both encoded with 6 bits.

Once the chromosomes have been decoded, it is possible to assess the performance of the individual members of a population through an objective function. In the natural world, this

would be an individual's ability to survive in its environment.

Before reproduction, each individual is assigned a fitness value derived from its performance which is evaluated by the objective function. Highly fit individuals, relative to the whole population, have a high probability of being selected for mating.

During the reproduction stage, highly fit individuals are selected from the population and recombined to produce the next generation. Genetic operators manipulate the genes of the chromosomes directly on the assumption that the combination of certain individuals' gene codes, will on average, produce fitter individuals. A recombination operator is used to exchange genetic information. Single point crossover is the simplest recombination operator. As an example, consider the following two parent binary strings.

$$P_1 = 10010110$$

$$P_2 = 10111000$$

A position, i, is selected uniformly at random in the range, $[1, L_{ind}\text{-}1]$, where L_{ind} is the length of the string. The genetic information is then exchanged between the two individuals about the point i. The offspring that are produced from the parents above, with $i=5$, have the following form:

$$O_1 = 10010000$$

$$O_2 = 10111110$$

This crossover is performed with a probability P_x when pairs are chosen for breeding. Another genetic operator, mutation, is also applied to the individuals with a probability P_m. In a binary string representation, mutation will cause a single bit, allele, to change its state. For example, mutating the fourth bit of O_1 leads to the new string O_{1m}.

$$O_{1m} = 10000000$$

Mutation is generally considered a background operator that ensures that the probability of searching a particular subspace of the problem is never zero. In addition, mutation tends to inhibit convergence to a local optimum.

This process is repeated until the maximum number of generations, set by the user, or a convergence criterion is reached. In this way, the average performance of the individuals in the population is expected to increase.

There are four significant differences between GAs and traditional optimization methods:

- GAs search a population of points in parallel, not as a single point.
- GAs require no derivative information, only the objective function, and corresponding fitness levels influence the direction of the search.
- GAs use probabilistic transition rules, not deterministic ones.
- GAs operate on an encoded parameter set, rather than the parameter set itself.

It is important to note that GAs provide a number of potential solutions (the final population), and the choice is left to the user.

5.2. Major Elements of a Genetic Algorithm

5.2.1. Population representation and initialization

GAs operate on a number of potential solutions, called a population. Populations are typically composed of between thirty and one hundred individuals. The most common representation of chromosomes in GAs is the binary string. Each decision variable is encoded as a binary string, and these are concatenated to form a chromosome.

The first step in a GA is to create the initial population. Usually, the required number of individuals are generated using a random number generator that uniformly distributes numbers in a desired range.

5.2.2. The objective and fitness functions

The objective function is used to provide a measure on an individual's performance. In the case of a minimization problem the fittest individuals will have the lowest corresponding objective function values. Objective functions must be created by the user.

A fitness function is usually used to transform the objective function value into a relative measure of fitness,

$$F_i(x) = g(f_i(x)) \tag{5.1}$$

where f_i is the objective function and g transforms the value of the objective function into a nonnegative fitness, F_i. In minimization problems this type of mapping is always necessary as the lower values of the objective function correspond to fitter individuals.

5.2.3. Selection - Stochastic Universal Sampling

Selection is the process of determining which individuals are chosen for reproduction. Efficient selection methods are needed to reduce the GA's execution time.

The time used by the other sections of a GA (excluding evaluation of the objective function) are of the order of (L_{ind} . N_{ind}). This is defined as time complexity, where L_{ind} and N_{ind} are the length and number of individuals. A selection algorithm should not contribute to increasing the time complexity of the GA.

Stochastic Universal Sampling (SUS) is a single phase sampling algorithm. SUS uses N_s equally spaced pointers, where N_s is the number of individuals required for breeding. The individuals are then chosen by generating N_s pointers and those individuals whose cumulative fitness spans the position of these pointers are selected. The sequence of pointers, p_r is obtained from the following formula:

$$p_r = \frac{S_F}{N_s}\left[\rho + [0,\cdots,N_s - 1]\right]$$

where

$$S_F = \sum_{i=1}^{N_{ind}} F_i \tag{5.2}$$

ρ = random number in the range [0 1]

The time complexity of SUS is of the order of N_s, therefore the time for execution of SUS is proportional to the number of individuals required for breeding.

The following example illustrates the SUS technique. A population of eight individuals, with their corresponding fitness values, are shown in the first two columns of Table 5.1. The cumulative fitness S_F, and the pointers p_r, are presented in the third and forth columns of Table 5.1. In this example, individuals 1, 2, 3, 4, 6, and 8 are selected by the SUS routine. In the cases where more than one individual lies between pointers, only the first is selected.

Table 5.1 Example of Stochastic Universal Sampling.

i	F_i	S_F	p_r
			1.0125
1	1.5	1.5	
			2.3408
2	1.35	2.85	
			3.6691
3	1.21	4.06	
			4.9975
4	1.07	5.13	
5	0.92	6.05	
			6.3258
6	0.78	6.83	
7	0.64	7.47	
			7.6541
8	0.50	7.97	

5.2.4. Single point crossover

The basic operator for the production of new chromosomes in GAs is crossover. Crossover produces new individuals that contain some parts of both parents' genetic material. The simplest form of crossover is that of single point crossover, described in the overview of GAs.

5.2.5. Mutation

In nature, mutation is a random process where one allele of a gene is replaced by another, producing a new genetic structure. In GAs mutation is randomly applied with a low probability, in the range [0.001 0.01]. Mutation is considered a background operator and is used to ensure that the probability of searching the entire is space is never zero.

5.2.6. Reinsertion

Once a new population has been produced and recombined with the individuals from the old population, the fitness values of the individuals may be evaluated. If fewer individuals are produced by recombination than the size of the original population, then a fractional difference between the old and the new populations is known as the generation gap.

To maintain the size of the original population, the new individuals have to be inserted into the old population. Similarly, if not all the new individuals are to be used or if more offspring than the original population size is produced, then a reinsertion scheme must be used to determine which individuals are to exist in the new population.

5.2.7. Termination of the Genetic Algorithm

It is difficult to formally specify a convergence criterion for GAs. The fitness of a population may remain static for a number of generations before a superior individual is found. Common practice is to terminate the GA after a prespecified number of generations and then test the final population for a solution. If no acceptable solutions are found the GA may be restarted or a new search initiated.

5.3. Example Problem

In this section the problem of finding the minimum of the MATLAB 'peaks' function is considered. Figure 5.1 displays the three dimensional graph of this function, defined by the following equation.

$$z = 3(1-x)^2\exp(-x^2-(y+1)^2) - 10(\frac{x}{5}-x^3-y^5)\exp(-x^2-y^2) - \frac{1}{3}\exp(-(x+1)^2-y^2) \quad 5.3$$

The GA was solved using the Genetic Algorithm Toolbox for MATLAB. The m-files and a table of the results are to be found in Appendix G. The maximum number of generations was set to fifty.

An initial population size of thirty individuals is used. A study of one hundred test cases was performed. In each test case the initial population is different. The results obtained are reasonable, with the solution (x=0.2903, y=-1.6452, z=-6.5048) found in seventy-four percent of the test cases. Figure 5.1 shows the initial and final population from the solution for one of the test cases of the simple GA. The crosses and circles represent the initial and final populations respectively. The minimum point can clearly be seen on Figure 5.1.

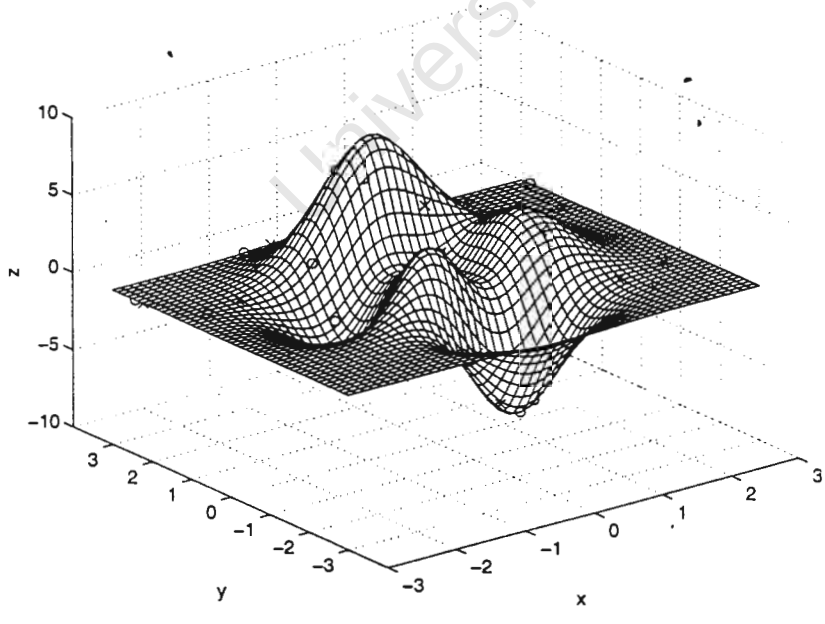


Figure 5.1 The 'Peaks' surface with the initial and solution populations of a GA with a population size of thirty individuals.

5.4. GAs for Gross Error Detection

In the following section, GAs were used as a replacement for the serial elimination techniques to minimize the objective functions (statistical tests) in the search for gross errors. Two statistical tests are evaluated for use as objective functions. The global test statistic and recursive gross error detection method are applied to a flow network with simulated gross errors. The flow network presented by Serth and Heenan (1985) shown in Fig. 3.9 was used.

5.4.1. Objective functions

Two statistical tests, the global test statistic (Reilly and Carpani, 1963; Madron *et al.*, 1977; Almasy and Szatno, 1982) and the recursive gross error detection method (Crowe, 1988) were compared as choices of the GA's objective function. Two gross errors were simulated in the network and the pair of measurements that minimized the objective function were sought. All the variables of the network were considered as measured.

These two statistics were chosen because they return a single value from which a decision can be made as to whether the gross errors have been identified. The two objective functions differ in that, while the global test requires the generation of the $\boldsymbol{C}$ matrix from QR decomposition, the recursive technique used the original balance equations $\boldsymbol{H}$ to calculate the statistic.

5.4.1.1. Global test statistic

For each set of measurements that are tested for gross errors, the suspect set of measurements is removed and the projected constraint matrix $\boldsymbol{C}$ formed. The global test statistic (Reilly and Carpani, 1963; Madron *et al.*, 1977; Almasy and Szatno, 1982), τ_{GT}, is then calculated.

$$\tau_{GT} = w^T\left[C\Sigma C^T\right]^{-1}w$$
$$\text{where} \quad w = Cy - d \tag{5.4}$$

where C and d are obtained from the constraint equations, and C after QR decomposition in

accordance with the method presented in subsection 2.1.4.

5.4.1.2. Recursive gross error detection

Crowe (1988) states that the value of the least squares objective function, J at the solution is

$$J = r_b^T V^{-1} r_b = (Hy)^T \left[H\Sigma H^T\right]^{-1} Hy \qquad 5.5$$

where $\boldsymbol{H}$ represents the constraint equations, and y the vector of measurements. Furthermore, it has a chi-square distribution (Crowe, 1988). The removal of the correct set of measurements containing the gross errors would cause the statistic to be below the threshold for the chi square with the appropriate confidence level and number of degrees of freedom. This reduction in the objective function can be calculated by the following equation, Eq. (5.6), without performing the corresponding QR decomposition of $\boldsymbol{H}$.

$$\Delta J = -(r_b^T V^{-1} B) G_l (B^T V^{-1} r_b)$$

where

$$G_l = B^T V B \qquad 5.6$$

$$V = H\Sigma H^T$$

The matrix $\boldsymbol{B}$ contains the columns of $\boldsymbol{H}$ corresponding to the measurements that are being removed. This approach is limited to the case where the balance equations are linear in the measurements.

5.4.2. Simulation and Detection of Gross Errors

Two gross errors were simulated in the network. The magnitudes of the gross errors were allowed to vary randomly between ten and one hundred percent of the respective flowrates. An initial population of thirty individuals was chosen and the maximum number of generations set to six.

An individual in the population consists of a pair of measurement indices. These entries in the individual are defined as two integers in the range [1 28]. The objective function evaluates the test statistic (global or recursive). Based on the individual's objective function value it may be chosen for reproduction with another. The off-spring are reinserted into the population forming a new generation.

From the final population, the individual with the minimum objective function value was chosen as the solution.

It is possible to determine the number of calculations that would represent searching the entire subspace, i.e. all possible positions of the gross errors by using the formulae below. The number of ways of selecting r objects from n distinct objects (Miller and Freund, 1985):

$$\binom{n}{r} = \frac{n!}{r!(n - r)!} \qquad 5.7$$

There are a total of twenty-eight measurements in the flow network therefore, to test the entire subspace (all combinations of selecting two measurements out of the twenty-eight) would represent three hundred and seventy-eight calculations. With the chosen population size and number of generations, one hundred and eighty calculations would be performed by the GA. This represents forty-eight percent of the possible calculations.

One hundred test cases were performed and for each case the position and size of the gross errors was changed at random.

In addition, a purely random search for the gross errors was performed. The number of tests performed in the random search was matched to the same number of tests performed by the competing GAs. Tables of results and m-files are to be found in Appendix H. Table 5.2 summarises the results obtained.

Table 5.2 Results of the one hundred gross error detection test cases, using GAs and a random search.

Method	Objective Function	Time [s]	Success Rate [%]
GA	Recursive Method	270	20
GA	Global	1655	3
Random Search	Recursive Method	126	22

The results displayed in Table 5.2 indicated the poor success rate of the GAs in gross error detection. The most successful GA method performance approximately equals the performance of a random search. The time differential between the GAs using the recursive method and global test statistic for the objective function is a result of the recursive method predicting the reduction in the objective function; while the global test requires that the measurements actually be removed and the projection matrix $\boldsymbol{C}$ formed before the statistics of the resulting measurement structure can be evaluated.

Due to the poor performance of the GAs it was decided to investigate the complete surface upon which these algorithms search for a minimum. The surface depends on the function being used as the objective function in the GA. In the example considered the recursive gross error detection method and the global test statistic are used to evaluate the effect of removing two measurements from the measurement structure. The removal of the correct pair of measurements, those containing the gross errors, is assumed to cause the greatest reduction in the objective function and thus be the reported minimum of the objective function.

Two cases are presented; one where the GA correctly identified the gross error and the other where it failed. In the case where the gross error was placed at positions five and twenty-one the GA correctly identified the gross error; while when the error was placed at positions one and seventeen the GA failed to identify the gross errors correctly. Table 5.3 shows the size of the measurements and gross errors.

Table 5.3 Characteristics of the induced gross errors.

	GA failed to identify the Gross Error			GA succeeded in identifying the Gross Error	
Position	1	17		5	21
Gross Error size	480	33340		32392	17508
Measurement size	1354	67099		86812	107987
% GE of meas. size	35	49		37	16

Table 5.4 Subset of the results for the minimum of the objective function surfaces for various placements of the gross errors.

GA - Gross Errors, 1 & 17			GA - Gross Errors, 5 & 21			GA - Gross Errors, 5 & 21		
Predicted errors		Recursive Objective	Predicted errors		Recursive Objective	Predicted errors		Global Objective
10	17	2.5104	5	21	5.1985	5	21	2.4524
17	22	3.4765	5	17	34.9425	1	21	104.8094
5	17	3.5324	5	20	38.0897	10	21	213.0984
1	17	3.6386	5	19	40.7819	11	21	223.3135
14	21	3.9673	5	7	41.2424	12	21	243.9394
14	17	3.9673	5	14	41.9495	7	9	253.667
17	21	3.9673	5	18	42.6798	7	21	253.667
15	24	4.0402	4	5	44.7533	9	21	253.667
17	24	4.0402	5	9	46.6913	16	21	257.1995
15	17	4.0402	5	28	48.7686	21	25	259.4346

Table 5.4 shows the combinations of errors which give the ten lowest values of the objective functions for each case considered. Figures 5.2 to 5.5 show the complete surface for all the combinations of removing two measurements from the original measurement structure of twenty-eight measurements. The surfaces are only drawn on the right half of the diagrams because the left half mirrors the former, i.e. the effect on the objective function due to the removal of measurements x and y has the same effect as removing measurements y and x.

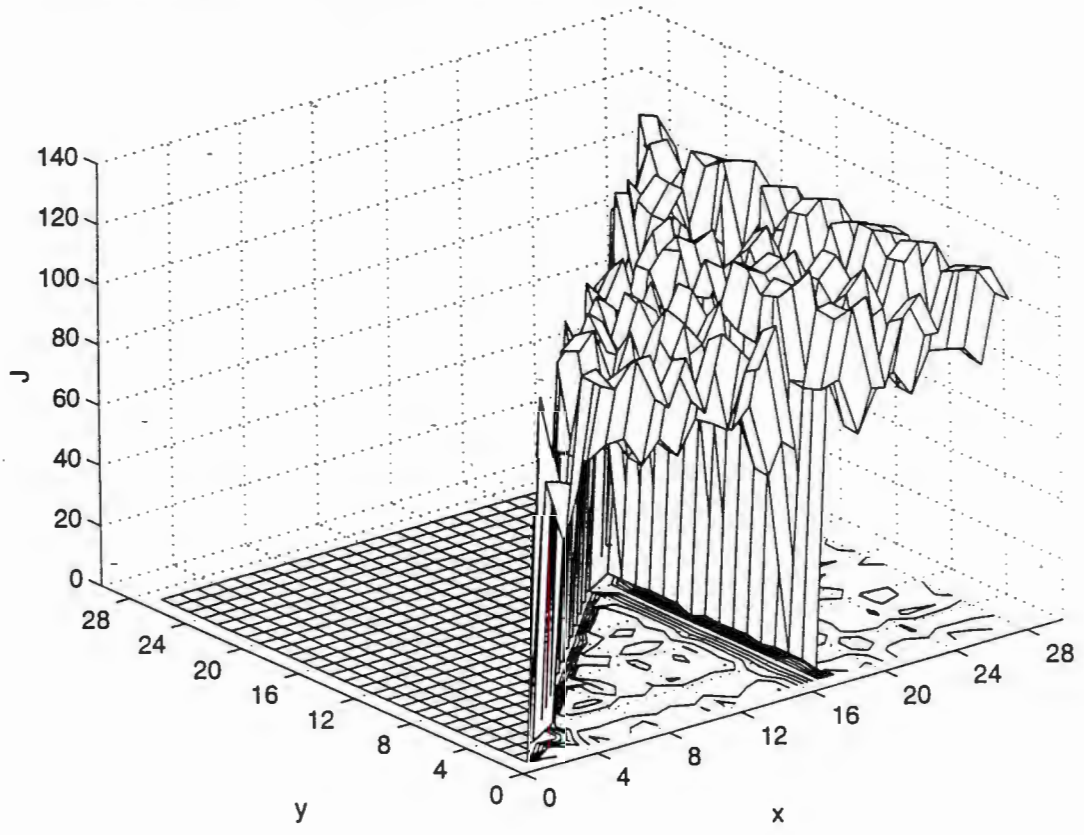


Figure 5.2 The surface of the recursive gross error method, gross errors at positions 1 & 17

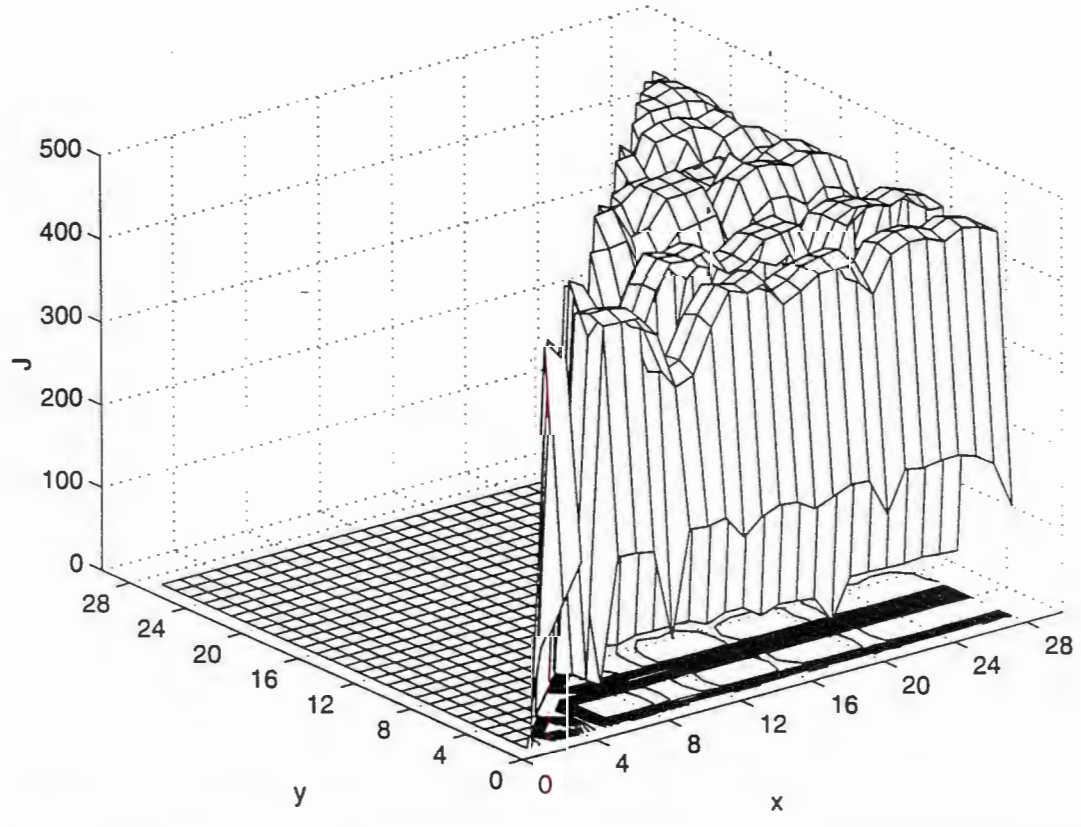


Figure 5.3 The surface of the recursive gross error method, gross errors at positions 5 & 21

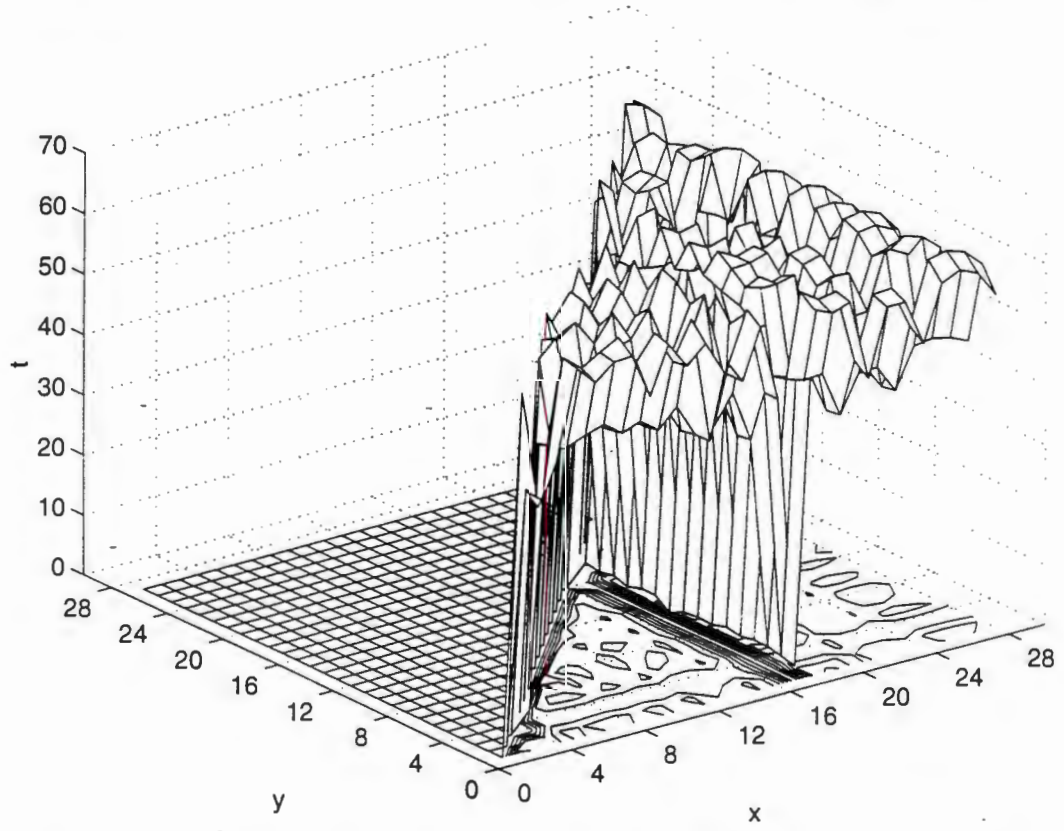


Figure 5.4 The surface of the global test statistic, gross errors at positions 1 & 17

