Interval and Ellipsoidal Uncertainty Models

Andrzej Bargiela

Department of Computing, The Nottingham Trent University,

Nottingham NG1 4BU, UK

1 Introduction

In this Chapter, we present results derived in the context of state estimation of a class of real-life systems that are driven by some poorly known factors. For these systems, the representation of uncertainty as confidence intervals