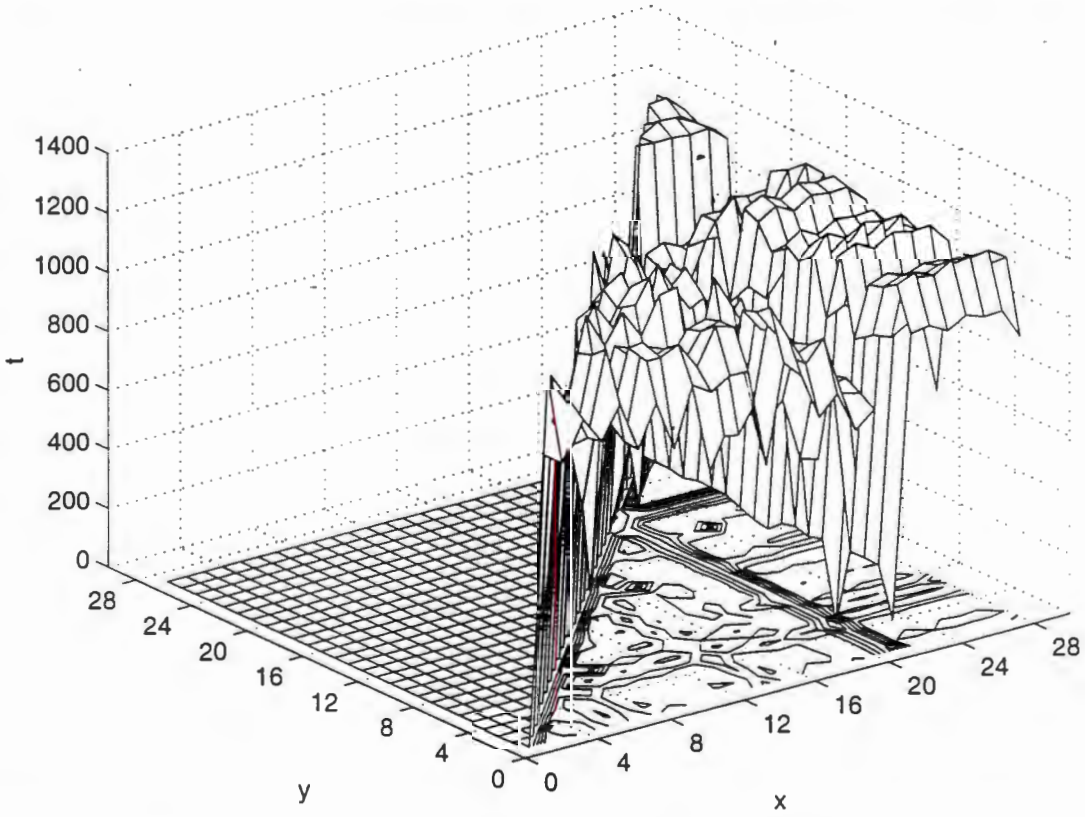


Figure 5.5 The surface of the global test statistic, gross errors at positions 5 & 21

From the results in Table 5.4 and Fig. 5.2 to 5.5 it is easy to see why GAs perform so badly in gross error detection. From Table 5.4, when the error is placed at points one and seventeen, the minimum value of the recursive gross error detection test does not correspond to the positions at which the gross errors were introduced. This is a characteristic of the statistical tests being used for the objective functions in the GAs. In comparison with the test surface in Fig. 5.1, the surfaces in Fig 5.2 to 5.5 can be seen to be very irregular. The surfaces contain no gradient leading to the minimum but instead look almost random in nature. Although GAs do not need gradient information, without a well defined minimum point its performance can be expected to be poor.

The performance of the GA method is so poor that it rates comparably with a random search, which, from inspection of the surfaces in Fig. 5.2 to 5.5, is not surprising. Due to the irregular surfaces the performance of the GAs is reduced to the equivalent of a random search. The fault does not lie with the GA, but in the selection of the objective function. The minimum value of the statistical objective function depends on the position and size of the gross error and may not lie at the position at which the gross error was placed, as shown in Table 5.4. If the minimum of the objective function does not correspond to the measurements containing gross errors, then an optimization method will fail to find the solution.

In addition, it can be seen, Fig. 5.5 and Table 5.3, that the surface defined by using the global test statistic is even more severely irregular than that obtained using the recursive gross error detection method. For this reason the GA using the global measurement statistic performs poorly when compared to the other results obtained.

6. INDUSTRIAL CASE STUDY

6.1. General Description

The crude preheat train of a local refinery was used as an industrial example for data reconciliation, gross error analysis, sensitivity analysis, measurement structure synthesis and confidence interval calculation. The flow network is presented in Fig. 6.1.

The network consists of eight heat exchangers, three mixers, one splitter and a desalter. Each stream in Fig. 6.1 contains one of the following fluids, crude oil, light diesel (LDSL), top pump around (TPA), middle pump around (MPA), visbreaker gas oil (VGO), visbreaker bottoms (VBO) and water. Top and middle pump around are drawn from the respective positions on an atmospheric column. The crude enters from the top of Fig. 6.1 and flows downwards, splitting into two streams flowing respectively through heat exchangers 2E201 and 2E202, and 2E301, 2E2, 2E6 and 2E3 before rejoining, after which water (stream 23) is added before it flows into the desalter 2D1. From 2D1 the crude flows through 2E4A and 2E4B, and is split into two streams before entering the first furnace. TPA flows through exchanger 2E201, MPA though 2E202 and then 2E6, VBO through 2E301, light diesel through 2E2, VGO from two streams joins together before flowing through 2E3, and VBO flows through exchangers 2E4B and then 2E4A.

With the exception of the VGO mixer before 2E3, the addition of water before 2D1 and the balance around 2D1, heat balances are performed on each unit in the flowsheet. Mass balances are performed on all of the units. The addition of the heat balances for the VGO mixer and around 2D1 would produce no pertinent information without more information about upstream processes. For practical reasons it was decided to limit the size of the flow sheet under investigation. For example: without information on the upstream temperatures a heat balance around the VGO mixer before 2E3 would match the temperatures of the input streams to the measured output stream temperature; this would be an unnecessary addition to the model equations.

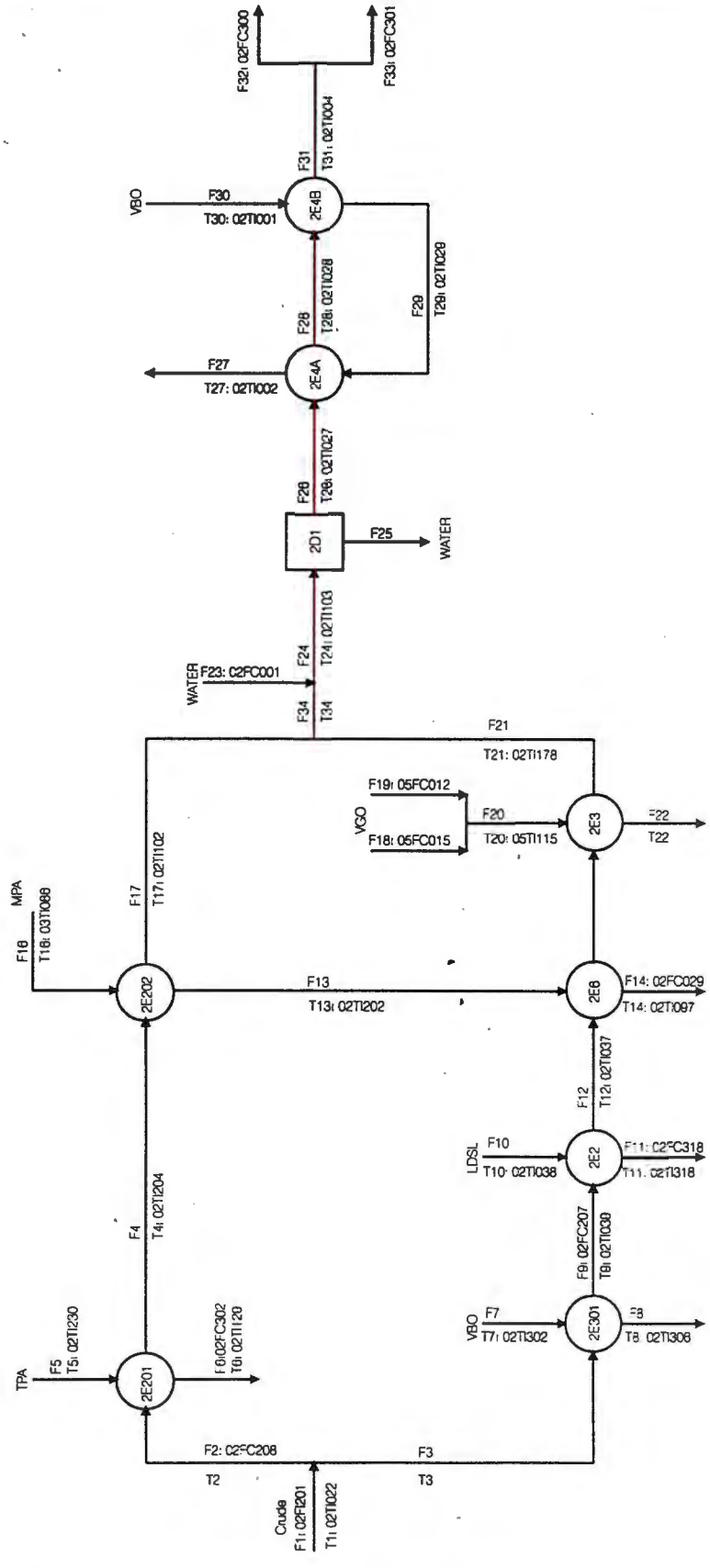


Figure 6.1 The crude preheat heat exchanger network.

The data obtained is in the form of two-hourly averages over a period of three days. This data has been scaled for one hundred units of crude entering the network. In Fig. 6.1 measurements are represented by their tags; for example, 02FI201 is a flow indicator on plant 2 (the crude unit) and 03TI086 is a temperature indicator on plant 3. In the case of 03TI086, the MPA from the atmospheric column on plant 2 is piped to plant 3, used for heat exchange and returned to plant 2 for heat exchange with the crude in heat exchanger 2E202. Unmeasured variables are represented on Fig. 6.1 as Fk or Tk where k represents the stream number.

6.1.1. Flow correction

Flow measurement devices generally involve flow through orifice plates and a measure of the velocity of the flow is obtained from the pressure drop across the orifice plate. The pressure drop is then converted to an electronic signal. At the refinery, these electronic signals are linked to a DCS.

The calculation of the velocity of the fluid from the pressure drop across the orifice plate depends on the density of the fluid. Consequently, each flowmeter is designed for a given density. When the flow is measured on the plant, the actual fluid density may differ from the design density. Therefore, the flow rate measured by the instrument must be corrected for this discrepancy in density. The following correction equations are used (Smith, 1995):

$$\text{Corr. Flow} = (\text{DCS value})\frac{SG_{design@20°C}}{SG_{actual@20°C}}\sqrt{\frac{SG_{actual@F,T}}{SG_{design@F,T}}} \qquad 6.1$$

$$\text{where} \quad SG_{actual@F,T} = SG^{2}_{actual@20°C} - 0.0011(T - 20))^{0.5}$$

6.1.2. Calculation of enthalpy curves

The data for the generation of enthalpy curves for each of the streams is calculated by using a spreadsheet developed and supplied by the refinery. The enthalpy at a specific temperature is calculated from the Watson characterization and the acentric factors. These two factors are calculated from the ATSM D86 distillation data and the specific gravity of the fluid in question. The temperature is varied over a likely range and the equations for the enthalpy

curves are generated by regressing the data to fit quadratic equations.

6.2. Analysis of Original Flow Network

The initial reconciliation and analysis of the industrial heat exchanger network is shown in Table 6.1. Data reconciliation, statistical tests, a sensitivity analysis, the assignment of confidence intervals and the classification of variables were performed on the system.

Using the approach of Madron (1985), the variance of the residuals ($r = y - x^*$) can be calculated from which the 100(1-α)% confidence intervals for each measurement can be found:

$$y_i - x_i^* = \pm z_\alpha \sqrt{e_i^T \Sigma_r e_i}$$

where

$$\Sigma_r = \Sigma C^T [C \Sigma C^T]^{-1} C \Sigma \qquad 6.2$$

where e_i is a vector of zeros with unity in the i^{th} position, α the statistical level of significance equals 0.05 and $z_{\alpha/2}$ is obtained from statistical tables of the normal variate (Miller and Freund, 1985).

6.2.1. Initial results of the industrial case study

The data reconciliation is nonlinear because of the inclusion of heat balances. The reconciliation was performed using a Gauss-Newton iterative scheme incorporating QR decomposition to decouple the unmeasured variables from the least squares optimization, as presented in subsection 2.1.4. All calculations are MATLAB and the m-files are presented in Appendix I.

The process was assumed to be at steady state. Standard deviations of five and two percent were used for the flow and temperature measurements respectively. Table 6.1 shows the results of the data reconciliation and subsequent analysis of the system using the first dataset.

Table 6.1 Results from analysis of the original heat exchanger network.

		1	2	3	4	5	6	7
z	TAG	Initial Measurement	Recon Value	zi	Variance	Sensitivity	Confidence Interval	Variable Classification
1	F1-FI201	100.00	58.2004	9.2992	25.0000	3.21E-04	8.7652	R
2	T1-TI022	86.30	87.0394	30.0858	2.9791	1.56E-05	0.0479	R
3	F2-FC208	52.75	-0.7568	20.2915	6.9551	1.30E-04	5.1415	R
4	T2	100.00	87.0394					O
5	F3	100.00	58.9571					O
6	T3	100.00	87.0394					O
7	F4	100.00	-0.7568					O
8	T4-TI204	26.00	25.9633	26.9169	0.2704	2.13E-06	0.0027	R
9	F5	100.00	0.3544					O
10	T5-TI230	196.60	194.3623	30.0859	15.4606	2.60E-05	0.1450	R
11	F6-FC302	0.37	0.3544	30.0858	0.0003	5.24E-05	0.0009	R
12	T6-TI129	87.40	87.7739	30.0858	3.0555	4.86E-06	0.0242	R
13	F7	100.00	30.5785					O
14	T7-TI302	192.10	192.1	0	14.7610			NR
15	F8	100.00	30.5785					O
16	T8-TI306	138.30	138.3	0	7.6508			NR
17	F9-FC207	48.94	58.9571	9.1632	5.9881	9.51E-04	2.1314	R
18	T9-TI039	119.90	119.8735	0.0144	5.7504	2.48E-03	3.5964	R
19	F10	100.00	6.0848					O
20	T10-TI038	217.05	217.039	0.0144	18.8443	4.47E-04	1.4928	R
21	F11-FC318	6.09	6.0848	0.0144	0.0926	5.92E-04	0.0702	R
22	T11-TI316	156.25	156.2552	0.0144	9.7656	2.70E-04	0.7087	R
23	F12	100.00	58.9571					O
24	T12-TI037	127.75	127.1994	0.2625	6.5280	1.04E-03	4.0899	R
25	F13	100.00	-0.226					O
26	T13-TI202	111.95	111.7951	15.6216	5.0131	4.90E-06	0.0193	R
27	F14-FC029	12.87	-0.226	20.3607	0.4141	6.77E-05	1.2542	R
28	T14-TI097	117.25	117.248	0.2751	5.4990	2.80E-06	0.0138	R
29	F15	100.00	58.9571					O
30	T15-TI040	126.65	127.2211	0.2758	6.4161	1.03E-03	4.0375	R
31	F16	100.00	-0.226					O
32	T16-03TI086	310.10	311.6496	20.3621	38.4648	6.22E-06	0.1484	R
33	F17	100.00	-0.7568					O
34	T17-TI102	106.05	105.6127	20.3622	4.4986	4.58E-06	0.0419	R
35	F18-5FC015	3.29	3.2938	0	0.0271			NR
36	F19-5FC012	6.39	6.3853	0	0.1019			NR
37	F20	100.00	9.6791					O
38	T20-05TI115	255.90	255.9	0	26.1939			NR
39	F21	100.00	58.9571					O
40	T21-TI178	126.30	126.3	0	6.3807			NR
41	F22	100.00	9.6791					O
42	T22	100.00	260.5365					O
43	F23-FC001	5.10	5.0963	0	0.0649			NR
44	F34	100.00	58.2004					O
45	F24	100.00	63.2966					O
46	T24-TI103	115.90	115.6801	0.1299	5.3731	3.02E-04	3.3016	R
47	F25	100.00	-20.2809					O
48	T34	100.00	126.5638					O
49	F26	100.00	83.5775					O
50	T26-TI027	115.95	115.6801	0.1593	5.3778	3.01E-04	3.3042	R
51	F28	100.00	83.5775					O
52	F31	100.00	83.5775					O
53	F32-FC300	41.87	41.8724	0	4.3833			NR
54	F33-FC301	41.71	41.7051	0	4.3483			NR
55	F27	100.00	20.5348					O
56	T27-TI002	208.45	208.9325	0.5754	17.3806	3.54E-04	1.6353	R
57	T28-TI028	129.50	130.5566	0.5754	6.7081	1.48E-03	3.5810	R
58	F29	100.00	20.5348					O
59	T29-TI029	257.90	256.5596	0.5754	26.6050	9.44E-04	4.5427	R
60	F30	100.00	20.5348					O
61	T30-TI001	319.80	320.7179	0.5754	40.9088	4.60E-04	3.1109	R
62	T31-TI004	152.30	151.6848	0.5754	9.2781	7.01E-04	2.0848	R
63	T25	100.00	115.6801					O

R = Redundant measurement
NR = Nonredundant measurement
O = Observable variable

The results for the data reconciliation and gross error detection of the original flow network are displayed in the 2nd and 3rd columns of Table 6.1. Analysis of the results shows that twelve measurements are reported as containing gross errors. The measurements whose measurement test statistic exceeds the critical value of z_c=3.15 are F1: 02FI201, T1: 02TI022, F2: 02FC208, T4: 02TI204, T5: 02TI230, F6: 02FC302, T6: 02TI129, F9: 02FC207, T13: 02TI202, F14: 02FC029, T16: 03TI086 and T17: 02TI102. A brief analysis of the values of the measurements around 2E202 reveals an obvious error. The crude output temperature T4: 02TI204 (25.9 °C) is lower than the input temperature T2 (85.9 °C). Although T2 is an unmeasured value it can be calculated from the temperature measurement T1: 02TI022. The most likely explanation is that the measurement T4: 02TI204 has failed and is reporting the ambient temperature.

The results obtained from the data reconciliation and calculation of the unmeasured variables are very poor. Negative flowrates in the following five variables are reported, F2: 02FC208, F4, F13, F14: 02FC029, F16, F17 and F25. These would imply the reversal of all but two (TPA: F5 and F6: 02FC302) of the flowrates in the upper leg of the network (2E201 and 2E202) and the addition of water to the desalter 2D1, neither of which is physically possible. The results indicate that the data is corrupted by at least one or more gross errors.

The sensitivity of the system, as reported in column 5 of Table 6.1, appears very low for a large majority of measurement variances. The reported values are the 1-norm values of the scaled sensitivity matrix and represent the effect of individual variances on the reconciled values of the remaining measurements. The system sensitivities are calculated using the method presented in subsection 3.1. The variance of measurement T9: 02TI039 is reported as having the largest effect on the data reconciliation scheme. The sensitivity analysis will be re-evaluated once all the gross errors have been removed. The 95% confidence intervals for the reconciled measurements are reported in column 6 of Table 6.1. The results from the variable classification, as shown in column seven of Table 6.1, report that nine of the thirty-five measurements are nonredundant while all the unmeasured variables are observable. The absence of unobservable variables is a 'good' feature of this measurement structure; however the nonredundant measurements are potential sites where gross errors can exist. The presence of nonredundant measurements degrades the performance of the data reconciliation.

In Fig. 6.4 to 6.7 the solid lines represent the reconciled values of the measurements while the points represent the original reported values. The figures show the adjustment made by the reconciliation procedure across all the datasets considered.

6.5. Measurement Structure Synthesis

Present in the industrial case study's original measurement structure are a number of nonredundant measurements. These are clearly shown in Table 6.1. Using the principles developed in Chapter 4, new measurement structures with improved variable classification properties will be proposed.

6.5.1. LSA solution

The variables are constrained as follows: all measurements are to be redundant and all unmeasured variables are to be observable. The GSA was applied to the industrial case study, but failed to reach a solution within an acceptable time. The problem size is too large for an exhaustive search for the global solution. The problem relating problem size and solvability in the GSA was addressed by formulation of the LSA, which, although not guaranteed to reach the global solution has been shown to provide acceptable solutions in a short period of time. Multiple applications of the LSA have been applied to the industrial case study in order to find an optimum measurement structure. The table of the initial starting measurement structures is presented in Appendix I.

The results are presented in Table 6.5. The 10 columns apply to the ten applications of the LSA. In the columns of Table 6.5 zero entries represent a measurement and unity an unmeasured variable. The number of equivalent solution sets is shown in Table 6.5. As in subsections 4.3.1.2. and 4.3.2.2. only one of these equivalent solutions is displayed. The times taken to reach these solutions are also reported in Table 6.5.

Table 6.5 Measurement Structure Synthesis results

	TAG	Applications of the LSA									
		1	2	3	4	5	6	7	8	9	10
1	F1-FI201	0	1	0	0	0	0	0	0	1	0
2	T1-TI022	0	0	0	0	0	0	1	1	1	0
3	F2-FC208	1	0	1	0	1	1	0	1	0	0
4	T2	1	1	1	1	1	1	0	0	0	1
5	F3	1	1	1	1	1	0	1	1	0	1
6	T3	1	1	1	1	1	1	1	1	1	1
7	F4	1	1	1	1	1	1	1	1	1	1
8	T4-TI204	0	0	1	0	1	1	1	0	0	1
9	F5	0	0	1	1	0	0	0	1	0	0
10	T5-TI230	0	0	0	0	0	0	0	0	0	0
11	F6-FC302	1	1	0	0	1	1	1	0	1	1
12	T6-TI129	0	0	0	0	0	0	0	0	0	0
13	F7	1	0	0	0	0	1	0	1	0	0
14	T7-TI302	0	0	0	0	0	0	0	0	0	0
15	F8	0	1	1	1	1	0	1	0	1	0
16	T8-TI306	0	0	0	0	0	0	0	0	0	0
17	F9-FC207	1	1	1	1	1	1	1	1	1	1
18	T9-TI039	0	1	1	1	0	1	1	0	1	0
19	F10	0	0	1	1	1	0	1	0	1	0
20	T10-TI038	0	0	0	0	0	0	0	0	0	0
21	F11-FC318	1	1	0	0	0	0	0	1	0	0
22	T11-TI316	0	0	0	0	0	0	0	0	0	0
23	F12	1	1	1	1	1	1	1	1	1	1
24	T12-TI037	1	1	0	0	1	1	1	1	0	1
25	F13	1	0	0	1	1	0	0	1	0	0
26	T13-TI202	1	0	0	0	0	0	1	0	0	0
27	F14-FC029	1	1	1	0	1	1	1	0	1	1
28	T14-TI097	0	0	1	0	0	0	0	0	0	0
29	F15	1	1	1	1	1	1	1	1	0	1
30	T15-TI040	1	1	0	1	0	1	1	1	1	1
31	F16	0	1	1	1	0	1	1	1	1	1
32	T16-03TI086	0	1	0	1	0	0	0	0	1	0
33	F17	1	1	1	1	1	1	1	1	0	0
34	T17-TI102	0	0	0	0	1	1	0	1	0	1
35	F18-5FC015	0	0	0	0	0	0	0	0	0	0
36	F19-5FC012	0	0	0	0	0	0	0	0	0	0
37	F20	1	1	0	0	1	1	0	1	1	1
38	T20-05TI115	0	0	0	0	0	0	0	0	0	0
39	F21	1	1	1	1	1	1	1	1	1	1
40	T21-TI178	0	1	0	1	1	0	0	0	1	0
41	F22	0	0	1	1	0	0	1	0	1	1
42	T22	0	0	0	0	0	0	0	0	0	0
43	F23-FC001	0	0	0	0	0	1	1	1	0	0
44	F34	1	1	1	1	1	1	1	1	1	1
45	F24	0	1	1	1	0	0	0	0	0	0
46	T24-TI103	0	0	0	0	0	0	0	0	1	0
47	F25	1	0	0	0	1	0	0	0	0	0
48	T34	1	0	1	0	0	0	1	0	0	0
49	F26	1	1	1	1	0	1	1	1	1	1
50	T26-TI027	1	1	0	1	1	1	0	1	0	1
51	F28	1	1	1	1	1	1	1	1	0	1
52	F31	1	1	1	1	1	1	1	1	1	1
53	F32-FC300	0	0	0	0	0	0	0	0	0	0
54	F33-FC301	0	0	0	0	0	0	0	0	0	0
55	F27	1	1	1	0	1	1	0	1	1	0
56	T27-TI002	0	0	1	0	0	0	1	0	0	0
57	T28-TI028	0	0	0	0	0	0	0	0	0	0
58	F29	0	1	0	1	1	1	1	1	1	0
59	T29-TI029	0	0	0	0	0	0	0	0	0	0
60	F30	1	1	1	1	1	1	1	1	1	1
61	T30-TI001	0	0	0	0	0	0	0	0	0	0
62	T31-TI004	0	0	0	0	0	0	0	0	0	0
63	T25	1	1	1	1	1	1	1	1	1	1
Number of Measurements		34	32	34	35	34	34	33	34	37	39
	Solution Set	2	2	2	4	3	4	9	1	2	1
	Time [Min]	20.17	8.37	19.88	10.18	4.95	10.72	18.41	4.04	1.77	5.36

The original measurement structure consisted of thirty-five measurements, twenty-eight unmeasured variables and contained a number of nonredundant measurements. The results in Table 6.5 show that in most cases, the LSA was able to find better (in terms of the number of measurements) measurement structures that satisfy the variable constraints. These measurement structures contain only redundant measurements and observable variables.

The optimal solution is represented in column 2 of Table 6.5. The time taken to reach a solution varies greatly over all the applications of the LSA. The characteristic of the LSA to find local solutions is easily seen in the results of Table 6.5. The results obtained vary from 32 to 39 measurements. Unlike the examples in Chapter 4, the global solution is unknown and a solution would have to be chosen from those in column 2 of Table 6.5.

6.5.2. Retro-fit problem

Since an original measurement structure exists, it might not be appropriate to employ one of the LSA solutions to correct its shortcomings. For this reason the retro-fit problem was formulated. It involves identifying the modifications that need to be made to the original measurement structure. The following problems were considered:

1) What measurements need to be added to ensure that all the original measurements are redundant.
2) What measurements need to be added to ensure that all the original measurements have a degree of redundancy of one.

6.5.2.1. Problem 1

This problem is solved using a slightly modified version of the LSA. Instead of a random set of initial unmeasured variables, the original measurement structure is used as the starting point of the LSA. The variables are constrained as follows: all the original measurements are to be measured, and only redundant measurements and unobservable variables may exist.

The LSA adds measurements to remove all the nonredundant measurements and then attempts to remove the original measurements in search for the optimal measurements structure. However, the original measurements cannot be removed because they are constrained to be

measured. Not been able to improve on the measurement structure the LSA terminates, and the solution identifies the modifications that need to be made to the initial measurement.

The solution was obtained in 2.4 seconds and reports that the conversion of variables F7, T22 and F25 to measurements, would ensure that all the measurements are redundant.

6.5.2.2. Problem 2

The same modified LSA described in the previous section was used. The variables are constrained as in problem 1, in addition the original measurements must have a degree of redundancy equal to one.

The LSA failed to reach a solution because measurements exist that fail the degree of redundancy constraint, even if all the variables are measured. There are fifteen of these measurements, namely: T5: 02TI230, T6: 02TI129, T7: 02TI302, T8: 02TI306, T10: 02TI038, T11: 02TI316, F18: 05FC015, F19: 05FC012, T20: 05TI115, T28: 02TI028, T29: 02TI029, T30: 02TI001, T3102TI004, F32: 02FC300 and F33: 02FC301. For example, even if all the variables are measured, measurement T5: 02TI230 does not have a degree of redundancy of one, because the deletion of measurement T6: 02TI129 will cause T5: 02TI230 to become nonredundant. Measurements T6: 02TI129 and T5: 02TI230 are the only temperature measurements on the TPA line into heat exchanger 2E201. If any one is removed, the other would become unobservable upon deletion, and is therefore nonredundant. Similar situations exist for the remaining measurements listed above.

The variable constraints of the problem need to be modified, since not all the original measurements are able to have a degree of redundancy of one. The choice of a subset of measurements that can be constrained as having a degree of redundancy is left to the user. Once this choice has been made the LSA can be used to identify the optimum measurement structure.

7. CONCLUSION

This dissertation has presented the results of an investigation into enhancements that can be made to the data reconciliation and gross error techniques.

A review of different approaches to data reconciliation and gross error detection was included. This covers steady state and dynamic data reconciliation, variable classification, gross error detection, and the development of data reconciliation in the minerals processing arena.

An investigation into the sensitivity of the data reconciliation procedure to the measurement variances was conducted. This produced a sensitivity matrix through the differentiation of the reconciliation equation with respect to the measurement variances. An analysis of the sensitivity matrix allows one to rank measurement variances with respect to their influence on the reconciliation equation. It was found that the reconciliation procedure could have entirely different sensitivities to two measurements similar in value and variance, at different positions in a flow network. Furthermore, it was discovered that the sensitivity of the reconciliation procedure is carried through to the statistical measurement test. Inaccuracy in the variance of certain measurements can cause uncertainties in the calculated statistics. Measurements that initially failed the measurement test can pass if their variances are erroneously over-estimated. This emphasizes the need to have accurate measurement variances. The sensitivity analysis identifies the measurement variances to which the reconciled system is highly sensitive.

The Global Search Algorithm (GSA) to find the optimal measurement structure subject to variable classification constraints was developed. Variables can be classified as measured or unmeasured, redundant or nonredundant, observable or unobservable, and a degree of redundancy can be imposed on any measurement/s. Although the procedure uses techniques to limit the search, a large number of combinations still need to be tested. Without the placement of classification constraints on the majority of the variables, this method can soon

become computationaly intractable for large systems (> 30 measurements). Notwithstanding these results, success was achieved using this algorithm in medium sized systems. The Local Search Algorithm (LSA) was developed to overcome the limitations of the GSA. The LSA performs a restricted search and is not guaranteed to find the global optimum solution. However the LSA does provide 'good' solutions in a fraction of the time required by the GSA. In addition, it has been shown that multiple applications of the LSA with different starting points may lead to the global optimum being found. In the examples considered, the LSA found the global optimum at least once after ten applications. Thus if the problem size prohibits the application of the GSA, the LSA can be used to find a good solution to the measurement structure synthesis problem.

As an introduction, a description of the basic structure and major components of a Genetic Algorithm has been presented. It was proposed that Genetic Algorithms replace the serial elimination schemes in gross error detection. The global test statistic and recursive identification method were used for objective functions of the Genetic Algorithms. The performance of the statistics were seriously affected by size and position of the gross errors. This results in a situation where the minimum of the statistic's surface does not correspond to the position of the gross errors. This fact coupled with the extremely irregular surface of the statistical tests, degrades the performance of Genetic Algorithms to that of a random search.

Data reconciliation was performed on an industrial example. The subsequent search for gross errors, using the Modified Iterative Measurement Test method, identified two gross errors. A sensitivity analysis was performed, and correctly identified which measurement variances the reconciliation procedure is most and least sensitive to. Variable classification indicated that the original measurement structure contained several nonredundant measurements. Measurement structure synthesis was applied to the problem in order to obtain a new measurement structure devoid of nonredundant measurements. The size of the problem prohibited the GSA in finding a solution in an acceptable time. Measurement structure synthesis was applied successfully to the industrial case study using the LSA. The LSA returns a measurement structure with fewer measurements than the original, but with no nonredundant measurements. It is proposed that the new measurement structure is an

improvement, because it has improved variable classification properties (the absence of nonredundant measurements) with fewer measurements than the original. In addition the modifications needed to retro-fit the original measurement structure, eliminating the nonredundant measurements, were identified. These successful applications of the LSA shows its ability to solve the measurement structure synthesis problem on an industrial scale.

REFERENCES

Albuquerque, J.S. and Bielger, L.T., "Gross Error Detection and Variable Classification in Dynamic Systems", presented for publication in *AIChE*, (1995)

Aldrich, C. and van Deventer, J.S.J., "The Use of Connectionist Systems to Reconcilied Inconsistent Process Data", *The Chemical Engineering Journal*, **54**, 125-135, (1994)

Almasy, G.A. and Sztano, T., "Checking and Correction of Measurements on the Basis of Linear System Model" *Prob. Control Information Theory*, **4**(57), 57-69, (1975)

Almasy, G.A. and Mah, R.S.H., "Estimation of Measurement Error Variances from Process Data", *Industrial Engineering Chemical Process Design Development*, **23**, 779-784, (1984)

Almasy, G.A., "Principles of Dynamic Balancing", *AIChE Journal*, **36**(9), 1321-1330, (1990)

Beasly, D., Bull, D.R. and Martin, R.R., "An Overview of Genetic Algorithms: Part 1, Fundamentals", *University Computing*, **52**(2), 58-69, (1993)

Britt, H.I. and Luecke, R.H., "The Estimation of Parameters in Nonlinear Implicit Models", *Technometrics*, **15**(2), 233-247, (1973)

Broyden, C.G., "A Class of Methods for Solving Nonlinear Simultaneous Equations" *Mathematics of Computation*, **19**, 577, (1965)

Crowe, C.M., Campos, Y.A.G. and Hrymak, A., "Reconciliation of Process Flow Rates by Matrix Projection. Part I: Linear Case", *AIChE Journal*, **29**(6), 881-888, (1983)

Crowe, C.M., "Reconciliation of Process Flow Rates by Matrix Projection. Part II: The

Nonlinear Case", *AIChE Journal*, **32**(4), 616-623, (1986)

Crowe, C.M., "Recursive Indentification of Gross Errors in Linear Data Reconciliation", *AIChE Journal,* **34**(4), 541-550, (1988)

Crowe, C.M., "Test of Maximum Power for Detection of Gross Errors in Process Constraints" *AIChE Journal*, **35**(5), 869-872, (1989)

Crowe, C.M., "Observability and Redundancy of Process Data for Steady State Reconciliation", *Chemincal Engineering Science*, **44**(12), 2909-2917, (1989)

Crowe, C.M., "Data Reconciliation - Progress and Challenges", *Journal of Process Control*, **6**(2/3), 89-98, (1996)

Cutting, G.W., "Estimation of Interlocking Mass-Balances on Complex Mineral Beneficiation Plants", *International Journal of Mineral Processing*, **3**, 207-218, (1976)

Darouach, M., Ragot, R., Zasadzinski, M. and Krzakala, G., "Maximum Likelihood Estimator of Measurement Error Variances in Data Reconciliation", *IFAC, AIPAC Symposium,* **2**, 135-139, (1989)

Dantzig, G., *Linear Programming and Extensions.*, Princeton University Press, NJ (1963)

Duran, M.A. and Grossmann, I.E., "A Mixed-Integer Nonlinear Programming Algorithm for Process Systems Synthesis", *AIChE Journal*, **32**(4), 592-606, (1986)

Eitelberg, E., "Optimal Estimation for Engineers", NOYB Press, South Africa, (1991)

Figueroa, J.L. and Romagnoli, J.A., "A Strategy for Dynamic Data Reconciliation within the MATLAB Environment", *Chemica*, **2**, 819-825, (1992)

Fraleigh, J.B. and Beauregard, R.A., "Linear Algebra", Addison-Wesley Publishing Company, New York, (1990)

Gleb, A. "Applied Optimal Estimation", MIT Press, Cambridge, (1974)

Hlaváček, V., "Analysis of a Complex Plant-Steady State and Transient Behavior", *Computers and Chemical Engineering*, Vol. 1, 75-100, (1977)

Holland, J., "Adaptation in Natural and Artificial Systems", University of Michigan Press, (1975)

Hodouin, D. and Everell, M.D., "A Hierarchical Procedure fro Adjustment and Material Balancing of Mineral Processes Data", *International Journal of Mineral Processing*, **7**, 91-116, (1980)

Iordache, C., Mah, R.S.H. and Tamhane A.C., "Performance Studies of the Measurement Test for Detection of Gross Errors in Process Data", *AIChE Journal*, **31**(7), 1187-1201, (1985)

Jang, S.S., Joseph, B. and Mukai, H., "Comparison of Two Approaches to On-line Parameter and State Estimation on Nonlinear Systems", *Industrial Engineering Chemical Process Design Development*, **25**, 809-814, (1986)

Kramer, M.A. and Mah, R.S.H., "Model-Based Monitoring", in *Foundations of Computer Aided Process Operations*, Rippin, D.W.T., Hale, J.C. and Davis, J.F., eds., 45-67, CACHE (1993)

Keller, J.Y., Zasadzinski, M. and Darouach, M., "Analytical Estimator of Measurement Error Variances in Data Reconciliation", *Computers in Chemical Engineering*, **16**(3), 185-188, (1992)

Klimpel, R., "Estimation of Weight Ratios Given Component Make-Up Analyses of Streams", AIME Annual Meeting, New Orleans, **266**, (1978)

Knepper, J.C. and Gorman, J.W., "Statistical Analysis of Constrained Data Sets", *AIChE Journal*, **26**(2), 260-264, (1980)

Kuehn, D.R. and Davidson, H., "Computer Control II. Mathematics of Control", *Chemical Engineering Progress*, **57**(6), 44-47, (1961)

Kumar, V.R., Singh, S.R., Garg, M.O. and Narasimhan, S., "Rage - A Software Tool for Data Reconciliation and Gross Error Detection", in *Foundations of Computer Aided Process Operations*, Rippin, D.W.T., Hale, J.C. and Davis, J.F., eds., 429-435, CACHE (1993)

Laguitton, D. and Wilson, J.M.D., "Material Balance: A Step to Towards Modelling of Fine Grained Sulphide Ores Treatment", Canadian Minerals Processors, Ottawa, January 24, (1979)

Lasdon, L.S. and Waren, A.D., "GRG2 User's Guide", Department of Management Science and Information Systems, University of Texas at Austin 78712, (1989)

Lawrence, P.J., "Data Reconciliation: Getting Better Information", *Hydrocarbon Processing*, 55-60, June (1989)

Liebman, M.J. and Edgar, T.F., "Data Reconciliation for Nonlinear Processes", *AIChE Annual Meeting*, Washington, D. C., November 1988

Liebman, M.J., Edgar, T.F. and Lasdon, L.S., "Efficient Data Reconciliation and Estimation for Dynamic Processes using Nonlinear Programming Techniques", *Computers and Chemical Engineering*, **16**(10/11), 963-986, (1992)

Madron, F., Veverka, V. and Vanecek, V., "Statistical Analysis of Material Balance of a

Chemical Reactor", *AIChE Journal*, **23**(4), 482-486, (1977)

Madron, F., "A New Approach to the Indentification of Gross Errors in Chemical Engineering Measurements", *Chemical Engineering Science*, **40**(10), 1855-1860, (1985)

Madron, F. and Veverka, V. "Optimal Selection of Measuring Points in Complex Plants by Linear Models", *AIChE Journal*, **38**(2), 227-236, (1992)

Mah, R.S.H., Stanley, G.M. and Downing, D.M., "Reconciliation and Rectification of Process Flow and Inventory Data", *Industrial Engineering Chemical Process Design Development*, **15**(1), 175-183, (1976)

Mah, R.S.H. and Tamhane, A.C., "Detection of Gross Errors in Process Data", *AIChE Journal*, **28**(5), 828-830, (1982)

Meyer, M., Koehret, B. and Enjalbert, M., "Data Reconciliation on Multicomponent Network Process", *Computers in Chemical Engineering*, **17**(8), 807-817, (1993)

Miller, I. and Freund, J.E., "*Probability and Statistics for Engineers*", Prentice-Hall Internation, New Jersey, (1985)

Narasimhan S. and Mah, R.S.H., "Generalized Likelihood Ratio Method for Gross Error Identification", *AIChE Journal*, **33**(9), 1514-1521, (1987)

Narasimhan, S., "Maximum Power Tests for Gross Error Detection using Likelihood Ratios", *AIChE Journal*, **36**(10), 1589-1591, (1990)

Narasimhan, S. and Harikumar, P., "A Method to Incorporate Bounds in Data Reconciliation and Gross Errors Detection - 1. The Bounded Data Reconciliation Problem", *Computers and Chemical Engineering*, **17**(11), 1115-1120, (1993a)

Narasimhan, S. and Harikumar, P., "A Method to Incorporate bounds in Data Reconciliation and Gross Error Detection - II. Gross Error Detection Strategies", *Computer in Chemical Engineering*, **17**(11), 1121-1128, (1993b)

Nogita, S., "Statistical Test and Adjustment of Process Data", *Industrial Engineering Chemical Process Design Development*, **11**(2), 197-200, (1972)

Pai, C.C.D. and Fisher, G.D., "Application of Broyden's Method to Reconciliation of Nonlinearly Constrained Data", *AIChE Journal*, **34**(5), 873-876, (1988)

Reilly, P.M. and Carpani, R.E., "Application of Statistical Theory of Adjustment to Material Balances", 13th Canadian Chemical Engineering Conference, Montreal, Quebec, (1963)

Ripps, D.L., "Adjustment of Experimental Data", *Chemical Engineering Progress Symposium Series*, **55**(61), 8-13, (1965)

Rollins, D.K. and Davis, J.F., "Unbiased Estimation of Gross Errors in Process Mesurements", *AIChE Journal*, **38**(4), 563-572, (1992)

Rollins, D.K. and Davis, J.F. "Gross Error Detection when Variance-Covariance Matrices are Unknown", *AIChE Journal*, **39**(8), 1335-1341, (1993)

Rollins, D.K. and Devanathan, S., "Unbiased Extimation in Dynamic Data Reconciliation", *Process Systems Engineering*, **39**(8), 1330-1334, (1993)

Rosenberg, J., Mah, R.S.H. and Iordache, C., "Evaluation of Schemes For Detecting Errors in Process Data", *Industrial Engineering Chemical Research*, **26**, 555-564, (1987)

Rosenbrock, H.H., "An Automated Method of Finding the Greatest or Least Value of a Function", *Computer Journal*, **3**, 175-184, (1960)

Serth, R.W. and Heenan, W.A., "Gross Error Detection and Data Reconciliation in Steam-Metering Systems", *AIChE Journal*, **32**(5), 733-742, (1986)

Serth, R.W., Weart, M.T. and Heenan, W.A., "On the Choice of Regression Variables in Nonlinear Data Reconciliation", *Chemical Engineering Communications*, **79**, 141-152, (1989)

Smith, H.W. and Ichiyen, N., "Computer Adjustment of Metallurigal Balances", *The Canadian Mining and Metallurgical - CIM Bulletin*, **66**, 97-100, (1973)

Smith, T.L., Personal Communication, 1995

Stanley, G.M. and Mah, R.S.H., "Observability and Redundancy in Process Data Estimation", *Chemical Engineering Science*, **36**, 259-272, (1981)

Stephenson, G.R. and Shewchuk, C.F., "Reconciliation of Process Data with Process Simulation", *AIChE Journal*, **32**(2), 247-254, (1986)

Swartz, C.L.E., "Data Reconciliation for Generalized Flowsheet Applications", Presented at the American Chemical Society Nation Meeting, Dallas Texas, April 10-14, (1989)

Tamhane, A.C., "A Note on the use of Residuals for Detecting an Outlier in Linear Regression", *Biometrika* **69**(2), 488-489, (1982)

Tamhane, A.C. and Mah, R.S., "Data Reconciliation and Gross Error Detection in Chemical Process Networks", *Technometrics*, **27**(4), 409-422, (1985)

Tjoa, I.B. and Biegler, L.T., "Simultaneous Strategies for Data Reconciliation and Gross Error Detection of Nonlinear Systems", *Computers and Chemical Engineering*, **15**(10), 679-690, (1991)

Václavek, V., "Studies on System Engineering - III Optimal Choice of the Balance

Measurements in Complicated Chemical Engineering Systems", *Chemical Engineering Science*, **24**, 947-955, (1969)

Verneuil, V.S., Yang, P. and Madron, F., "Banish Bad Plant Data", *Chemical Engineering Progress*, 47-51, October (1992)

Villadsen, J. and Michelsen, M.L., "Solution of Differential Equation Models by Polynomial Approximation", Prentice Hall, Englewood Cliffs, NJ, (1978)

Watkins, D.S., "*Fundamentals of Matrix Computations*", John Wiley & Sons, New York, (1991)

Wiegel, R.L., "Advances in Minerals Processing Material Balances", *Canadian Metallurgical Quarterly*, **11**(2), 413-424, (1972)

Willsky, A.S. and Jones, H.L., "A Generalized Likelihood Ratio Approach to State Estimation in Linear Systems Subject to Abrupt Changes", *Proceedings IEEE Conference of Decision and Control*, 846, (1974)

White, J.W., Winslow, R.L., and Rossiter, G.L., "A Useful Technique for Metallurgical Mass Balances Applications in Grinding", *International Journal of Mineral Processing*, **4**, 39-49, (1977)

Woollacott, L.C. and Strange, W., "Guidelines for the Derivation of Reliable Material Balances from Plant Data", *Journal of the South African Institute of Minning and Metallurgy*, **87**(7), 207-217, (1987)

APPENDIX A: Derivation of the formulae for the differentiation of a matrix inverse

Derivation of the formulae for differeniation of $\boldsymbol{C}\,\boldsymbol{\Sigma}\,\boldsymbol{C}^{T}$ with respect to σ_k, where $\boldsymbol{\Sigma}$ = diag $(\sigma_1, \ldots, \sigma_m)$

Let $A = C\ \Sigma\ C^T$

Let $B = A^{-1}$

$\therefore AA^{-1} = AB = I$

$$\frac{\partial}{\partial\sigma_k}AB = \frac{\partial}{\partial\sigma_k}I = 0$$

$$A\left[\frac{\partial}{\partial\sigma_k}B\right] + \left[\frac{\partial}{\partial\sigma_k}A\right]B = 0$$

$$A\left[\frac{\partial}{\partial\sigma_k}B\right] = -\left[\frac{\partial}{\partial\sigma_k}A\right]B$$

$$\frac{\partial}{\partial\sigma_k}B = -A^{-1}\left[\frac{\partial}{\partial\sigma_k}A\right]B$$

$$\frac{\partial}{\partial\sigma_k}A^{-1} = -A^{-1}\left[\frac{\partial}{\partial\sigma_k}A\right]A^{-1}$$

APPENDIX B: Problem data and m-files for the example network

MEASURED VARIABLES

All the variables are measured namely, A, B, C, D, E, F and G.

BALANCE EQUATIONS

NODE 1: A + F + D - B = 0

NODE 2: B - C = 0

NODE 3: C - D - E = 0

NODE 4: E - F - G = 0

Table B1 Matrix ***H*** representing the balance equations

	Measurements						
	A	B	C	D	E	F	G
NODE 1	1	-1	0	1	0	1	0
NODE 2	0	1	-1	0	0	0	0
NODE 3	0	0	1	-1	-1	0	0
NODE 4	0	0	0	0	1	-1	-1

```
%CALCULATE THE SENSITIVITY MATRIX FOR THE SIMPLE FLOW NETWORK

clear

%INPUT DATA - start
load c.dat
load y.dat
load s.dat
%INPUT DATA - stop

%INITIAL DATA RECONCILIATION - start
i=inv(c*s*c');
r=s*c'*i*c*y;
x=y-r;
%INITIAL DATA RECONCILIATION - stop

%SENSITIVITY ANALYSIS - start
for n=1:7;

     %VARIABLE INITIALISATION
     z = 4;
     %dVAR
     dVAR = zeros(7);
     dVAR(n,n) = 1;
     %dIA
     A = c*s*c';
     IA = inv(A);
     for i=1:z,
          for j=1:z,
               dA(i,j) = c(i,n)*c(j,n);,
          end,
     end
     dIA = -IA*dA*IA;
     dx = - (dVAR*c'*IA + s*c'*dIA) *c*y;
     DX(:,n)=dx;
end
%SENSITIVITY ANALYSIS - stop

%SCALE FOR EFFECT - start
for i=1:7;
     for j=1:7,
          DX(i,j) = DX(i,j) / y(i);
          DX(j,i) = DX(j,i) * s(i,i);,
     end,
end
%SCALE FOR EFFECT - stop
```

```
%EVALUATE CHANGES IN RECONCILIED FLOWRATES DUE TO THE
%CHANGES IN MEASUREMENT VARIANCE

clear
%INPUT DATA - start
load c.dat
load y.dat
load s.dat
%INPUT DATA - stop

m=7;    %Number of measurements
mn=1    %****Measurement under investigation****
Xct=11; %Total colums in X
j=2;    %Pointer in X
as=[0];

%INITIAL VALUES FOR THE RECONCILIED MEASUREMENTS - start
i=inv(c*s*c');
r=s*c'*i*c*y;
x=y-r;
X(:,1)=x;
%INITIAL VALUES FOR THE RECONCILIED MEASUREMENTS - stop

%EVALUATE CHANGES - start
for n=10:10:100,
     S=n/100;
     s(mn,mn)=(v(mn)*S)^2;
     as(n/10)=s(mn,mn);
     i=inv(c*s*c');
     r=s*c'*i*c*y;
     x=y-r;
     X(:,j)=x;
     j=j+1;
end
%EVALUATE CHANGES - stop

%CONVERT CHANGES IN RECONCILIED MEASUREMENTS TO % CHANGES - start
for k=1:m,
     for l=2:Xct,
          ll=l-1;
          FX(k,ll)=( X(k,1)-X(k,l) ) / X(k,1) * 100;,
     end,
end
%CONVERT CHANGES IN RECONCILIED MEASUREMENTS TO % CHANGES - stop
```

APPENDIX C: Problem data and m-files for the steam metering network

MEASURED VARIABLES

All the variables are measured namely, 1 to 28.

BALANCE EQUATIONS

(the numbers represent the measurement tags)

NODE 1: $1 + 2 + 4 - 3 = 0$

NODE 2: $7 + 8 - 5 - 6 - 9 = 0$

NODE 3: $5 - 1 - 10 = 0$

NODE 4: $10 + 11 - 12 = 0$

NODE 5: $3 + 13 - 11 - 14 - 15 - 16 - 17 = 0$

NODE 6: $6 - 2 - 13 = 0$

NODE 7: $14 + 18 - 7 - 19 - 20 - 21 = 0$

NODE 8: $15 + 22 - 18 - 23 - 24 = 0$

NODE 9: $12 + 16 - 22 - 25 = 0$

NODE 10: $19 + 23 + 27 - 26 = 0$

NODE 11: $20 + 26 + 28 - 8 = 0$

Table C1 Matrix H representing the balance equations

	Measurements																											
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	1	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	-1	-1	1	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	-1	0	0	0	1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	1	1	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	1	0	0	0	0	0	0	0	-1	0	1	-1	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0
6	0	-1	0	0	0	1	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	-1	0	0	0	0	0	0	1	0	0	0	1	-1	-1	-1	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	-1	0	0	0	1	-1	-1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	-1	0	0	-1	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	-1	1	0
11	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1

Table C2 True flowrates of Serth and Heenan network

Stream No.	Flow Rate 1E3 [kg/h]
1	0.86
2	1.00
3	111.82
4	109.95
5	53.27
6	112.27
7	2.32
8	164.05
9	0.86
10	52.41
11	14.86
12	67.27
13	111.27
14	91.86
15	60.00
16	23.64
17	32.73
18	16.23
19	7.95
20	10.50
21	87.27
22	5.45
23	2.59
24	46.64
25	85.45
26	81.32
27	70.77
28	72.23

Table C3 Scaled sensitivity matrix for the Serth and Heenan network

	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14
x1	-0.1646	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x2	0.0000	-0.0254	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x3	-0.0748	-0.0151	0.0449	0.0224	0.0094	-0.0078	-0.0018	-0.0039	-0.0026	-0.0043	-0.0219	-0.0014	-0.0118	-0.0178
x4	0.0899	0.0103	0.0449	0.0224	0.0094	-0.0078	-0.0018	-0.0039	-0.0026	-0.0043	-0.0218	-0.0014	-0.0118	-0.0178
x5	-0.1015	0.0010	0.0014	0.0007	0.0689	-0.0025	-0.0070	0.0018	0.0012	-0.0313	0.0236	-0.0061	-0.0037	0.0009
x6	0.0551	-0.0080	-0.0141	-0.0071	-0.0296	0.0175	-0.0470	0.0082	0.0053	0.0134	-0.0228	0.0023	0.0263	-0.0098
x7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.1002	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x8	-0.0464	-0.0070	-0.0127	-0.0064	0.0393	0.0150	0.0462	0.0100	-0.0080	-0.0178	0.0008	-0.0037	0.0226	-0.0089
x9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	-0.0145	0.0000	0.0000	0.0000	0.0000	0.0000
x10	0.0631	0.0010	0.0014	0.0007	0.0689	-0.0025	-0.0070	0.0018	0.0012	-0.0313	0.0236	-0.0061	-0.0037	0.0009
x11	-0.0048	0.0000	0.0007	0.0004	-0.0052	0.0004	0.0006	0.0000	0.0000	0.0024	-0.0732	-0.0010	0.0006	0.0005
x12	0.0583	0.0010	0.0021	0.0011	0.0637	-0.0020	-0.0064	0.0018	0.0012	-0.0289	-0.0496	-0.0071	-0.0031	0.0014
x13	0.0551	0.0174	-0.0141	-0.0071	-0.0296	0.0175	-0.0470	0.0082	0.0053	0.0134	-0.0228	0.0023	0.0263	-0.0098
x14	-0.0063	0.0013	0.0174	0.0087	-0.0058	0.0053	-0.0497	0.0026	0.0016	0.0026	0.0144	0.0009	0.0080	-0.0376
x15	-0.0022	0.0005	0.0060	0.0030	-0.0022	0.0018	-0.0005	0.0009	0.0006	0.0010	0.0050	0.0003	0.0027	0.0050
x16	-0.0049	0.0001	0.0021	0.0010	-0.0053	0.0008	0.0006	0.0002	0.0001	0.0024	0.0054	0.0006	0.0012	0.0014
x17	-0.0016	0.0004	0.0046	0.0023	-0.0016	0.0014	0.0003	0.0007	0.0004	0.0007	0.0038	0.0002	0.0021	0.0031
x18	0.0000	0.0000	0.0001	0.0001	0.0000	0.0000	-0.0015	0.0000	0.0000	0.0000	0.0001	0.0000	0.0001	0.0016
x19	-0.0002	0.0000	0.0001	0.0000	0.0001	0.0001	0.0006	0.0001	0.0000	-0.0001	0.0001	0.0000	0.0002	-0.0003
x20	-0.0007	-0.0001	0.0001	0.0000	0.0004	0.0003	0.0013	0.0002	-0.0001	-0.0002	0.0002	0.0000	0.0004	-0.0006
x21	-0.0054	0.0015	0.0173	0.0086	-0.0064	0.0049	0.0471	0.0024	0.0017	0.0029	0.0141	0.0009	0.0074	-0.0350
x22	0.0002	0.0000	-0.0001	0.0000	0.0003	0.0000	0.0000	0.0000	0.0000	-0.0001	-0.0002	0.0000	0.0000	0.0000
x23	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x24	-0.0019	0.0005	0.0058	0.0029	-0.0020	0.0017	0.0009	0.0008	0.0006	0.0009	0.0047	0.0003	0.0026	0.0033
x25	0.0532	0.0011	0.0043	0.0021	0.0581	-0.0012	-0.0058	0.0020	0.0013	-0.0264	-0.0440	-0.0065	-0.0018	0.0028
x26	-0.0162	-0.0025	-0.0045	-0.0022	0.0138	0.0052	0.0161	0.0035	-0.0028	-0.0063	0.0003	-0.0013	0.0079	-0.0030
x27	-0.0160	-0.0024	-0.0046	-0.0023	0.0136	0.0051	0.0155	0.0034	-0.0028	-0.0062	0.0001	-0.0013	0.0077	-0.0027
x28	-0.0294	-0.0045	-0.0083	-0.0041	0.0251	0.0095	0.0288	0.0063	-0.0051	-0.0114	0.0003	-0.0024	0.0143	-0.0052
1-NORM	0.8519	0.1012	0.2117	0.1056	0.4583	0.1102	0.4337	0.0626	0.0589	0.2084	0.3527	0.0463	0.1660	0.1656
RANK	6	25	18	24	11	22	13	26	27	19	14	28	21	20

	s15	s16	s17	s18	s19	s20	s21	s22	s23	s24	s25	s26	s27	s28
x1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x3	0.0545	-0.0358	-0.0802	-0.0164	-0.0485	-0.0114	-0.0224	-0.0882	-0.1354	-0.1370	-0.0053	-0.0129	0.0237	0.0110
x4	0.0545	-0.0358	-0.0802	-0.0164	-0.0485	-0.0114	-0.0224	-0.0882	-0.1354	-0.1370	-0.0053	-0.0129	0.0237	0.0110
x5	-0.0031	0.0136	0.0042	0.0004	-0.0101	-0.0100	0.0012	0.0563	-0.0210	0.0070	-0.0107	0.0059	-0.0106	-0.0050
x6	0.0292	-0.0248	-0.0426	-0.0100	-0.1078	-0.0746	-0.0115	-0.0724	-0.2517	-0.0724	0.0027	0.0272	-0.0479	-0.0227
x7	0.0000	0.0000	0.0000	-0.0001	0.0002	0.0001	0.0000	0.0000	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000
x8	0.0261	-0.0113	-0.0384	-0.0095	-0.1181	-0.0847	-0.0103	-0.0161	-0.2729	-0.0654	-0.0080	0.0331	-0.0585	-0.0277
x9	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
x10	-0.0031	0.0136	0.0042	0.0004	-0.0101	-0.0100	0.0012	0.0563	-0.0210	0.0070	-0.0107	0.0059	-0.0106	-0.0050
x11	-0.0015	0.0030	0.0022	0.0004	0.0022	0.0011	0.0006	0.0115	0.0056	0.0037	-0.0018	0.0000	0.0000	0.0000
x12	-0.0046	0.0166	0.0064	0.0008	-0.0079	-0.0089	0.0018	0.0678	-0.0155	0.0106	-0.0125	0.0059	-0.0105	-0.0050
x13	0.0292	-0.0248	-0.0426	-0.0100	-0.1078	-0.0746	-0.0115	-0.0724	-0.2517	-0.0724	0.0027	0.0272	-0.0479	-0.0227
x14	-0.0441	0.0234	0.0523	0.2472	-0.1783	-0.0914	-0.0441	0.0469	0.0737	0.0761	0.0034	0.0085	-0.0137	-0.0067
x15	0.1432	0.0083	0.0180	-0.2781	0.0072	0.0008	0.0039	-0.2717	-0.5021	-0.3236	0.0010	0.0029	-0.0051	-0.0024
x16	-0.0044	-0.1016	0.0063	0.0012	0.0043	0.0016	0.0018	0.0374	0.0122	0.0106	-0.0061	0.0007	-0.0012	-0.0006
x17	-0.0095	0.0062	-0.2017	0.0029	0.0085	0.0020	0.0039	0.0154	0.0236	0.0239	0.0009	0.0022	-0.0041	-0.0019
x18	0.0161	0.0001	0.0003	-0.4549	-0.0061	-0.0029	-0.0017	0.0213	0.0390	0.0261	0.0000	0.0000	-0.0001	0.0000
x19	-0.0001	0.0001	0.0003	-0.0017	-0.4114	0.0013	0.0005	0.0005	0.0067	0.0005	0.0000	0.0010	0.0020	0.0002
x20	0.0000	0.0002	0.0002	-0.0030	0.0047	-0.1924	0.0008	0.0007	0.0046	0.0006	-0.0001	-0.0007	0.0013	0.0006
x21	-0.0278	0.0232	0.0522	-0.2028	0.2215	0.0966	-0.0472	0.0670	0.1012	0.1011	0.0035	0.0083	-0.0170	-0.0076
x22	-0.0017	-0.0004	-0.0002	-0.0024	-0.0002	-0.0001	-0.0001	0.4891	-0.0048	-0.0032	0.0003	0.0000	0.0000	0.0000
x23	0.0004	0.0000	0.0000	0.0005	0.0003	0.0001	0.0000	0.0006	-0.8972	0.0007	0.0000	0.0001	0.0002	0.0000
x24	0.1250	0.0077	0.0175	0.1739	0.0128	0.0036	0.0055	0.1954	0.3513	-0.3537	0.0013	0.0028	-0.0053	-0.0024
x25	-0.0072	-0.0845	0.0129	0.0043	-0.0030	-0.0072	0.0037	-0.3839	0.0015	0.0245	-0.0189	0.0066	-0.0118	-0.0056
x26	0.0094	-0.0040	-0.0135	-0.0027	-0.1947	0.0385	-0.0037	-0.0056	-0.4258	-0.0228	-0.0028	0.0739	-0.1312	0.0185
x27	0.0091	-0.0041	-0.0138	-0.0015	0.2164	0.0372	-0.0043	-0.0066	0.4648	-0.0241	-0.0028	0.0728	-0.1333	0.0183
x28	0.0168	-0.0075	-0.0251	-0.0037	0.0718	0.0691	-0.0074	-0.0113	0.1482	-0.0431	-0.0051	-0.0400	0.0714	-0.0468
1-NORM	0.6207	0.4509	0.7156	1.4455	1.8035	0.8315	0.2117	2.0827	4.1671	1.5471	0.1057	0.3516	0.6310	0.2218
RANK	10	12	8	5	3	7	17	2	1	4	23	15	9	16

Entry (x1,s1) is the differential of the first variable w.r.t the first variance

```
%CALCULATE THE SENSITIVITY MATRIX FOR THE STEAM METERING NETWORK

clear

%DATA INPUT - start
load c.dat
load s.dat
load y.dat
nm=28
n1=0;
%DATA INPUT - stop

%INITIAL DATA RECONCILIATION - start
i=inv(c*s*c');
r=s*c'*i*c*y;
x=y-r;
%INITIAL DATA RECONCILIATION - stop

%SENSITIVITY ANALYSIS - start
for n=1:nm;

     %VARIABLE INITIALISATION
     z = 11;
     %dVAR
     dVAR = zeros(nm);
     dVAR(n,n) = 1;
     A = c*s*c';
     IA = inv(A);
     for i=1:z,
          for j=1:z,
               dA(i,j) = c(i,n)*c(j,n);,
          end,
     end
     dIA = -IA*dA*IA;
     dx = - (dVAR*c'*IA + s*c'*dIA) *c*y;
     DX(:,n)=dx;;
end
%SENSITIVITY ANALYSIS - stop

%SCALE FOR EFFECT - start
for i=1:nm,
     for j=1:nm,
          DX(i,j) = DX(i,j) / x(i);
          DX(j,i) = DX(j,i) * s(i,i);,
     end,
end
%SCALE FOR EFFECT - stop

%CALCULATE THE 1-NORM OF THE SENSITIVITY MATRIX - start
for n=1:nm, n1=n1+abs(DX(n,:));, end
DX(nm+1,:)=n1;
[S,I]=sort(n1);
for n=1:nm, DX(nm+2,I(n))=-(n-(nm+1));, end
%CALCULATE THE 1-NORM OF THE SENSITIVITY MATRIX - stop
```

```
%EVALUATE THE CHANGES IN RECONCILIED FLOWRATES DUE TO CHANGES
%IN THE MEASUREMENT VARIANCES

clear

%INPUT DATA - start
load c.dat
load y.dat
load s.dat
load ytrue.dat
load I.dat
%INPUT DATA - stop

m=28;   %Number of measurements
Xct=6;  %Total colums in X, initial plus 5-85% step 25%

%INITIAL RECONCILIATION - start
i=inv(c*s*c');
r=s*c'*i*c*y;
x=y-r;
%INITIAL RECONCILIATION - stop

%EVALUATE CHANGES - start
X(:,1)=x;
mn=23;          %Measurement under investigation
j=2;            %Pointer in X
for n=5:20:85,
  S=n/100;                    %Convert counter to percentage
  s(mn,mn)=(ytrue(mn)*S)^2;   %Adjust variance under investigation
  i=inv(c*s*c');              %Reconcile START
  r=s*c'*i*c*y;
  x=y-r;                      %Reconcile END
  X(:,j)=x;          %Place 'new' values in X
  j=j+1;                      %Increment counter
end
s(mn,mn)=(0.025*ytrue(mn))^2; %Reset var to 2.5% of flow rate
%EVALUATE CHANGES - stop

%CONVERT RECONCILED FLOWS INTO PERCENTAGE CHANGES - start
for k=1:m,
  for l=2:Xct,
    ll=l-1;
    FX(k,ll)=( X(k,1)-X(k,l) ) / X(k,1) * 100;,
  end,
end
%CONVERT RECONCILED FLOWS INTO PERCENTAGE CHANGES - stop
```

APPENDIX D: Problem data and m-files for the NH_3 synthesis network

Table D1 True values of the measurements.

N_2^1	H_2^1	Ar^1	N_2^2	Ar^2	N_2^3	NH_3^4	H_2^5
33.00	89.00	0.40	101.00	20.20	69.00	62.00	205.00

MEASURED VARIABLES

Stream 1: N_2^1 , H_2^1 , Ar^1

Stream 2: N_2^2 , Ar^2

Stream 3: N_2^3

Stream 4: NH_3^4

Stream 5: H_2^5

UNMEASURED VARIABLES

Stream 2: H_2^2

Stream 3: H_2^3 NH_3^3 Ar^3

Stream 5: N_2^5 Ar^5

Stream 6: N_2^6 H_2^6 Ar_2^6

Stream 7: N_2^7 H_2^7 Ar^7

Node 2(REACTOR): E

CONSTRAINT EQUATIONS (for i components)

NODE 1: (Mixer)

$$F_2 - F_1 - F_7 = 0 \quad \text{or}$$

$$F_{2,i} - F_{1,i} - F_{7,i} = 0$$

NODE 2: (Reactor)

$$F_3 - F_2 - m^T\xi = 0 \quad \text{or}$$

$$F_{3,i} - F_{2,i} - \Sigma_{j=1}^{J} \nu_{j,i}\, \xi_j = 0$$

NODE 3: (Separator)

$$F_5 - F_4 - F_3 = 0 \quad \text{or}$$

$$F_{5,i} - F_{4,i} - F_{3,i} = 0$$

NODE 4: (Splitter)

$$F_6 - F_7 - F_5 = 0 \quad \text{or}$$

$$F_{6,i} - F_{7,i} - F_{5,i} = 0$$

SPLITTER CONDITIONS:

$F_6 - \% F_5 = 0$ or

$F_{6,I} - \% F_{5,I} = 0$

Table D2 Matrix ***A*** representing the linearised constraint equations with respect to the measured variables.

		S 1			S 2		S 3	S 4	S 5
		N_2	H_2	Ar	N_2	Ar	N_2	NH_3	H_2
N 1	N_2	-1	0	0	1	0	0	0	0
	H_2	0	-1	0	0	0	0	0	0
	Ar	0	0	-1	0	1	0	0	0
N 2	N_2	0	0	0	-1	0	1	0	0
	H_2	0	0	0	0	0	0	0	0
	NH_3	0	0	0	0	0	0	0	0
	Ar	0	0	0	0	-1	0	0	0
N 3	N_2	0	0	0	0	0	-1	0	0
	H_2	0	0	0	0	0	0	0	1
	NH_3	0	0	0	0	0	0	1	0
	Ar	0	0	0	0	0	0	0	0
N 4	N_2	0	0	0	0	0	0	0	0
	H_2	0	0	0	0	0	0	0	-1
	Ar	0	0	0	0	0	0	0	0
SC	N_2	0	0	0	0	0	0	0	0
	H_2	0	0	0	0	0	0	0	-%
	Ar	0	0	0	0	0	0	0	0

Table D3 Matrix ***B*** representing the linearised constraint equations with respect to the unmeasured variables.

		S 2	S 3			S 5		S 6			S 7			E
		H_2	H_2	NH_3	Ar	N_2	Ar	N_2	H_2	Ar	N_2	H_2	Ar	E
N 1	N_2	0	0	0	0	0	0	0	0	0	-1	0	0	0
	H_2	1	0	0	0	0	0	0	0	0	0	-1	0	0
	Ar	0	0	0	0	0	0	0	0	0	0	0	-1	0
N 2	N_2	0	0	0	0	0	0	0	0	0	0	0	0	1
	H_2	-1	1	0	0	0	0	0	0	0	0	0	0	3
	NH_3	0	0	1	0	0	0	0	0	0	0	0	0	-2
	Ar	0	0	0	1	0	0	0	0	0	0	0	0	0
N 3	N_2	0	0	0	0	1	0	0	0	0	0	0	0	0
	H_2	0	-1	0	0	0	0	0	0	0	0	0	0	0
	NH_3	0	0	-1	0	0	0	0	0	0	0	0	0	0
	Ar	0	0	0	-1	0	1	0	0	0	0	0	0	0
N 4	N_2	0	0	0	0	-1	0	1	0	0	1	0	0	0
	H_2	0	0	0	0	0	0	0	1	0	0	1	0	0
	Ar	0	0	0	0	0	-1	0	0	1	0	0	1	0
SC	N_2	0	0	0	0	-%	0	1	0	0	0	0	0	0
	H_2	0	0	0	0	0	0	0	1	0	0	0	0	0
	Ar	0	0	0	0	0	-%	0	0	1	0	0	0	0

The '%' represents the split fraction at the splitter and equals 0.02003

```
% CALCULATE THE SENSITIVITY MATRIX AND EVALULATE THE CHANGE IN
% THE MEASUREMENT TEST STATISTIC

clear
%INPUT DATA - start
load a.dat
load b.dat
load y.dat
load sd.dat
s=sd;
nm=8;
nu=13;
%INPUT DATA - stop

nv=nu+nm;
S=s;
z=y;
ZI=zeros(nm,1);

%QR DECOMPOSITION & DECOUPLE UNMEASURED VARS. - start
[q,r11]=qr(b);
q1=q(:,1:13);
q2=q(:,14:17);
r11=r11(1:13,:);
C=q2'*a;
%QR DECOMPOSITION & DECOUPLE UNMEASURED VARS. - stop

%DATA RECONCILIATION - start
i=inv(C*S*C');
r=S*C'*i*C*z;
x=z-r;
[C_rn,C_cn]=size(C);
%DATA RECONCILIATION - stop

%SENSITIVITY ANALYSIS - start
c=C;
s=S;
for n=1:nm
     %dVAR
     dVAR = zeros(8);
     dVAR(n,n) = 1;

     %dIA
     A = c*s*c';
     IA = inv(A);
     for i=1:C_rn,
          for j=1:C_rn,
               dA(i,j) = c(i,n)*c(j,n);,
          end,
     end
     dIA = -IA*dA*IA;

     dx = - (dVAR*c'*IA + s*c'*dIA) *c*y;
     DX(:,n)=dx;;
end
DXus=DX;
%
%
%%SCALE SENSITIVITY MATRIX - start
for i=1:nm;
```



```
     for j=1:nm,
          DX(i,j) = DX(i,j) / x(i);
          DX(j,i) = DX(j,i) * S(i,i);,
     end,
end
%%SCALE SENSITIVITY MATRIX - stop
sum=0;
for n=1:nm, sum=sum+abs(DX(n,:));, end
[ss,ii]=sort(sum);
DXout=ii';
DXout(:,2)=ss';
%SENSITIVITY ANALYSIS - stop

%EVALUATE EFFECT OF MEASUREMENT TEST STATISTIC - start
i=1;                    %insert measurement index
index=[mi];
for nnm=1:
  m=index(nnm);
  nn=1;
  for nnn=0.75:0.05:1.25
    S(m,m)=nnn*S(m,m);
    i=inv(C*S*C');
    r=S*C'*i*C*z;
    x=z-r;
    z_num=inv(S)*r;
    z_den=C'*i*C;
    for n=1:nm
      zi(n,1)=abs(z_num(n)) / sqrt(z_den(n,n));
    end,
    S(m,m)=S(m,m)/nnn;
    ZI(:,nn)=zi;
    nn=nn+1;
  end,
end
%EVALUATE EFFECT ON MEASUREMENT TEST STATISTIC - stop
```

```
%EVALUATE % CHANGE IN RECONCILIED FLOWRATES DUE
%TO CHANGES IN MEASUREMENT VARIANCE

clear
%INPUT DATA - start
load a.dat
load b.dat
load y.dat
load s.dat
%INPUT DATA - stop

sold=s;
Xct=12;          %Total colums in X, initial plus 5-85% step 25%
ri=1;            %Initial row pointer in X
mn = 8;          %Number of measurements
umn = 13;                %Number of unmeasurements
q2mrk = umn + 1;         %Marker for form. of Q2
vertmrk = 17;            %Vertical marker in Q, = nodes * components
norml = zeros(8,1);
sorted = zeros(8,1);

%QR DECOMPOSITION & DECOUPLE UNMEASURED VARS. - start
[q,r]=qr(b);
q1=q(:,1:umn);
q2=q(:,q2mrk:vertmrk);  %q a sqr of ( vertmrk x vertmrk )
r11=r(1:umn,1:umn);
c=q2'*a;
%QR DECOMPOSITION & DECOUPLE UNMEASURED VARS. - stop

%EVALUATE %CHANGE - start
I=[1:mn];
for itern=1:mn,
iter=I(itern);
rt=ri+mn-1;
j=2;
%%Initial Reconciliation - start
i=inv(c*s*c');
r=s*c'*i*c*y;
x=y-r;
X(ri:rt,1)=x;
%%Initial Reconciliation - stop
%
for n=0.75:0.05:1.25,
     s(iter,iter)=s(iter,iter)*n;       %Adjust variance under investigation
     i=inv(c*s*c');              %Reconcile START
     r=s*c'*i*c*y;
     x=y-r;                      %Reconcile END
     X(ri:rt,j)=x;               %Place 'new' values in X
     j=j+1;                      %Increment counter
     s(iter,iter)=s(iter,iter)/n;
end
%
%%CONVERT RECONCILIED FLOWS INTO % CHANGES - start
for k=ri:rt,
     for l=2:Xct,
          ll=l-1;
          FX(k,ll)=( X(k,1)-X(k,1) ) / X(k,1) * 100;,
     end,
end
%%CONVERT RECONCILIED FLOWS INTO % CHANGES - stop
```



```
%
ri=rt+2;
end
%EVALUATE %CHANGE - stop
```

APPENDIX E: Algorithms, m-files and data for measurement structure synthesis

GSA Algorithm

In this algorithm there are a number of important matrices:

1) ***OXD_P*** stores all the previously evaluated 'good'measurement structures.
2) ***OXD*** stores the 'good'measurement structures evaluated in the current generation.
3) Γ stores all the previously evaluated 'bad'measurement structures.
4) ***RXD*** stores the 'bad'measurement structures evaluated in the current generation.
5) ***VAR*** variable classification constraint matrix.

The GSA Algorithm:

1) Initialize the variables: Set ***OXD***, ***OXD_P***, ***RXD*** and $\Gamma = 0$, $n_d=1$
2) LOAD ***VAR***, ***H***, n_m
3) IF the $n_d = 1$,

   FOR all i in n_m

   IF ***VAR***$(i,1) \neq 1$, place i in ***OXD_P***

   NEXT

   ELSE set ***OXD_P*** = ***OXD***
4) FOR each measurement structure in the matrix, ***OXD_P.***

4.1) Assign a vector y_j^* for the current measurement structure.

4.2) IF $n_d = 1$, let $y_j = y_j^*$, ELSE FOR each possible deletion of n_d measurements based on y_j. Form y_j from y_j^*.[1]

4.2.1) IF $n_d \neq 1$, IF y_j contains a variable constrained as measured GOTO 4.2).

4.2.2) IF $n_d \neq 1$ test y_j against Γ for previously evaluated 'bad'subsets of measurement structures, using integer cuts. IF y_j fails the integer cut criterion, GOTO 4.2).

4.2.3) Form the ***A*** and ***B*** matrices, corresponding to the measurement structure described in y_j, from the balance equations H. Form ***C***, ***Q***, ***R*** from ***A*** and ***B***.[2]

4.2.4) Check that the measurements specified in ***VAR*** are redundant. IF these constraints are satisfied GOTO 4.2.5), ELSEIF the measurement structure in y_j fails this test, place the indices of y_j in a column in ***RXD*** and GOTO 4.2).[2]

4.2.5) Check that the measurements specified in ***VAR*** have a degree of redundancy of one. IF these constraints are satisfied GOTO 4.2.6), ELSEIF the measurement structure in y_j fails this test, place the indices of y_j in a column

in ***RXD*** and GOTO 4.2).[2]

4.2.6) Check that the variables specified in ***VAR*** are observable. IF these constraints are satisfied, place the indices of y_j in ***OXD*** assign a cost to this measurement structure and GOTO 4.2), ELSEIF the measurement structure in y_j fails this test, place the indices of y_j in a column in ***RXD*** and GOTO 4.2).

4.3) IF $n_d = 1$ let Γ = ***RXD***, ELSE append ***RXD*** onto Γ.

4.4) IF ***OXD*** is empty STOP. The solution set is ***OXD_P***, ELSE set $n_d = n_d + 1$ and GOTO 3).

[1] The method used to perform this step was presented in Section 4.1.1.

[2] The formulation of the matrices describing the new measurement structure and the variable classification have been discussed in Sections 2.1.4., 2.3, and 4.1.3.

LSA Algorithm

In this algorithm there are a number of important matrices:

1) ***OXD_P*** stores all the previously evaluated 'good'measurement structures.
2) ***OXD*** stores the 'good'measurement structures evaluated in the current generation.
3) Γ stores all the previously evaluated 'bad'measurement structures.
4) ***RXD*** stores the 'bad'measurement structures evaluated in the current generation.
5) ***VAR*** variable classification constraint matrix.
6) ***NREDNM*** stores the list of nonredundant measurements.
7) ***MEAS*** stores the indices of the measurements.
8) ***UNMEAS*** stores the indices of the unmeasured variables.
9) ***NDOR*** stores the list of measurements that are constrained to have a set degree of redundancy.

The LSA Algorithm:

1) Load ***H***, ***VAR***
2) Evaluate the number of balance equations, z, the number of rows in ***H***.
3) Form the initial measurement structure with z - 1 unmeasured variables such that the rank of ***B*** equals the number of columns present, and store in vector y_j^i
4) Perform the initial variable classification on the measurement structure defined by

y_j^i, forming a list of nonredundant measurements, ***NREDNM***.

5) Find all the nonzeros entries in y_j^i and store as a list of the variables ***UNMEAS***. Set the number of unmeasured variables $n_u = z - 1$.

6) IF ***NREDNM*** is NOT empty, let $y_j = y_j^i$.

6.1) Find the combination of unmeasured variables in ***UNMEAS*** whose addition to the measurement structure removes the nonredundancies in the measurements listed in ***NREDNM***, let y_j^1 denote this measurement structure. IF no such selection exists GOTO 3).

7) From the second column of ***VAR*** form the list ***NDOR*** of measurements which are specified to have a degree of redundancy equal to one.

8) IF ***NDOR*** is NOT empty, let $y_j = y_j^1$.

8.1) Set ***UNMEAS***=0 AND find all the nonzeros entries in y_j and store as a list of the variables in ***UNMEAS***.

8.2) Find the combination of unmeasured variables in ***UNMEAS*** that results in all the measurements in ***NDOR*** having the correct degree of redundancy. Let y_j^2 equal the corresponding measurement structure.

9) Calculate the indices of all the measurements in y_j^2 and store in ***MEAS***. Let n_m equal the number of elements in ***MEAS***.

10) Set ***OXD***, ***OXD_P***, ***RXD*** and $\Gamma = 0$, n_d=1

11) IF the $n_d = 1$,

FOR all i in n_m

IF ***VAR***(***MEAS***(i),1) $\neq$ 1, place ***MEAS***(i) in ***OXD_P***

NEXT

ELSE set ***OXD_P*** = ***OXD***

12) FOR each measurement structure in the matrix, ***OXD_P***.

12.1) Assign a vector y_j^* for the current measurement structure.

12.2) IF $n_d = 1$, let $y_j = y_j^*$, ELSE FOR each possible deletion of n_d measurements based on y_j. Form y_j from y_j^*.[1]

12.2.1) IF $n_d \neq 1$, IF y_j contains a variable constrained as measured GOTO 12.2).

12.2.2) IF $n_d \neq 1$ test y_j against Γ for previously evaluated 'bad'subsets of measurement structures, using integer cuts. IF y_j fails the integer cut criterion, GOTO 12.2).

12.2.3) Form the ***A*** and ***B*** matrices, corresponding to the measurement structure

described in y_j, from the balance equations H. Form $\boldsymbol{C}$, $\boldsymbol{Q}$, $\boldsymbol{R}$ from $\boldsymbol{A}$ and $\boldsymbol{B}$.[2]

12.2.4) Check that the measurements specified in ***VAR*** are redundant. IF these constraints are satisfied GOTO 4.2.5), ELSEIF the measurement structure in y_j fails this test, place the indices of y_j in a column in ***RXD*** and GOTO 12.2).[2]

12.2.5) Check that the measurements specified in ***VAR*** have a degree of redundancy of one. IF these constraints are satisfied GOTO 12.2.6), ELSEIF the measurement structure in y_j fails this test, place the indices of y_j in a column in ***RXD*** and GOTO 12.2).[2]

12.2.6) Check that the variables specified in ***VAR*** are observable. IF these constraints are satisfied, place the indices of y_j in ***OXD*** assign a cost to this measurement structure and GOTO 12.2), ELSEIF the measurement structure in y_j fails this test, place the indices of y_j in a column in ***RXD*** and GOTO 12.2).

12.3) IF $n_d = 1$ let $\boldsymbol{\Gamma} = \boldsymbol{RXD}$, ELSE append ***RXD*** onto $\boldsymbol{\Gamma}$.

12.4) IF ***OXD*** is empty STOP. The solution set is ***OXD_P***, ELSE set $n_d = n_d + 1$ and GOTO 11).

[1] The method used to perform this step was presented in Section 4.1.1.

[2] The formulation of the matrices describing the new measurement structure and the variable classification have been discussed in Sections 2.1.4., 2.3, and 4.1.3.

```
% PROGRAM LISTING FOR THE GLOBAL SEARCH ALGORITHM
clear
tic
%DATA INPUT - start
load c.dat
load VAR.dat
%DATA INPUT - stop
%VARIABLE INITIALISATION - start
nm=24;                    %Number of measurements
nodes=11;                 %Number of Nodes
nd_max=nodes-1;           %Max number of deleted measurements
T_QR=0;                   %Time counter for QR decomps.
FCOST=0;
oxd_p_rn=1;
nd=1
end_of_line=0;
FCOST=0;
ucalc=find(VAR(:,4));
ucalc_rn=nnz(ucalc);
cost=[ones(size(1:nm))];        %Unit cost function
%VARIABLE INITIALISATION - stop
%GSA - start
while end_of_line ==0
  rxd=zeros(1,nd);        %*  Initialise 'bad and good measurement structures'
  oxd=zeros(1,nd+1);      %** rxd and oxd respectively
  nm_iter=nm-nd;          %Number of measurements in this iteration
  dxREDN=0;               %*  Flags marking 'bad and good measurement
  dxOBJ=0;                %** structures'.
  if nd==1,                       %*  For the initial case when two measurements
    for n=1:nm                    %*  insert into oxd all the variables that
      if VAR(n,1)~=1              %*  may be deleted, ie those not specified
        oxd_p(oxd_p_rn,1)=n;      %*  to be measured.
        oxd_p_rn=oxd_p_rn+1;      %*
      end,                        %*
    end,                          %*
  end,
  [nxd_f,dummy]=size(oxd_p);      %Size of 'previous good measurement structures'
  for nxd=1:nxd_f                 %For all 'previous good measurement structures'
    if nd==1
      yj(oxd_p(nxd),1)=1;
      [A,B,Ia,Ib]=ab(c,nodes,yj,nm,nd);
      [NREDN,OBJ,rxd]=reval(c,Ia,Ib,VAR,nm,A,B,nodes,md,rxd,yj,ucalc,ucalc_rn);
      if OBJ==1 %*  Flag showing that 'nd' deletions has one or
        dxOBJ=1;          %** more good measurement structures.
        [oxd]=ceval(yj,Ia,cost,nm_iter,nd,oxd); %Eval the cost
      end,
      if NREDN==1         %*  Flag showing that 'nd' deletions has one or
        dxREDN=1;         %** more bad measurement structures.
      end,
      yj(oxd_p(nxd),1)=0;
    else
      yj=zeros(nm,1);
      for n=1:nd-1,               %*  Place 'previous good measurement
        yj(oxd_p(nxd,n),1)=1;     %*  structures' into 'deletion vector'
      end,                        %** yj
      yj_indx=find(yj);
      for n=yj_indx(nd-1,1)+1:nm        %Start from last measurement in yj + one
        if VAR(n,1)~=1
          yj(n,1)=1;
          SKIP=icut(yj,Qi);, end                %*  Check for subsets of bad
```



```
          if SKIP==0                    %** measurement structure
            [A,B,Ia,Ib]=ab(c,nodes,yj,nm,nd);
            [NREDN,OBJ,rxd]=reval(c,Ia,Ib,VAR,nm,A,B,nodes,nd,rxd,yj,ucalc,ucalc_
rn);   %Test for NREDN meas.
            if OBJ==1   %*  Flag showing that 'nd' deletions has one or
              dxOBJ=1;  %** more good measurement structures.
              [oxd]=ceval(yj,Ia,cost,nm_iter,nd,oxd);   %Eval the cost
            end,
            if NREDN==1 %*  Flag showing that 'nd' deletions has one or
              dxREDN=1; %** more bad measurement structures.
            end,
          end,
          yj(n,1)=0;
        end,
      end,
    end,
  end,
  if dxREDN==1
    if Qi(1)==0           %*  For the initial case of 2 deletions
      Qi=rxd';            %** initialise Qi
    else
      [Qi_rn,Qi_cn]=size(Qi);                   %*  Update Qi by inserting
      [rxd_rn,rxd_cn]=size(rxd);                %*  rxd onto the end of Qi
      Qi_trans=zeros(nd,1);                     %*  Qi is expanded (rows
      Qi_trans(1:Qi_rn,1:Qi_cn)=Qi;             %*  and columns) by this
      Qi_trans(:,Qi_cn+1:Qi_cn+rxd_rn)=rxd';    %** addition
      Qi=Qi_trans;
    end,
  end,
  if dxOBJ==1;
    save oxd.dat oxd -ascii
    save oxd_p.dat oxd_p -ascii
    oxd_p=oxd;          %Set 'previous good measurement structure' to rxd
    [S,I]=sort(oxd(:,nd+1));            %*  Find optimum for current nd del.
    FCOST(nd,1)=oxd(I(1),nd+1);         %*
    FCOST(nd,2:nd+1)=oxd(I(1),1:nd);    %** Store current optimum.
  else
    end_of_line=1;
  end,
  nd=nd+1
end
%GSA - stop
toc

%SUBROUTINES

function [A,B,Ia,Ib]=ab(c,nodes,yj,nm,nd)
%AB  Splits the c matrix into matrices A and B representing
%    measured and unmeasured variables respectively
%
%    c is the original incidence matrix
%    Yj is a vector representing while measurements are to be
%    deleted
%    nd is the number of measurements been deleted
%    nm is the number of measurements
Ia=zeros(nm,nm-nd);
Ib=zeros(nm,nd);
Ia_cn=1;
Ib_cn=1;
for n=1:nm
```

```
  if yj(n,1)==0          %*  If the variable is measured place an
    Ia(n,Ia_cn)=1;       %*  one in the corresponding place in Ia
    Ia_cn=Ia_cn+1;       %*  and increment the column counter in Ia
  else                   %*
    Ib(n,Ib_cn)=1;       %*  else do the same for Ib representing
    Ib_cn=Ib_cn+1;       %** the unmeasured variables
  end,
end,
A=c*Ia;          %*  Form A and B from c
B=c*Ib;          %** using Ia and Ib.

function SKIP=icut(yj,Q)
%ICUT   Performs integer cuts to exclude previous bad deletions
%       yj=vector of measurements to be deleted
%       Q=matrix of cons. for previous bad deletions
%       maxdel=max number of deletions allowed for system
%
%       Returns SKIP=1, if yj or subsets of yj are have previously
%       caused non redundant measurements to be present.
SKIP=0;
n_c=1;                    %Qi column counter,presently under inspection
[Q_rn,Q_cn]=size(Q);              % *  While nothing has failed and the end
while SKIP==0 & n_c<=Q_cn         % * *of Qi is yet to be reached
  Qbar=0;
  Sum_yj_eQi=0;
  for n_r=1:Q_rn,
    if Q(n_r,n_c)~=0                              % *  Sum of all elements in yj
      Sum_yj_eQi=Sum_yj_eQi+yj(Q(n_r,n_c),1);     % * *with reference from Qi
    end,
  end,
  Qbar=nnz(Q(:,n_c));             %Number of nonzero entries in Qi
  if Sum_yj_eQi<=Qbar-1           %Integer cut condition
    SKIP=0;
  else
    SKIP=1;
  end,
  n_c=n_c+1;     %Increment Qi column counter
end

function [NREDN,OBJ,rxd]=reval(c,Ia,Ib,VAR,nm,A,B,nodes,nd,rxd,yj,ucalc,ucalc_rn)

%REVAL    Evaluates if the measurement structure defined by
%    A and B causes non redundant measurements to be
%    present.
%    Returns NREDN=1 and OBJ=0 if non redundant measurements
%    are detected
OBJ=1;           %*  Flags for NREDN measurements and good deletions,
NREDN=0;         %** expect no NREDN measurements
UNOBS=0;
M=[1:nm]*Ia;
[Q,R,E]=qr(B);
rR=rank(R);
Q2_cn=rR+1;
if Q2_cn <= nodes, Q2=Q(:,Q2_cn:nodes);, end %Split Q2 from Q
C=Q2'*A;
[C_rn,C_cn]=size(C);
Ck=1;
while Ck<=C_cn & OBJ==1          %Moving along the columns of C while present
  if VAR(M(Ck),3)~=1             %If Ck'th measurement may not be NREDN
    Col_Sum=0;                   %* Check column in C for a NREDN meas.
```



```
    for Cl=1:C_rn,               %*
      Col_Sum = Col_Sum + abs(C(Cl,Ck));%*
    end,                         %*
    if Col_Sum <= 0.01           %*  If a columns of zeros exist in C NREDN
      NREDN=1;                   %*  measurements exist, if rank R is less
      OBJ=0;                     %** than 'nd' unobservable variables exist
    end,
  end,
  Ck=Ck+1;
end
if rR<nd & NREDN==0
  U=[1:nm]*Ib*E;
  UNOBS=ucheck(ucalc,ucalc_rn,U,R,rR,nd);
  if UNOBS==1
    NREDN=1;
    OBJ=0;
  end,
end,
NREDNM=nnz(VAR(:,2));
if NREDN==0 & NREDNM>0
  [NREDN,OBJ]=frdn1(c,M,yj,VAR,nm,nd,nodes);
end
if NREDN==1
  ind_yj=find(yj);                          %*  Place the bad measurement
  [rxd_rn,rxd_cn]=size(rxd);                %*  structures onto the end
  if rxd(rxd_rn,1)~=0, rxd_rn=rxd_rn+1;, end,%*  of rxd
  for n=1:nd                                %*
    rxd(rxd_rn,n)=ind_yj(n,1);                   %**
  end,
end

function UNOBS=ucheck(ucalc,ucalc_rn,U,R,rR,nd)
%       checks that certain specified unmeasured variables
%       are observable. These variables are flaged in the
%       4th column of VAR
%       returns UNOBS=1 if measurement structure fails the
%       test.
UNOBS=0;
CHECK=0;
check_rn=1;
n=1;
while UNOBS==0 & n<=ucalc_rn
  nn=rR+1;
  while UNOBS==0 & nn<=nd
    if ucalc(n)==U(nn)
      UNOBS=1;
    end,
    nn=nn+1;
  end,
  if UNOBS==0
    for nn=1:rR
      if ucalc(n)==U(nn)
        CHECK=1;
        check(check_rn)=nn;
        check_rn=check_rn+1;
      end,
    end,
  end,
  n=n+1;
end,
```

```
if CHECK==1
  SUM=0;
  R11=R(1:rR,1:rR);
  R12=R(1:rR,rR+1:nd);
  iR11R12t=(inv(R11)*R12);
uindx=nd-rR;
if uindx>0
  for n=1:nd-rR
    SUM=SUM+abs(iR11R12t(:,n));,
  end,
else
  SUM=zeros(check_rn-1,1);
end
  n=1;
  while UNOBS==0 & n<=check_rn-1
    if SUM(check(n))>0.0001
      UNOBS=1;
    end
    n=n+1;
  end,
end,

function [NREDN,OBJ]=frdn1(c,M,yj,VAR,nm,nd,nodes)
%       checks whether the measurement structure represented by
%       Ia ensures a Degree of Redundancy of one in those measurements
%       specified in the second row of VAR.
%
%       Returns NREDN=1 , OBJ=0 if the measurement structure fails the test
NREDN=0;
OBJ=1;
frdn_x=find(VAR(:,2));
frdn_rn=nnz(frdn_x);
nmd=nm-nd;
C_cn=nmd-2;
k=1;
while k<=nmd & OBJ==1
    yj(M(k),1)=1;
    [A2,B2,Ia]=ab(c,nodes,yj,nm,nd);
    M2=[1:nm]*Ia;
    check=0;
    x=1;
    [Q,R,E]=qr(B2);
    rR=rank(R);
    Q2=Q(:,rR+1:nodes);
    C=Q2'*A2;
    for n=1:frdn_rn
      if frdn_x(n)~=M(k)
        check(x)=frdn_x(n);
        x=x+1;
      end
    end
    check_rn=nnz(check);
    x=1;
    while x<=check_rn & OBJ==1
      Ck=0;
      n=1;
      while Ck==0 & n<=C_cn
        if M2(n)==check(x)
          Ck=n;
        else
```



```
          Ck=0;
        end
        n=n+1;
      end
      if Ck~=0
        Col_Sum=0;              %** measurement structure is still considered goo
d
        for Cl=1:nodes-rR,
          Col_Sum = Col_Sum + abs(C(Cl,Ck));
        end,
        if Col_Sum <= 0.01      %*  If a columns of zeros exist in C NREDN
          NREDN=1;              %*  measurements exist, if rank R is less
          OBJ=0;                %** than 'nd' unobservable variables exist
        end,
      end,
      x=x+1;
    end
    yj(M(k),1)=0;
  k=k+1;
end

function [oxd]=ceval(yj,Ia,cost,nm_iter,nd,oxd)
%CEVAL  Evaluates the cost function for if the present measurement
%       structure defined by A and B is valid and stores it in matrix
%       oxd or stores measurements that have been deleted in matrix
%       rxd if they caused the presence of non redundant measurements.
ind_yj=find(yj);
COST_sum=ones(nm_iter,1);
COST=cost*Ia*COST_sum;          %Evaluate cost of current structure
[oxd_rn,oxd_cn]=size(oxd);
if oxd(oxd_rn,1)~=0, oxd_rn=oxd_rn+1;, end,
for n=1:nd                              %*  Place current measurement
  oxd(oxd_rn,n)=ind_yj(n,1);            %*  structure, with cost at
end                                     %*  the end of oxd
oxd(oxd_rn,nd+1)=COST;                  %**
```

```
% PROGRAM LISTING FOR THE LOCAL SEARCH ALGORITHM
clear
tic
%DATA ENTRY
load serth.dat
load serthVAR.dat
%DATA ENTRY - STOP
INIM=zeros(1,10);
ANSM=zeros(1,4);
for iter=1:10              %10 ITERATIONS - START
T_i=toc;
%VARIABLE INITIALIASATION
c=serth;
VAR=serthVAR;
[nodes,nm]=size(c);
VAR(:,4)=ones(nm,1);
NREDNM=[0];
%VARIABLE INITIALIASATION - STOP
%INITIAL SUB-OPTIMUM MEASUREMENT STRUCTURE
maxdel=nodes-1;
INDX=[1:3 5 6 10:14 16 18:20 22 23 26];
yj=indyj(INDX,c);
nd=nnz(yj);
INIM(iter,1:nd)=find(yj)';
YJ=yj;
%INITIAL SUB-OPTIMUM MEASUREMENT STRUCTURE - STOP

%INITIAL VARIABLE CLASSIFICATION
%
[A,B,Ia,Ib]=ab(c,nodes,yj,nm,nd);
[Q,R,E]=qr(B);
rR=rank(R)
R11=R(1:rR,1:rR);
R12=R(1:rR,rR+1:nd);
Q1=Q(:,1:rR);
Q2=Q(:,rR+1:nodes);
%% CHECK FOR NONREDUNDANT MEASUREMENTS
Aindx=[1:nm]*Ia;
C=Q2'*A;
[C_r,C_c]=size(C);
NREDNM_n=1;
for n=1:C_c
  Col_C_sum=0;
  for nn=1:C_r
    Col_C_sum=Col_C_sum+abs(C(nn,n));    %SUM THE nth COLUMN OF C
  end,
  if Col_C_sum < 1e-2
    NREDNM(NREDNM_n)=Aindx(n);           %REDN MEAS IF COLUMN OF ZEROS
    NREDNM_n=NREDNM_n+1;
  end,
end
%% CHECK FOR NONREDUNDANT MEASUREMENTS - STOP
%
%INITIAL VARIABLE CLASSIFICATION - STOP
%ADD MEASUREMENTS TO ELINIMATE NONREDUNDANCY
yji=yj;
if NREDNM(1)~=0
  nz_yj=find(yj);
  nnz_yj=nnz(yj);
  STOP=0;
```



```
  n=1;
  while STOP==0
    nn=1;
    while nn <= nnz_yj & STOP==0
      yj=yji;
      yj(nz_yj(nn))=0;
      nd=nnz(yj);
      [A,B,Ia,Ib]=ab(c,nodes,yj,nm,nd);
      Astarindx=[1:nm]*Ia;
      [Q,R,E]=qr(B);
      rR=rank(R);
      Q2=Q(:,rR+1:nodes);
      C=Q2'*A;
      [C_r,C_c]=size(C);
      Col_C_sum=0;
      for nnn=1:C_r
        Col_C_sum=Col_C_sum+abs(C(nnn,:));
      end,
      CHANGE=0;
      nnzNREDNM=nnz(NREDNM);
      for nnnn=1:nnzNREDNM
        for A_n=1:C_c
          if Astarindx(A_n)==NREDNM(nnnn) & VAR(NREDNM(nnnn),3)==0
            if Col_C_sum(A_n) > 1e-2
              yji(:,1)=yj;
              CHANGE=1;
            end,
          end,
        end,
      end,
      if CHANGE==1;
        NREDNM=[0];
        NREDNM_n=1;
        for n=1:C_c
          if Col_C_sum(n) < 1e-2
            NREDNM(NREDNM_n)=Astarindx(n);
            NREDNM_n=NREDNM_n+1;
          end,
        end
      end,
      nnzNREDNM=nnz(NREDNM);
      CONTIN=0;
      for nnnn=1:nnzNREDNM
        for A_n=1:C_c
          if Astarindx(A_n)==NREDNM(nnnn) & VAR(NREDNM(nnnn),3)==0
            if Col_C_sum(A_n) < 1e-2
              CONTIN=1;
            end,
          end,
        end,
      end,
      if CONTIN==1
        nn=nn+1;
      elseif CONTIN==0
        STOP=1;
      end,
    end,
  end,
end,
YJ(:,2)=yji;
```

```
%ADD MEASUREMENTS TO ELINIMATE NONREDUNDANCY - STOP
%ENFORCE A DEGREE OF REDUNDANCY OF ONE IN CERTAIN MEASUREMENTS
yji=YJ(:,2);
DOR1=find(VAR(:,2));
nDOR1=nnz(VAR(:,2));
nDOR1_p=0;
if nnz(VAR(:,2))>0
  nz_yj=find(yj);
  nnz_yj=nnz(yj);
  STOP=0;
  n=1;
  while STOP==0
    n2=1;
    while n2 <= nnz_yj & STOP==0
      DOR1ref=ones(1,nDOR1);
      yj=yji;
      yj(nz_yj(n2))=0;
      m_ref=ones(nm,1);
      m_ref=m_ref-yj;
      nnz_m_ref=nnz(m_ref);
      m_ref_indx=find(m_ref);
      for n3=1:nnz_m_ref
        yj(m_ref_indx(n3))=1;
        nd=nnz(yj);
        [A,B,Ia,Ib]=ab(c,nodes,yj,nm,nd);
        A_indx=[1:nm]*Ia;
        [Q,R,E]=qr(B);
        rR=rank(R);
        Q2=Q(:,rR+1:nodes);
        C=Q2'*A;
        [C_r,C_c]=size(C);
        for n4=1:C_c
          for n5=1:nDOR1
            if DOR1(n5)==A_indx(n4)
              Col_C_sum=0;
              for n6=1:C_r
                Col_C_sum=Col_C_sum+abs(C(n6,n4));
              end,
              if Col_C_sum<1e-2
                DOR1ref(n5)=0;
              end,
            end,
          end,
        end,
        yj(m_ref_indx(n3))=0;
      end,
      nDOR1ref=nnz(DOR1ref)
      if nDOR1ref<nDOR1 & nDOR1ref~=0
        yji=yj;
      elseif nDOR1ref==nDOR1
        yji=yj;
        STOP=1;
      end,
      yj(nz_yj(n2))=1;
      n2=n2+1;
    end,
  end,
end
YJ(:,3)=yji;
%ENFORCE A DEGREE OF REDUNDANCY OF ONE IN CERTAIN MEASUREMENTS - STOP
```



```
%REMOVE INITIAL MEASUREMENTS IN AN EFFORT TO FIND THE MINIMUM
ref=ones(nm,1);
ref=ref-YJ(:,1);
ref_indx=find(ref);
nnz_ref=nnz(ref)
Qi=[0];
ucalc=find(VAR(:,4));
ucalc_rn=nnz(ucalc);
GOOD=zeros(nm,1);
GOOD_n=1;
BAD=GOOD;
BAD_n=1;
deli=nnz(yj)+1;
nd=1
end_of_line=0;
oxd_p_rn=1;
while end_of_line == 0 & nd <= nnz_ref & deli<=maxdel
  rxd=zeros(1,nd);  %*  Initialise 'bad and good measurement structures'
  oxd=zeros(1,nd);  %** rxd and oxd respectively
  dxREDN=0;         %*  Flags marking 'bad and good measurement
  dxOBJ=0;          %** structures'.
  eof=0;
  if nd==1,
    for n=1:nnz_ref
      if VAR(ref_indx(n),1)~=1
        oxd_p(oxd_p_rn,1)=n;
        oxd_p_rn=oxd_p_rn+1;
      end,
    end,
  end,
  [nxd_f,dummy]=size(oxd_p);
  for nxd=1:nxd_f
    yj=yji;
    if nd==1
      test=[0];
      test=oxd_p(nxd);
      yj(ref_indx(oxd_p(nxd)),1)=1;
      md=nnz(yj);
      [A,B,Ia,Ib]=ab(c,nodes,yj,nm,md);
      [NREDN,OBJ,rxd]=reval(c,Ia,Ib,VAR,nm,A,B,nodes,md,rxd,yj,ucalc,ucalc_rn);
      if OBJ==1
        dxOBJ=1;
        [oxd]=ceval(yj,Ia,nd,oxd,test);
            GOOD(:,GOOD_n)=yj;
            GOOD_n=GOOD_n+1;
       end,
      if NREDN==1
        dxREDN=1;
            BAD(:,BAD_n)=yj;
            BAD_n=BAD_n+1;
      end,
      yj(ref_indx(oxd_p(nxd)),1)=0;
    else
      test=[0];
      for n=1:nd-1,
        test(n)=oxd_p(nxd,n);
        yj(ref_indx(oxd_p(nxd,n)),1)=1;
      end,
      test_r=nnz(test);
      for n=oxd_p(nxd,n)+1:nnz_ref
```

```
        if VAR(ref_indx(n),1)~=1;
          test(test_r+1)=n;
          yj(ref_indx(n),1)=1;
          if Qi(1)~=0, SKIP=icut(yj,Qi); end,
          if SKIP==0
            md=nnz(yj);
            [A,B,Ia,Ib]=ab(c,nodes,yj,nm,md);
            [NREDN,OBJ,rxd]=reval(c,Ia,Ib,VAR,nm,A,B,nodes,md,rxd,yj,ucalc,ucalc_
rn);
            if OBJ==1
              dxOBJ=1;
              [oxd]=ceval(yj,Ia,nd,oxd,test);
              GOOD(:,GOOD_n)=yj;
              GOOD_n=GOOD_n+1;
            end,
            if NREDN==1
              dxREDN=1;
              BAD(:,BAD_n)=yj;
              BAD_n=BAD_n+1;
            end,
          end,
          yj(ref_indx(n),1)=0;
        end,
      end,
    end,
  end,
  if dxREDN==1
    if Qi(1)==0
      Qi=rxd';
    else
      [Qi_rn,Qi_cn]=size(Qi);
      [rxd_rn,rxd_cn]=size(rxd);
      Qi_trans=zeros(md,1);
      Qi_trans(1:Qi_rn,1:Qi_cn)=Qi;
      Qi_trans(:,Qi_cn+1:Qi_cn+rxd_rn)=rxd';
      Qi=Qi_trans;
    end,
  end,
  if dxOBJ==1;
    save oxd.dat oxd -ascii
    save oxd_p.dat oxd_p -ascii
    oxd_p=oxd;
  else
    end_of_line=1;
  end,
  nd=nd+1
end
%REMOVE INITIAL MEASUREMENTS IN AN EFFORT TO FIND THE MINIMUM - STOP
T_f=toc;
[GOOD_n,GOOD_c]=size(GOOD);
fyj=GOOD(:,GOOD_c);
nnz_yj=nnz(fyj);
ANSM(iter,1:nnz_yj)=find(fyj)';
TIME(iter)=T_f-T_i;
end,            %10 ITERATIONS - STOP
save ANSM.dat ANSM -ascii
save INIM.dat INIM -ascii
save TIME.dat TIME -ascii
```



```
%SUBROUTINES

function yj=indyj(INDX,c)
% Forms independent set of initial unmeasured variables for
% the LSA
INDX_n=nnz(INDX);
[nodes,nm]=size(c);
yj=zeros(nm,1);
YJ=yj;
RB=1;
n=1;
while n<nodes
  randni=round(INDX_n*rand);
  if randni~=0
    yj(INDX(randni),1)=1;
    nd=nnz(yj);
    [A,B,Ia,Ib]=ab(c,nodes,yj,nm,nd);
    rB=rank(B);
    if rB==RB
      YJ=yj;
      RB=RB+1;
      n=n+1;
    end,
    yj=YJ;
  end,
end

function [A,B,Ia,Ib]=ab(c,nodes,yj,nm,nd)
%AB  Splits the c matrix into matrices A and B representing
%    measured and unmeasured variables respectively
%
%    c is the original incidence matrix
%    Yj is a vector representing while measurements are to be
%    deleted
%    nd is the number of measurements been deleted
%    nm is the number of measurements
Ia=zeros(nm,nm-nd);
Ib=zeros(nm,nd);
Ia_cn=1;
Ib_cn=1;
for n=1:nm
  if yj(n,1)==0          %*  If the variable is measured place an
    Ia(n,Ia_cn)=1;       %*  one in the corresponding place in Ia
    Ia_cn=Ia_cn+1;       %*  and increment the column counter in Ia
  else                   %*
    Ib(n,Ib_cn)=1;       %*  else do the same for Ib representing
    Ib_cn=Ib_cn+1;       %** the unmeasured variables
  end,
end,
A=c*Ia;          %*  Form A and B from c
B=c*Ib;          %** using Ia and Ib.

function SKIP=icut(yj,Q)
%ICUT   Performs integer cuts to exclude previous bad deletions
%       yj=vector of measurements to be deleted
%       Q=matrix of cons. for previous bad deletions
%       maxdel=max number of deletions allowed for system
%
%       Returns SKIP=1, if yj or subsets of yj are have previously
%       caused non redundant measurements to be present.
```

```
SKIP=0;
n_c=1;                    %Qi column counter,presently under inspection
[Q_rn,Q_cn]=size(Q);              % *  While nothing has failed and the end
while SKIP==0 & n_c<=Q_cn         % * *of Qi is yet to be reached
  Qbar=0;
  Sum_yj_eQi=0;
  for n_r=1:Q_rn,
    if Q(n_r,n_c)~=0                                % *  Sum of all elements in yj
      Sum_yj_eQi=Sum_yj_eQi+yj(Q(n_r,n_c),1);       % * *with reference from Qi
    end,
  end,
  Qbar=nnz(Q(:,n_c));             %Number of nonzero entries in Qi
  if Sum_yj_eQi<=Qbar-1           %Integer cut condition
    SKIP=0;
  else
    SKIP=1;
  end,
  n_c=n_c+1;    %Increment Qi column counter
end

function [NREDN,OBJ,rxd]=reval(c,Ia,Ib,VAR,nm,A,B,nodes,nd,rxd,yj,ucalc,ucalc_rn)

%REVAL    Evaluates if the measurement structure defined by
%    A and B causes non redundant measurements to be
%    present.
%    Returns NREDN=1 and OBJ=0 if non redundant measurements
%    are detected
% ,ucalc,ucalc_rn
OBJ=1;          %*  Flags for NREDN measurements and good deletions,
NREDN=0;        %** expect no NREDN measurements
UNOBS=0;
M=[1:nm]*Ia;
[Q,R,E]=qr(B);
rR=rank(R);
Q2_cn=rR+1;
if Q2_cn <= nodes, Q2=Q(:,Q2_cn:nodes);, end %Split Q2 from Q
C=Q2'*A;
[C_rn,C_cn]=size(C);
Ck=1;
while Ck<=C_cn & OBJ==1           %Moving along the columns of C while present
  if VAR(M(Ck),3)~=1              %If Ck'th measurement may not be NREDN
    Col_Sum=0;                    %* Check column in C for a NREDN meas.
    for Cl=1:C_rn,                %*
      Col_Sum = Col_Sum + abs(C(Cl,Ck));%*
    end,                          %*
    if Col_Sum <= 0.01            %*  If a columns of zeros exist in C NREDN
      NREDN=1;                    %*  measurements exist, if rank R is less
      OBJ=0;                      %** than 'nd' unobservable variables exist
    end,
  end,
  Ck=Ck+1;
end
if rR<nd & NREDN==0
  U=[1:nm]*Ib*E;
  UNOBS=ucheck(ucalc,ucalc_rn,U,R,rR,nd);
  if UNOBS==1
    NREDN=1;
    OBJ=0;
  end,
end,
```



```
NREDNM=nnz(VAR(:,2));
if NREDN==0 & NREDNM>0
  [NREDN,OBJ]=frdn1(c,M,yj,VAR,nm,nd,nodes);
end

if NREDN==1
  ind_yj=find(yj);
  [rxd_rn,rxd_cn]=size(rxd);
  if rxd(rxd_rn,1)~=0, rxd_rn=rxd_rn+1;, end,
  rxd(rxd_rn,1:nd)=ind_yj';
end

function UNOBS=ucheck(ucalc,ucalc_rn,U,R,rR,nd)
%       checks that certain specified unmeasured variables
%       are observable. These variables are flaged in the
%       4th column of VAR
%       returns UNOBS=1 if measurement structure fails the
%       test.
UNOBS=0;
CHECK=0;
check_rn=1;
n=1;
while UNOBS==0 & n<=ucalc_rn
  nn=rR+1;
  while UNOBS==0 & nn<=nd
    if ucalc(n)==U(nn)
      UNOBS=1;
    end,
    nn=nn+1;
  end,
  if UNOBS==0
    for nn=1:rR
      if ucalc(n)==U(nn)
        CHECK=1;
        check(check_rn)=nn;
        check_rn=check_rn+1;
      end,
    end,
  end,
  n=n+1;
end,
if CHECK==1
  SUM=0;
  R11=R(1:rR,1:rR);
  R12=R(1:rR,rR+1:nd);
  iR11R12t=(inv(R11)*R12);
uindx=nd-rR;
if uindx>0
  for n=1:nd-rR
    SUM=SUM+abs(iR11R12t(:,n));,
  end,
else
  SUM=zeros(check_rn-1,1);
end
  n=1;
  while UNOBS==0 & n<=check_rn-1
    if SUM(check(n))>0.0001
      UNOBS=1;
    end
    n=n+1;
```

```
  end,
end,

function [NREDN,OBJ]=frdn1(c,M,yj,VAR,nm,nd,nodes)
%       checks whether the measurement structure represented by
%       Ia ensures a Degree of Redundancy of one in those measurements
%       specified in the second row of VAR.
%
%       Returns NREDN=1 , OBJ=0 if the measurement structure fails the test

NREDN=0;
OBJ=1;
frdn_x=find(VAR(:,2));
frdn_rn=nnz(frdn_x);
nmd=nm-nd;
C_cn=nmd-2;
k=1;
while k<=nmd & OBJ==1
    yj(M(k),1)=1;
    [A2,B2,Ia]=ab(c,nodes,yj,nm,nd);
    M2=[1:nm]*Ia;
    check=0;
    x=1;
    [Q,R,E]=qr(B2);
    rR=rank(R);
    Q2=Q(:,rR+1:nodes);
    C=Q2'*A2;
    for n=1:frdn_rn
      if frdn_x(n)~=M(k)
        check(x)=frdn_x(n);
        x=x+1;
      end
    end
    check_rn=nnz(check);
    x=1;
    while x<=check_rn & OBJ==1
      Ck=0;
      n=1;
      while Ck==0 & n<=C_cn
        if M2(n)==check(x)
          Ck=n;
        else
          Ck=0;
        end
        n=n+1;
      end
      if Ck~=0
        Col_Sum=0;              %** measurement structure is still considered goo
d
        for Cl=1:nodes-rR,
          Col_Sum = Col_Sum + abs(C(Cl,Ck));
        end,
        if Col_Sum <= 0.01      %*  If a columns of zeros exist in C NREDN
          NREDN=1;              %*  measurements exist, if rank R is less
          OBJ=0;                %** than 'nd' unobservable variables exist
        end,
      end,
      x=x+1;
    end
    yj(M(k),1)=0;
```



```
  k=k+1;
end

function [oxd]=ceval(yj,Ia,nd,oxd,test)
%CEVAL  Evaluates the cost function for if the present measurement
%       structure defined by A and B is valid and stores it in matrix
%       oxd or stores measurements that have been deleted in matrix
%       rxd if they caused the presence of non redundant measurements.
[oxd_rn,oxd_cn]=size(oxd);
if oxd(oxd_rn,1)~=0, oxd_rn=oxd_rn+1;, end,
oxd(oxd_rn,:)=test;
```

Table E1 LSA starting points for the steam metering network.

LSA run	Indices of unmeasured variables									
1	1	2	5	11	13	14	16	18	20	23
2	1	5	6	10	11	16	18	19	22	26
3	3	5	6	10	11	12	19	22	23	26
4	1	2	6	11	12	13	19	20	22	23
5	3	5	6	10	13	16	18	19	22	26
6	1	3	5	12	13	14	19	20	22	23
7	2	3	5	6	11	16	18	20	22	23
8	3	5	6	11	12	13	14	20	23	26
9	1	2	3	5	11	12	19	20	22	23
10	2	5	6	12	13	16	18	20	22	23

Table E2 LSA starting points for the madron network.

LSA run	Indices of unmeasured variables									
1	1	4	7	15	16	18	19	20	21	23
2	1	3	5	7	9	20	21	22	23	24
3	2	4	15	17	18	19	20	21	22	24
4	1	3	9	15	16	19	20	21	22	23
5	1	4	5	7	9	19	20	21	22	23
6	2	5	8	9	15	17	20	21	23	24
7	1	4	9	15	17	18	19	20	21	22
8	2	5	6	8	15	17	18	19	20	21
9	3	6	8	15	16	18	19	20	21	23
10	3	5	6	8	9	15	20	21	23	24

APPENDIX F: Problem data and for the Madron network

MEASURED VARIABLES

All the variables are measured namely, 1 to 24.

BALANCE EQUATIONS

(the numbers represent the measurement tags)

NODE 1: $5 + 15 - 16 = 0$

NODE 2: $16 + 8 + 6 - 7 - 17 - 18 = 0$

NODE 3: $9 + 18 - 10 - 19 = 0$

NODE 4: $4 + 10 - 3 = 0$

NODE 5: $3 + 20 - 1 - 2 = 0$

NODE 6: $11 - 21 = 0$

NODE 7: $21 + 22 - 15 = 0$

NODE 8: $12 - 13 - 20 = 0$

NODE 9: $13 - 14 = 0$

NODE 10: $17 - 23 = 0$

NODE 11: $19 - 24 = 0$

Table F1 Matrix ***H*** representing the balance equations

	Measurements																							
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	1	-1	1	0	0	0	0	0	0	0	1	-1	-1	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	1	-1	0	0	0	0	0
4	0	0	-1	1	0	0	0	-1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	-1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-1	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	1	1	0	0
8	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	-1	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	-1	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-1

APPENDIX G: Results and m-files for the example Genetic Algorithm problem

```
%PROGRAM LISTING FOR SIMPLE GENEITIC ALGORITHM
clear
NIND=30;                    %# of individuals
MAXGEN=50;                  %maximum generations allowed
NVAR=2;                     %# of variables
PRECI=5;                    %Precision of variables
GGAP=0.9;                   %Generation gap
ANSM=[0 0 0]                %Matrix of solutions

%BUILD FIELD DESCRIPTOR
FieldD=[rep([PRECI],[1,NVAR]);...
  rep([-3;3],[1,NVAR]); rep([1;0;1;1],[1,NVAR])];
for n=1:1%00
  %INITIALISE POPULATION
  Chrom=crtbp(NIND,NVAR*PRECI);
  gen=0;
  %EVALUATE INITIAL POPULATION
  XY=bs2rv(Chrom,FieldD);
  XYi=XY;
  X=XY(:,1);
  Y=XY(:,2);
  ObjV=peaks(X,Y);
  xo=X;, yo=Y;, zo=ObjV;,
  %GENERATION LOOP
  while gen < MAXGEN,
    %ASSIGN FITNESS
    FitnV=ranking(ObjV);
    %SELECT INDIVIDUALS FOR BREEDING
    SelCH=select('sus',Chrom,FitnV, GGAP);
    %RECOMBINE INDIVIDUALS
    SelCH=recombin('xovsp',SelCH,0.7);
    %APPLY MUTATION
    SelCH=mut(SelCH,0.5);
    %EVALUATE OFFSRING
    XY=bs2rv(SelCH,FieldD);
    X=XY(:,1);
    Y=XY(:,2);
    ObjVSel=peaks(X,Y);
    %REINSERT OFFSPRING INTO POPULATION
    [Chrom ObjV]=reins(Chrom,SelCH,1,1,ObjV,ObjVSel);
    gen=gen+1;
  end
  ANS=bs2rv(Chrom,FieldD);
  ANS(:,3)=ObjV;
  [ANS_Y,ANS_I]=min(ANS(:,3));
  ANSM(n,:)=ANS(ANS_I,:);
end

%Definittion of the PEAKS function
%
%       z =  3*(1-x).^2.*exp(-(x.^2) - (y+1).^2)
%            - 10*(x/5 - x.^3 - y.^5).*exp(-x.^2-y.^2)
%            - 1/3*exp(-(x+1).^2 - y.^2)
```

Table G1 GA solutions to the 'Peaks' problem

Population = 30		74 %
x	y	Obj
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.4839	-1.6452	-5.9434
0.2903	-1.6452	-6.5048
0.2903	-1.8387	-5.8970
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.4516	-6.1156
0.0968	-1.6452	-6.3911
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.4516	-6.1156
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.2903	-1.4516	-6.1156
0.2903	-1.4516	-6.1156
0.2903	-1.4516	-6.1156
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.4516	-6.1156
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048
0.0968	-1.6452	-6.3911
0.2903	-1.6452	-6.5048
0.2903	-1.6452	-6.5048

APPENDIX H: Results and m-files for the application of Genetic Algorithms to gross error detection.

```
%ZGEPN.M
%ADDS RANDOM NOISE AND SIMULATES GROSS ERRORS TO THE
%TRUE VALUES OF THE MEASUREMENTS

clear
load ytrue.dat
load S.dat

nm=28;
tests=100;
NVAR=2;
stop=0

%CREATE 100 RANDOM POSITIONS OF THE GROSS ERRORS - start
while stop==0
  [GEat,dummy]=rge(NVAR,nm,tests)
  if dummy==100
    stop=1;
  end,
end,
save GEat.dat GEat -ascii
%CREATE 100 RANDOM POSITIONS OF THE GROSS ERRORS - stop

% CREATE 100 RANDOM POSITIONS OF THE GROSS ERRORS - start
for n=1:100
  y=rnoise(ytrue,S);
  zgepn(:,n)=rvge(GEat,n,y,NVAR,S);
end
save zgepn.dat zgepn -ascii
% CREATE 100 RANDOM POSITIONS OF THE GROSS ERRORS - stop

%SUBROUTINES

function [GEat,dummy]=rge(NVAR,nm,tests)
% Generates 100 random placements for two Gross Errors, returns
% matrix GEat(x,y) of size tests x NVAR
%
% if returns dummy=100 implies correct execution.

GEat=zeros(tests,2);
for n=1:tests
  ne=1;
  STOP=0;
  while STOP==0
    x=rand*nm;
    if x < 0.5
     x=1;
    else
      x=round(x);
    end
    if ne==1, GEat(n,ne)=x; ne=ne+1;, end
    if ne==2
      if x~=GEat(n,1),
        STOP=1;
      end
    end,
    GEat(n,ne)=x;
  end
end
GEcolsub=GEat(:,1)-GEat(:,2);
```



```
dummy=nnz(GEcolsub);

function y=rnoise(ytrue,s)
% ADDS RANDOM NOISE TO THE MEASUREMENTS < 0,5
% OF THE TRUE MEASUREMENT VALUE
%load ytrue.dat
%load s.dat
nm=28;
y=zeros(nm,1);
for n=1:nm
  e_n=rand;
  y(n)=ytrue(n)+0.05*e_n*ytrue(n);
end

function z=rvge(GEat,n,y,NVAR,s)
%       computes value of the gross error as a random
%       value between 5 and 15 % of the corresponding
%       measurements standard deviation
%
VUB=1;
VLB=0.1;
z=y;
for nn=1:NVAR
  z(GEat(n,nn))=z(GEat(n,nn))+(VLB+rand*(VUB-VLB))*z(GEat(n,nn));
end
```

```
%GENETIC ALGORITHMS AND GROSS ERROR DETECTION
clear
tic
%DATA ENTRY - start
load A.dat
load S.dat
load zgepn.dat
load GEat.dat
%DATA ENTRY - stop
%VARIABLE INITIALISATION - start
nm=28;
NIND=30;                %# of individuals
MAXGEN=6;               %maximum generations allowed
NVAR=2;                 %# of variables
PRECI=5;                %Precision of variables
GGAP=0.9;               %Generation gap
ANSM=[0 0 0];           %Matrix of solutions
%VARIABLE INITIALISATION - stop
iV=inv(A*S*A');                 %inverse of Co-variance of residuals
%BUILD FIELD DESCRIPTOR
FieldD=[rep([PRECI],[1,NVAR]);...
  rep([0;28],[1,NVAR]); rep([1;0;1;1],[1,NVAR])];
for n=1:100
  ANSWER=0;
  z=zgepn(:,n);
  %INITIALISE POPULATION
  Chrom=crtbp(NIND,NVAR*PRECI);
  gen=0;
  %EVALUATE INITIAL POPULATION
  XY=bs2rv(Chrom,FieldD);
  XY=xyint(XY,nm);              %Round real values of XY to integers
  XYo=XY;
%  ObjV=cro(XY,A,z,iV);         %Recursive identification of GE
  ObjV=globalTS(XY,A,z,S,nm);   %Global statistic test ObjF
  %GENERATION LOOP
  while gen < MAXGEN,
    %ASSIGN FITNESS
    FitnV=ranking(ObjV);
    %SELECT INDIVIDUALS FOR BREEDING
    SelCH=select('sus',Chrom,FitnV, GGAP);
    %RECOMBINE INDIVIDUALS
    SelCH=recombin('xovsp',SelCH,0.7);
    %APPLY MUTATION
    SelCH=mut(SelCH,0.5);
    %EVALUATE OFFSPRING
    XY=bs2rv(SelCH,FieldD);
    XY=xyint(XY,nm);                    %Round real values of XY to integers
%    ObjVSel=cro(XY,A,z,iV);            %Recursive identification of GE
    ObjVSel=globalTS(XY,A,z,S,nm);      %Global statistic test ObjF
    %REINSERT OFFSPRING INTO POPULATION
    [Chrom ObjV]=reins(Chrom,SelCH,1,1,ObjV,ObjVSel);
    gen=gen+1;
  end
  ANS=bs2rv(Chrom,FieldD);
  ANS=xyint(ANS,nm);
  ANS(:,NVAR+1)=ObjV;
  FOUND=0;
  nn=1;
  while nn<=NIND
    if ANS(nn,1)-GEat(n,1)==0
```



```
      if ANS(nn,2)-GEat(n,2)==0
        FOUND=1;
        ni=nn;
      end,
    elseif ANS(nn,2)-GEat(n,1)==0
      if ANS(nn,1)-GEat(n,2)==0
        FOUND=1;
        ni=nn;
      end,
    end,
    nn=nn+1;
  end,
  if FOUND==1
    ANSM(n,:)=ANS(ni,:);
  end,
end
toc

%SUBROUTINES

function ObjV=cro(XY,A,z,iV)
% OBJECTIVE FUNCTION USING CROWES RECURSIVE GROSS ERROR
% DETECTION METHOD for removal of all measurements
% represented in XY
[XY_rn,XY_cn]=size(XY);
r=A*z;
J=r'*iV*r;
for n=1:XY_rn
  if XY(n,1)==XY(n,2)
    ObjV(n,1)=10000;
  else
    B=A(:,XY(n,1));
    B(:,2)=A(:,XY(n,2));
    G=B'*iV*B;
    dJ=-(r'*iV*B)*inv(G)*(B'*iV*r);
    ObjV(n,1)=J+dJ;
  end,
end,

function ObjV=globalTS(XY,c,z,S,nm)
%       THE GLOBAL TEST STATICTIC METHOD FOR GROSS ERROR DETECTION
%       for removal of all measurements represented in XY
[XY_rn,nd]=size(XY);
nodes=11;
Tinfcount=0;
%EVALUATE THE EFFECT OF DELETING TWO MEASUREMENTS
for n=1:XY_rn
  if XY(n,1)==XY(n,2)
    ObjV(n,1)=10000;
  else
    yj=zeros(nm,1);
    for nn=1:nd
      yj(XY(n,1))=1;
    end,
    %CALCULATION - start
    [A,B,Ia]=ab(c,nodes,yj,nm,nd);
    y=z'*Ia;
    y=y';
    s=Ia'*S*Ia;
    [Q,R,E]=qr(B);
```

```
    rR=rank(R);
    Q2=Q(:,rR:nodes);
    C=Q2'*A;
    w=C*y;
    t=w'*inv(C*s*C')*w;
    % CALCULATION - stop
    ObjV(n,1)=t;
  end
end

function XY=xyint(XY,nm)
%       Convert the real values of BS2RV(Chrom,FieldD) to integer
%       values to represent individual measurements in a measurement
%       structure.
%
%       Notes,  Values    <     0.5     get forced to 1
%               Values    >=    nm+0.5  get forced to nm
[XY_rn,XY_cn]=size(XY);
LVB=0.5;
UVB=nm+0.5;
for n=1:XY_rn
  for nn=1:XY_cn
    if XY(n,nn)<LVB
      XY(n,nn)=1;
    elseif XY(n,nn)>=UVB
      XY(n,nn)=nm;
    else
      XY(n,nn)=round(XY(n,nn));
    end,
  end,
end
```

```
%SEARCH THE ENTIRE SPACE FOR THE SIMULATED GROSS ERRORS
clear
NVAR=2;
%DATA ENTRY - start
load s.dat
load ytrue.dat
load zgepn.dat
load A.dat
%DATA ENTRY - stop
n=1
GEat=[1 17]              %POSITION OF GROSS ERRORS
y=rnoise(ytrue,s);       %GENERATE MEASUREMENT NOISE
z=rvge(GEat,n,y,NVAR,s);        %GENERATE RANDOM SIZED GROSS ERROR
%CHOICE OF OBJECTIVE FUNCTION - REMOVE COMMENT '%'TO INITIALISE
%ocro            %CROWES OBJECTIVE FUNCTION
%glob            %GLOBAL TEST OBJECTIVE FUNCTION
%rndsrch         %RANDOM SEARCH


%SUBROUTINES

function y=rnoise(ytrue,s)
%       generates random noise < 5% the measurement value
nm=28;
y=zeros(nm,1);
for n=1:nm
  e_n=rand;
  y(n)=ytrue(n)+0.05*e_n*ytrue(n);
end

function z=rvge(GEat,n,y,NVAR,s)
%       computes value of the gross error as a random
%       value between 5 and 15 % of the corresponding
%       measurements standard deviation
%
VUB=1;
VLB=0.1;
z=y;
for nn=1:NVAR
  z(GEat(n,nn))=z(GEat(n,nn))+(VLB+rand*(VUB-VLB))*z(GEat(n,nn));
end

%OCRO.M
iV=inv(A*s*A');
r=A*z;
J=r'*iV*r;
nm=28;
count=1;
for i=1:nm-1
  for j=i+1:nm
    B=A(:,i);
    B(:,2)=A(:,j);
    G=B'*iV*B;
    dJ=-(r'*iV*B)*inv(G)*(B'*iV*r);
    res(count,1)=i;
    res(count,2)=j;
    res(count,3)=J+dJ;
    count=count+1;
  end,
end,
```



```
%GLOB.M
nm=28;                          %number of measurements
nd=2;
yj=zeros(nm,1);
c=A;
S=s;
nodes=11;
Tinfcount=0;
T=zeros(1,3);
%EVALUATE THE EFFECT OF DELETING TWO MEASUREMENTS
for i=1:27
  for j=i+1:28,
    yj(i,1)=1;
    yj(j,1)=1;
    %Calculation Start - measurement bias only -
    [A,B,Ia]=ab(c,nodes,yj,nm,nd);
    y=z'*Ia;
    y=y';
    s=Ia'*S*Ia;
    [Q,R,E]=qr(B);
    Q2=Q(:,3:11);
    C=Q2'*A;
    w=C*y;
    t=w'*inv(C*s*C')*w;
    % Calculation Stop
    Tinfcount=Tinfcount+1;
    T(Tinfcount,1)=i;
    T(Tinfcount,2)=j;
    T(Tinfcount,3)=t;
    yj(i,1)=0;
    yj(j,1)=0;
  end,
end

%RNDSRCH.M
nm=28;
NIND=180;               %# of individuals
MAXGEN=1;               %maximum generations allowed
NVAR=2;                 %# of variables
PRECI=5;                %Precision of variables
GGAP=0.9;               %Generation gap
ANSM=[0 0 0];           %Matrix of solutions
res=ANSM;
iV=inv(A*s*A');         %inverse of Co-variance of residuals
rnd_count=1;
%BUILD FIELD DESCRIPTOR
FieldD=[rep([PRECI],[1,NVAR]);...
  rep([0;28],[1,NVAR]); rep([1;0;1;1],[1,NVAR])];
for n=1:100
gen=1;
z=zgepn(:,n);
  while gen <= MAXGEN,
  %IINITIALISE POPULATION
  Chrom=crtbp(NIND,NVAR*PRECI);
  %EVALUATE INITIAL POPULATION
  XY=bs2rv(Chrom,FieldD);
  XY=xyint(XY,nm);
  ObjV=cro(XY,A,z,iV);
  [ANS_Y,ANS_I]=min(ObjV);
```

```
  ANSM(gen,1:2)=XY(ANS_I,:);
  ANSM(gen,3)=ObjV(ANS_I);
  gen=gen+1;
  end
[ANSM_Y,ANSM_I]=sort(ANSM(:,3));
res(n,:)=ANSM(ANSM_I(1),:);
end
res(:,4:5)=GEat;
toc

function XY=xyint(XY,nm)
%       Convert the real values of BS2RV(Chrom,FieldD) to integer
%       values to represent individual measurements in a measurement
%       structure.
%
%       Notes,  Values    <     0.5     get forced to 1
%               Values    >=    nm+0.5  get forced to nm
[XY_rn,XY_cn]=size(XY);
LVB=0.5;
UVB=nm+0.5;
for n=1:XY_rn
  for nn=1:XY_cn
    if XY(n,nn)<LVB
      XY(n,nn)=1;
    elseif XY(n,nn)>=UVB
      XY(n,nn)=nm;
    else
      XY(n,nn)=round(XY(n,nn));
    end,
  end,
end
```

Table H1 Results for GA Approach to Gross Error Detection

Position of Gross Errors		SGA	20 % Correct		SGA	3 % Correct		Random Search 22 % Correct		
x	y	Crowe Obj	1st GE pos	2nd GE pos	Global Obj	1st GE pos	2nd GE pos	Crowe Obj	1st GE pos	2nd GE pos
1	17	2.1998	6	17	14.00	23	14	1.6265	17	23
5	21	2.5206	5	21	2.50	5	21	2.5206	5	21
16	27	3.8051	11	27	696.30	19	8	3.4071	16	27
26	9	1.7571	26	1	27.30	7	4	1.4171	26	12
3	2	1.2978	23	3	1036.60	22	4	1.2978	23	3
25	27	86.3746	25	19	833.00	22	13	86.3746	19	25
11	17	10.1343	17	12	64.90	23	6	15.8883	3	11
20	15	10.7157	7	15	558.90	22	3	5.6212	15	20
9	8	2.8156	8	3	942.00	7	9	2.0045	22	8
3	22	3.1263	22	3	71.70	9	27	5.3161	15	3
23	3	2.6055	3	7	53.20	7	24	2.7016	10	3
17	1	2.7974	3	4	54.40	23	8	2.7974	3	17
13	12	290.8029	12	7	1657.40	1	16	290.8029	12	7
5	19	0.3707	5	4	756.10	11	16	0.3589	13	5
23	13	2.5295	13	10	1138.80	2	26	2.9931	23	13
7	4	1.9050	13	4	501.00	1	23	1.9050	4	13
9	20	1.5304	25	20	3.20	7	1	3.5977	12	20
27	3	97.8681	13	2	222.70	19	15	41.6453	3	19
3	14	2.8571	3	14	866.20	18	20	2.8571	14	3
22	26	3.8148	16	26	554.40	20	24	4.9324	26	24
17	22	2.3875	17	23	34.60	9	26	2.3875	17	23
25	17	1.6826	25	17	122.80	22	7	1.6826	16	17
13	21	602.3875	14	2	1705.40	19	26	602.9577	2	21
5	8	38.1283	8	1	1412.70	7	14	38.1283	1	8
1	2	1.2014	23	13	2.00	2	5	1.0868	23	3
23	28	2.8420	28	18	501.40	20	25	1.8977	28	23
12	24	1.1346	24	12	357.40	18	22	1.1346	12	24
26	19	4.3176	26	20	186.50	20	3	4.6270	6	26
20	14	6.0952	26	14	222.80	17	9	6.0952	26	14
2	9	1.8022	25	4	3.90	9	23	1.8022	4	25
13	14	4.3361	13	14	1446.70	18	23	4.3361	13	14
10	7	2.4453	10	27	46.60	1	22	2.1876	10	13
13	1	1.5171	13	21	330.70	2	14	1.6151	11	13
8	7	3.7430	8	15	986.90	7	22	3.8814	8	6
20	23	2.9974	20	15	8.80	7	22	2.4772	23	20
24	21	353.5519	21	14	1071.50	17	16	91.8124	21	15
22	20	1.5293	22	28	6.20	7	12	2.1284	10	22
17	18	3.0567	14	24	66.60	23	5	3.0567	14	24
20	26	9.9509	20	9	22.90	20	9	3.4594	26	19
18	22	1.7156	16	24	6.40	23	24	1.7156	24	16
22	28	5.5288	28	12	128.50	20	1	4.4772	15	28
26	15	8.3490	15	23	593.60	22	1	11.3582	15	5
22	7	1.7197	20	5	2.70	1	12	1.0436	1	20
6	9	1.8149	5	6	297.30	9	15	1.4504	17	6
1	25	0.3592	25	28	420.40	22	19	1.6587	25	27
19	14	4.0648	14	23	198.80	18	27	4.0648	23	14
9	20	2.6483	4	20	6.40	7	6	3.6475	18	20
16	10	16.2073	10	25	846.20	11	1	16.2073	10	25
16	26	30.7015	26	25	459.10	20	2	2.3252	16	26
20	13	4.6707	13	7	116.70	2	16	2.7722	20	13
26	2	3.9110	26	11	753.20	20	7	2.9662	26	21
11	17	11.1476	11	14	48.80	7	11	10.6232	11	3
27	5	1.1184	5	27	829.30	11	19	1.1184	27	5
14	23	1.8797	15	18	119.00	18	12	2.9520	14	23
26	27	24.1618	19	28	117.10	28	16	53.1459	28	25
9	5	3.1408	5	16	108.30	9	5	2.8537	14	5
12	26	6.7271	12	26	1438.80	1	9	135.1591	19	12
16	13	2.1583	22	13	136.70	2	24	2.4949	13	25
11	26	37.2330	26	12	586.40	20	15	52.6616	17	26
22	1	1.2288	22	19	9.20	1	22	1.0276	22	20
2	20	2.2298	14	20	10.50	7	1	2.0652	20	27
27	26	1.2610	27	26	195.60	28	12	1.2610	27	26
23	21	1.5991	21	23	42.00	20	21	2.1760	18	21
27	19	3.7227	19	27	70.40	19	14	3.7227	19	21
1	15	1.2497	13	15	375.00	22	24	1.1837	15	17
10	12	3.3383	10	12	1366.70	16	14	256.4914	1	16
1	2	1.7359	8	3	3.60	1	21	2.0295	8	10
6	8	3.5185	6	8	1194.60	7	22	182.7444	7	20
13	1	3.5549	6	13	1210.10	2	25	3.1892	13	25
25	4	48.0805	25	16	1027.00	22	6	65.7599	11	25
24	2	1.1927	24	6	180.50	18	1	1.6651	17	24
11	18	3.4973	7	11	42.10	7	19	3.4627	11	8
14	13	3.7817	13	14	1137.60	18	17	3.7817	13	14
10	1	2.3648	10	14	688.10	11	4	3.4466	10	22
23	17	2.6503	17	20	11.60	7	12	2.7967	14	17
8	7	4.6086	8	5	63.20	7	24	4.6086	5	8
18	17	3.0384	10	17	23.50	23	20	3.3572	15	17
6	23	3.7301	6	23	201.80	9	8	3.7301	6	23
13	2	0.6465	10	13	659.10	2	11	1.6922	1	13
7	23	1.6705	23	14	3.70	7	10	1.1825	20	15
6	27	18.2589	19	6	145.30	19	5	30.1243	23	6
5	1	1.7940	6	5	623.70	11	23	1.7940	5	6
19	18	2.9603	1	18	19.30	23	2	4.7207	18	27
14	11	2.4510	11	14	102.40	11	5	19.3878	20	11
7	12	2.3239	3	12	173.00	1	22	2.5533	12	5
7	22	2.1097	7	22	10.40	1	8	1.8657	4	22
24	10	13.1256	12	10	909.30	11	3	3.3517	24	10
13	1	2.4387	13	21	78.10	2	14	2.6538	19	13
5	4	98.4446	1	5	736.40	11	28	2.5427	4	5
15	18	3.4231	15	18	161.30	22	1	3.4231	15	18
4	14	61.8977	4	21	558.00	1	28	1.7684	4	14
12	16	3.5290	16	11	98.10	1	18	3.5290	16	11
10	2	3.2279	25	10	760.10	11	1	3.2279	25	10
16	21	3.7324	16	14	102.00	11	23	16.1920	16	6
10	3	4.3968	10	3	82.40	1	26	4.3968	10	3
23	9	1.2384	28	23	4.20	7	9	1.2384	23	28
24	27	3.0922	24	23	517.40	19	22	3.0922	23	27
20	18	9.9186	23	19	62.20	23	12	3.8623	18	28
5	3	0.7747	3	5	414.90	9	24	0.7747	5	3
10	1	1.6842	5	10	308.10	11	10	1.3775	19	10

APPENDIX I: Data and m-files for the industrial case study

```
%       General Data Reconciliation of the CALTEX crude preheat train
%       Variances aribitarily assigned, Flow 5% of value, Temp .5% of value
%
%       Unmeasured variables set @ 100 units
tic
clear
nm=35;                  %Number of measurements
nu=28;
nv=nm+nu;               %Total number of variables
zm_indx=[1 3 11 43 17 21 27 53 54 35 36 2 10 12 8 26 34 14 16 18 22 20 24 28 30 4
0 46 50 56 59 57 61 62 38 32];
ANSM=zeros(nv,1);
INIM=ANSM;
zstar=0;

load sdata.dat
s=zeros(nv,nv);
load fcdata.dat
[fcdat_rn,fcdat_cn]=size(fcdata);

for dataset=1:fcdat_rn
  dataset
    z=ones(size(1:nv))'*100;
    for n=1:fcdat_cn
      z(zm_indx(n))=fcdata(dataset,n);
      s(zm_indx(n),zm_indx(n))=sdata(n);
    end,
    stop=0;
  so=s;

% compute yj - start
  yj=zeros(nv,1);
  yj(4)=1;
  yj(5)=1;
  yj(6)=1;
  yj(7)=1;
  yj(9)=1;
% yj(8)=1;
% z(8)=100;
  yj(13)=1;
  yj(15)=1;
  yj(19)=1;
  yj(23)=1;
  yj(25)=1;
  yj(29)=1;
  yj(31)=1;
% yj(32)=1;
% z(32)=100;
  yj(33)=1;
  yj(37)=1;
  yj(39)=1;
  yj(41)=1;
  yj(42)=1;
  yj(44)=1;
  yj(45)=1;
  yj(47)=1;
  yj(48)=1;
  yj(49)=1;
  yj(51)=1;
  yj(52)=1;
```



```
  yj(55)=1;
  yj(58)=1;
  yj(60)=1;
  yj(63)=1;
% compute yj - stop

  INIM(:,dataset)=z;
  Ia=zeros(nv,nm);
  Ia_rn=1;
  Ib=zeros(nv,nu);
  Ib_rn=1;
  for n=1:nv
    if yj(n)==0;
      Ia(n,Ia_rn)=1;
      Ia_rn=Ia_rn+1;
    elseif yj(n)==1;
      Ib(n,Ib_rn)=1;
      Ib_rn=Ib_rn+1;
    end,
  end
  Iht=Ia';
  [Iht_rn,Iht_cn]=size(Iht);
  Iht(Iht_rn+1:Iht_rn+nu,:)=Ib';
  y=(z'*Ia)';
  y_crit=y;
  s=Ia'*s*Ia;
  [Hz,AB]=network(z);

  while stop==0
    A=AB*Ia;
    B=AB*Ib;
    x=(z'*Ia)';
    u=(z'*Ib)';
    c=A*x+B*u-Hz;
    [Q,R,E]=qr(B);
    rR=rank(R);
    rQ=rank(Q);
    [Q_rn,Q_cn]=size(Q);
    [R_rn,R_cn]=size(R);
    Q1=Q(:,1:rR);
    Q2=Q(:,rR+1:Q_cn);
    R11=R(1:rR,1:rR);
    R12=R(1:rR,rR+1:R_cn);
    C=Q2'*A;
    d=Q2'*c;
    if rR<rQ                    %Normal case where Q2 exists
      i=inv(C*s*C');
      res=s*C'*i*(C*y-d);
      x=y-res;
    elseif rR==rQ               %If no Q2 exists, the measurements
      Q=Q1;                     %are not reconciles, but the un-
    end,                        %measured variables calc,using Q1
    u=inv(R11)*(Q1'*c-Q1'*A*x);
    u=u'*E';                    %rearange u in ori form as in B
    zstar(1:nm,1)=x;
    zstar(nm+1:nv)=u;           %zstar=[x|u]'
    zstar=zstar'*Iht;           %rearange sequence of vars. to same as z
    zstar=zstar';               %correct dimension
    [Hz,AB]=network(zstar);
    e2=abs(Hz);
```

```
    Me2=max(e2);
    e=abs(abs(x)-abs(y_crit));  %Stopping criterion based on the
    [Me,ii]=max(e)  ;           %max value in the contrains vector, Hz
    if Me<1e-5 stop=1;, end,
    z=zstar;
    y_crit=x;
  end,
  ANSM(:,dataset)=z;
end,
toc

%SUBROUTINES

function [Hz,AB]=network(z)
%       CALTEX RECONCILIATION PROBLEM
%       network forms the matrix for the linearised material and
%       energy balances based on the original or newly updated
%       values for the measurements in vector z
%
%       Requires the existance of the following for the
%       calculation of the Enthalpy and derivative of. curves.
%       Hcrude, Htpa, Hmpa, Hltdsl, Hvgo, Hvbo, Hwater

%Splitter, of Crude feed stream
Hz(1,1)=z(3)+z(5)-z(1);
Hz(2,1)=z(4)-z(2);
Hz(3,1)=z(6)-z(2);

AB(1,1)=-1;
AB(1,3)=1;
AB(1,5)=1;
AB(2,2)=-1;
AB(2,4)=1;
AB(3,2)=-1;
AB(3,6)=1;


%2E201
Hz(4,1)=z(11)*Htpa(z(12))-z(9)*Htpa(z(10)) + z(7)*Hcrude(z(8))-z(3)*Hcrude(z(4));

Hz(5,1)=z(11)-z(9);
Hz(6,1)=z(7)-z(3);

AB(4,11)=Htpa(z(12));
AB(4,9)=-Htpa(z(10));
AB(4,7)=Hcrude(z(8));
AB(4,3)=-Hcrude(z(4));
AB(4,12)=z(11)*dHtpa(z(12));
AB(4,10)=-z(9)*dHtpa(z(10));
AB(4,8)=z(7)*dHcrude(z(8));
AB(4,4)=-z(3)*dHcrude(z(4));
AB(5,11)=1;
AB(5,9)=-1;
AB(6,7)=1;
AB(6,3)=-1;

%2E301
Hz(7,1)=z(15)*Hvbo(z(16))-z(13)*Hvbo(z(14)) + z(17)*Hcrude(z(18))-z(5)*Hcrude(z(6
));
Hz(8,1)=z(15)-z(13);
```



```
Hz(9,1)=z(17)-z(5);

AB(7,15)=Hvbo(z(16));
AB(7,13)=-Hvbo(z(14));
AB(7,17)=Hcrude(z(18));
AB(7,5)=-Hcrude(z(6));
AB(7,16)=z(15)*dHvbo(z(16));
AB(7,14)=-z(13)*dHvbo(z(14));
AB(7,18)=z(17)*dHcrude(z(18));
AB(7,6)=-z(5)*dHcrude(z(6));
AB(8,15)=1;
AB(8,13)=-1;
AB(9,17)=1;
AB(9,5)=-1;

%2E2
Hz(10,1)=z(21)*Hltdsl(z(22))-z(19)*Hltdsl(z(20)) + z(23)*Hcrude(z(24))-z(17)*Hcru
de(z(18));
Hz(11,1)=z(21)-z(19);
Hz(12,1)=z(23)-z(17);

AB(10,19)=-Hltdsl(z(20));
AB(10,21)=Hltdsl(z(22));
AB(10,23)=Hcrude(z(24));
AB(10,17)=-AB(7,17);
AB(10,20)=-z(19)*dHltdsl(z(20));
AB(10,22)=z(21)*dHltdsl(z(22));
AB(10,24)=z(23)*dHcrude(z(24));
AB(10,18)=-AB(7,18);
AB(11,19)=-1;
AB(11,21)=1;
AB(12,23)=1;
AB(12,17)=-1;

%2E6
Hz(13,1)=z(27)*Hmpa(z(28))-z(25)*Hmpa(z(26)) + z(29)*Hcrude(z(30))-z(23)*Hcrude(z
(24));
Hz(14,1)=z(27)-z(25);
Hz(15,1)=z(29)-z(23);

AB(13,27)=Hmpa(z(28));
AB(13,25)=-Hmpa(z(26));
AB(13,29)=Hcrude(z(30));
AB(13,23)=-AB(10,23);
AB(13,28)=z(27)*dHmpa(z(28));
AB(13,26)=-z(25)*dHmpa(z(26));
AB(13,30)=z(29)*dHcrude(z(30));
AB(13,24)=-AB(10,24);
AB(14,27)=1;
AB(14,25)=-1;
AB(15,29)=1;
AB(15,23)=-1;

%2E202
Hz(16,1)=z(25)*Hmpa(z(26))-z(31)*Hmpa(z(32)) + z(33)*Hcrude(z(34))-z(7)*Hcrude(z(
8));
Hz(17,1)=z(25)-z(31);
Hz(18,1)=z(33)-z(7);

AB(16,25)=-AB(13,25);
```

```
AB(16,31)=-Hmpa(z(32));
AB(16,33)=Hcrude(z(34));
AB(16,7)=-AB(4,7);
AB(16,26)=-AB(13,26);
AB(16,32)=-z(31)*dHmpa(z(32));
AB(16,34)=z(33)*dHcrude(z(34));
AB(16,8)=-AB(4,8);
AB(17,25)=1;
AB(17,31)=-1;
AB(18,33)=1;
AB(18,7)=-1;

%VGO Mixer
Hz(19,1)=z(37)-z(35)-z(36);

AB(19,37)=1;
AB(19,35)=-1;
AB(19,36)=-1;

%2E3
Hz(20,1)=z(41)*Hvgo(z(42))-z(37)*Hvgo(z(38)) + z(39)*Hcrude(z(40))-z(29)*Hcrude(z
(30));
Hz(21,1)=z(41)-z(37);
Hz(22,1)=z(39)-z(29);

AB(20,41)=Hvgo(z(42));
AB(20,37)=-Hvgo(z(38));
AB(20,39)=Hcrude(z(40));
AB(20,29)=-AB(13,29);
AB(20,42)=z(41)*dHvgo(z(42));
AB(20,38)=-z(37)*dHvgo(z(38));
AB(20,40)=z(39)*dHcrude(z(40));
AB(20,30)=-AB(13,30);
AB(21,41)=1;
AB(21,37)=-1;
AB(22,39)=1;
AB(22,29)=-1;

%JOIN OF TWO CRUDE SPLITS
Hz(23,1)=z(44)*Hcrude(z(48))-z(33)*Hcrude(z(34))-z(39)*Hcrude(z(40));
Hz(24,1)=z(44)-z(33)-z(39);

AB(23,44)=Hcrude(z(48));
AB(23,33)=-AB(16,33);
AB(23,39)=-AB(20,39);
AB(23,48)=z(44)*dHcrude(z(48));
AB(23,34)=-AB(16,34);
AB(23,40)=-AB(20,40);
AB(24,44)=1;
AB(24,33)=-1;
AB(24,39)=-1;

%ADDITION OF WATER
Hz(25,1)=z(45)-z(43)-z(44);

AB(25,45)=1;
AB(25,43)=-1;
AB(25,44)=-1;

%2D1 - Desaltter - splitter
```



```
Hz(26,1)=z(49)+z(47)-z(45);
Hz(27,1)=z(50)-z(46);
Hz(28,1)=z(63)-z(46);

AB(26,49)=1;
AB(26,47)=1;
AB(26,45)=-1;
AB(27,50)=1;
AB(27,46)=-1;
AB(28,63)=1;
AB(28,46)=-1;

%Flow of crude through 2E4A/B to 2F201
%2E2A
Hz(29,1)=z(51)-z(49);

AB(29,51)=1;
AB(29,49)=-1;

%2E4B
Hz(30,1)=z(52)-z(51);

AB(30,52)=1;
AB(30,51)=-1;

%Splitter before 2F201
Hz(31,1)=z(53)+z(54)-z(52);

AB(31,53)=1;
AB(31,54)=1;
AB(31,52)=-1;

%2E4A
Hz(32,1)=z(55)*Hvbo(z(56))-z(58)*Hvbo(z(59)) + z(51)*Hcrude(z(57))-z(49)*Hcrude(z
(50));
Hz(33,1)=z(55)-z(58);

AB(32,55)=Hvbo(z(56));
AB(32,58)=-Hvbo(z(59));
AB(32,51)=Hcrude(z(57));
AB(32,49)=-Hcrude(z(50));
AB(32,56)=z(55)*dHvbo(z(56));
AB(32,59)=-z(58)*dHvbo(z(59));
AB(32,57)=z(51)*dHcrude(z(57));
AB(32,50)=-z(49)*dHcrude(z(50));
AB(33,55)=1;
AB(33,58)=-1;

%2E4B
Hz(34,1)=z(58)*Hvbo(z(59))-z(60)*Hvbo(z(61)) + z(52)*Hcrude(z(62))-z(51)*Hcrude(z
(57));
Hz(35,1)=z(58)-z(60);

AB(34,58)=-AB(32,58);
AB(34,60)=-Hvbo(z(61));
AB(34,52)=Hcrude(z(62));
AB(34,51)=-AB(32,51);
AB(34,59)=-AB(32,59);
AB(34,61)=-z(60)*dHvbo(z(61));
AB(34,62)=z(52)*dHcrude(z(62));
```

```
AB(34,57)=-AB(32,57);
AB(35,58)=1;
AB(35,60)=-1;
```

```
clear

%DATA ENTRY - start
load z_ini.dat          %initial measurement structure
%load z_d1.dat          %02TI204 deleted
%load z_d2.dat          %03TI086 deleted
z=z_ini;                %initial measurement structure
%z=z_d2;                %02TI204 deleted
%z=z_d2;                %03TI086 deleted
load s.dat              %initial measurement structure
%load s_d1.dat          %02TI204 deleted
%load s_d2.dat          %03TI086 deleted
%s=s_d1;                %02TI204 deleted
%s=s_d2;                %03TI086 deleted
%DATA ENTRY - stop

load fcdata.dat
nm=35;                  %Number of measurements
nu=28;
nv=nm+nu;               %Total number of variables
zm_indx=[1 3 11 43 17 21 27 53 54 35 36 2 10 12 8 26 34 14 16 18 22 20 24 28 30 4
0 46 50 56 59 57 61 62 38 32];
zini=ones(size(1:nv))'*100;
  [fcdat_rn,fcdat_cn]=size(fcdata);
  for n=1:fcdat_cn
    zini(zm_indx(n))=fcdata(1,n);
  end,
% compute yj - start
  yj=zeros(nv,1);
  yj=fyj(nv);
% yj(8)=1;                %02TI204 deleted
% yj(32)=1;               %03TI086 deleted
% compute yj - stop
  Ia=zeros(nv,nm);
  Ia_rn=1;
  Ib=zeros(nv,nu);
  Ib_rn=1;
  for n=1:nv
    if yj(n)==0;
      Ia(n,Ia_rn)=1;
      Ia_rn=Ia_rn+1;
    elseif yj(n)==1;
      Ib(n,Ib_rn)=1;
      Ib_rn=Ib_rn+1;
    end,
  end
  Iht=Ia';
  [Iht_rn,Iht_cn]=size(Iht);
  Iht(Iht_rn+1:Iht_rn+nu,:)=Ib';
  [Hz,AB]=network(z);
    A=AB*Ia;
    B=AB*Ib;
    x=(z'*Ia)';
    u=(z'*Ib)';
    c=A*x+B*u-Hz;
    [Q,R,E]=qr(B);
    rR=rank(R);
    rQ=rank(Q);
    [Q_rn,Q_cn]=size(Q);
    [R_rn,R_cn]=size(R);
```



```
    Q1=Q(:,1:rR);
    Q2=Q(:,rR+1:Q_cn);
    C=Q2'*A;
    d=Q2'*c;
%GLOBAL TEST - start
y=(zini'*Ia)';
v=rank(C);
w=C*y-d;
t=w'*inv(C*s*C')*w;
%GLOBAL TEST - stop
%MEASUREMENT TEST - start
xstar=(z'*Ia)';
zini=zini';
xini=(zini*Ia)';
r=xini-xstar;
rstar=inv(s)*r;
den=sqrt(C'*inv(C*s*C')*C);
Crowsum=0;
[C_nr,C_nc]=size(C);
for n=1:C_nr
  Crowsum=Crowsum+abs(C(n,:));
end,
for n=1:nm
  if abs(r(n))>1e-4
    zi(n)=abs(rstar(n))/den(n,n);
  end,
end,
%Measurement test zc = 3.15
%MEASUREMENT TEST - start - stop


%SUBROUTINES

function yj=fyj(nv)
%Set up the initial measurement structure
yj=zeros(nv,1);
yj(4)=1; yj(5)=1; yj(6)=1; yj(7)=1; yj(9)=1; yj(13)=1;
yj(15)=1; yj(19)=1; yj(23)=1; yj(25)=1; yj(29)=1; yj(31)=1;
yj(33)=1; yj(37)=1; yj(39)=1; yj(41)=1; yj(42)=1; yj(44)=1;
yj(45)=1; yj(47)=1; yj(48)=1; yj(49)=1; yj(51)=1; yj(52)=1;
yj(55)=1; yj(58)=1; yj(60)=1; yj(63)=1;
```

```
%SENSITIVITY ANALYSIS OF THE INDUSTRIAL CASE STUDY
clear

%DATA ENTRY - start
load z_d2.dat
load s_d2.dat
load Ianr.dat
%DATA ENTRY - stop
%VARIABLE INTIALISATION - start
nm=33;
nu=30;
nv=nu+nm;
S=s_d2;
z=z_d2;
FX=zeros(25,1);
FXout_i=1;
num=25;
yj=zeros(nv,1);
yj(4)=1;
yj(5)=1;
yj(6)=1;
yj(7)=1;
yj(9)=1;
%yj(8)=1;        %Gross error removed
yj(13)=1;
yj(15)=1;
yj(19)=1;
yj(23)=1;
yj(25)=1;
yj(29)=1;
yj(31)=1;
YJ(32)=1;       %Gross error removed
yj(33)=1;
yj(37)=1;
yj(39)=1;
yj(41)=1;
yj(42)=1;
yj(44)=1;
yj(45)=1;
yj(47)=1;
yj(48)=1;
yj(49)=1;
yj(51)=1;
yj(52)=1;
yj(55)=1;
yj(58)=1;
yj(60)=1;
yj(63)=1;
Ia=zeros(nv,nm);
Ia_rn=1;
Ib=zeros(nv,nu);
Ib_rn=1;
for n=1:nv
  if yj(n)==0;
    Ia(n,Ia_rn)=1;
    Ia_rn=Ia_rn+1;
  elseif yj(n)==1;
    Ib(n,Ib_rn)=1;
    Ib_rn=Ib_rn+1;
  end,
```



```
end
Iht=Ia';
[Iht_rn,Iht_cn]=size(Iht);
Iht(Iht_rn+1:Iht_rn+nu,:)=Ib';
[Hz,AB]=network(z);
A=AB*Ia;
B=AB*Ib;
x=(z'*Ia)';
u=(z'*Ib)';
c=A*x+B*u-Hz;           %Formation of system-stop
[Q,R,E]=qr(B);          %Reconciliation Start
rR=rank(R);
rQ=rank(Q);
[Q_rn,Q_cn]=size(Q);
[R_rn,R_cn]=size(R);
Q1=Q(:,1:rR);
Q2=Q(:,rR+1:Q_cn);
R11=R(1:rR,1:rR);
R12=R(1:rR,rR+1:R_cn);
C=Q2'*A;
C=C*Ianr;
d=Q2'*c;
[C_rn,dummy]=size(C);
x=(x'*Ianr)';
S=Ianr'*S*Ianr;
iCSCt=inv(C*S*C');
xh=(z'*Ia*Ianr)';
nm=25;
%VARIABLE INTIALISATION - stop
%SENSITIVITY ANALYSIS - start
for n=1:nm;
%%dVAR
     dVAR = zeros(nm);
     dVAR(n,n) = 1;
%%dIA
     for i=1:C_rn,
          for j=1:C_rn,
               dCSCt(i,j) = C(i,n)*C(j,n);,
          end,
     end
     diCSCt = -iCSCt*dCSCt*iCSCt;
     dx = -(dVAR*C'*iCSCt + S*C'*diCSCt)*C*x;
     DX(:,n)=dx;
end
DXus=DX;
%%SCALE FOR EFFECT - start
for i=1:nm;
     for j=1:nm,
          DX(i,j) = DX(i,j) / x(i);
          DX(j,i) = DX(j,i) * S(i,i);,
     end,
end
%%SCALE FOR EFFECT - stop
%SENSITIVITY ANALYSIS - start

sum=0;
for n=1:nm, sum=sum+abs(DX(n,:));, end
[ss,ii]=sort(sum);
DXout=ii';
DXout(:,2)=ss';
```

```
xho=xh;
for n=1:nm
  xh(n,n)=xh(n,1);
  if n~=1 xh(n,1)=0;, end
end,
load INIM.dat
z_ini=INIM(:,1);
y_crit=(z_ini'*Ia)';
y=y_crit;
S=s_d2;
nm=33;
indx=[1:33];
indx=indx*Ianr;
nnmm=[10 4];
for nnm=1:2
  m=indx(nnmm(nnm));
  nn=1;
  for nnn=0.2:0.2:1.8
    S(m,m)=nnn*S(m,m);
    [Hz,AB]=network(z_ini);
    z=z_ini;
  stop=0;
  while stop==0
    A=AB*Ia;
    B=AB*Ib;
    x=(z'*Ia)';
    u=(z'*Ib)';
    c=A*x+B*u-Hz;
    [Q,R,E]=qr(B);
    rR=rank(R);
    rQ=rank(Q);
    [Q_rn,Q_cn]=size(Q);
    [R_rn,R_cn]=size(R);
    Q1=Q(:,1:rR);
    Q2=Q(:,rR+1:Q_cn);
    R11=R(1:rR,1:rR);
    R12=R(1:rR,rR+1:R_cn);
    C=Q2'*A;
    d=Q2'*c;
    if rR<rQ
      i=inv(C*S*C');
      res=S*C'*i*(C*y-d);
      x=y-res;
    elseif rR==rQ
      Q=Q1;
    end,
    u=inv(R11)*(Q1'*c-Q1'*A*x);
    u=u'*E';
    zstar(1:nm,1)=x;
    zstar(nm+1:nv)=u;
    zstar=zstar'*Iht;
    zstar=zstar';
    [Hz,AB]=network(zstar);
    e=abs(abs(x)-abs(y_crit));
    [Me,ii]=max(e);
    if Me<1e-5 stop=1;, end,
    z=zstar;
    y_crit=x;
  end,
    xhp=x;
```



```
    xhp=(xhp'*Ianr)';
    S(m,m)=S(m,m)/nnn;
    FX(:,nn)=((abs(xhp)-abs(xho))'/xh*100)';
    nn=nn+1;
  end,
FXout(FXout_i:FXout_i+num-1,:)=FX
FXout_i=FXout_i+num+1;
end
```

	TAG	Initial measurements structures of the LSA										
		1	2	3	4	5	6	7	8	9	10	retro - fit
1	F1-FI201	1	0	1	1	1	0	1	1	0	1	0
2	T1-TI022	1	1	1	1	1	1	1	0	0	1	0
3	F2-FC208	1	1	1	1	0	0	1	0	1	0	0
4	T2	1	0	1	0	1	0	0	1	0	0	1
5	F3	1	1	1	1	1	1	1	1	1	0	1
6	T3	0	1	1	1	0	1	1	1	1	1	1
7	F4	0	1	0	1	1	1	1	1	1	1	1
8	T4-TI204	1	1	0	1	0	0	0	1	0	0	0
9	F5	1	1	0	1	1	0	0	1	1	0	1
10	T5-TI230	1	0	0	0	0	1	0	0	0	1	0
11	F6-FC302	0	1	0	0	1	1	1	0	1	1	0
12	T6-TI129	0	0	0	0	0	0	0	0	0	0	0
13	F7	0	1	0	0	1	0	1	0	1	0	1
14	T7-TI302	0	0	1	1	0	0	0	0	1	0	0
15	F8	1	1	1	1	0	1	0	1	0	1	1
16	T8-TI306	0	0	0	0	0	1	1	0	0	0	0
17	F9-FC207	1	1	1	0	0	1	1	1	1	1	0
18	T9-TI039	1	1	0	0	1	0	1	1	1	1	0
19	F10	0	0	0	0	0	1	1	1	1	1	1
20	T10-TI038	1	0	0	0	0	0	0	0	0	0	0
21	F11-FC318	1	0	1	1	1	1	0	0	0	0	0
22	T11-TI316	0	0	0	0	1	0	0	1	0	1	0
23	F12	1	1	1	1	1	1	1	0	1	1	1
24	T12-TI037	1	0	1	1	1	1	1	1	0	0	0
25	F13	1	0	0	1	1	0	0	1	1	1	1
26	T13-TI202	0	0	1	1	0	0	0	1	1	1	0
27	F14-FC029	1	1	1	0	0	1	1	0	1	0	0
28	T14-TI097	0	1	0	1	0	0	0	0	0	0	0
29	F15	1	0	1	1	1	1	1	1	1	1	1
30	T15-TI040	0	1	1	0	1	1	0	1	1	0	0
31	F16	0	1	1	0	1	1	1	1	0	1	1
32	T16-03TI086	0	0	0	0	0	1	1	0	1	1	0
33	F17	1	1	1	1	1	1	1	1	0	0	1
34	T17-TI102	0	0	0	0	1	1	0	1	0	1	0
35	F18-5FC015	1	0	0	0	0	1	1	1	0	0	0
36	F19-5FC012	0	1	0	0	1	0	0	0	0	1	0
37	F20	1	1	1	1	1	1	0	1	1	1	1
38	T20-05TI115	0	0	0	1	0	0	0	0	0	0	0
39	F21	0	1	0	1	1	1	0	0	1	1	1
40	T21-TI178	1	1	0	1	1	0	1	0	1	0	0
41	F22	0	0	1	1	0	0	0	0	1	1	1
42	T22	1	0	1	0	0	0	0	0	0	0	1
43	F23-FC001	0	1	1	0	1	0	0	0	1	0	0
44	F34	1	1	1	1	1	0	1	1	1	1	1
45	F24	1	0	0	1	0	1	1	1	0	1	1
46	T24-TI103	1	1	1	1	1	1	0	1	0	1	0
47	F25	1	1	1	0	1	1	1	1	1	1	1
48	T34	0	0	1	0	0	0	1	0	0	0	1
49	F26	0	1	1	0	0	0	1	0	1	1	1
50	T26-TI027	1	0	0	1	1	0	1	0	1	1	0
51	F28	0	0	0	1	1	1	0	1	0	1	1
52	F31	1	1	1	1	0	1	1	1	1	0	1
53	F32-FC300	0	0	1	1	0	1	0	0	1	1	0
54	F33-FC301	1	1	0	0	1	0	1	1	0	0	0
55	F27	0	0	0	1	1	0	1	0	0	1	1
56	T27-TI002	0	1	0	0	0	1	0	1	1	1	0
57	T28-TI028	0	1	0	0	0	0	0	0	1	0	0
58	F29	1	1	1	1	1	1	0	1	1	0	1
59	T29-TI029	1	0	1	1	0	0	1	1	0	1	0
60	F30	1	1	1	0	0	1	1	1	1	1	1
61	T30-TI001	0	0	0	0	1	1	0	0	0	0	0
62	T31-TI004	1	0	1	1	0	0	1	0	0	0	0
63	T25	0	1	1	0	1	1	1	1	1	0	1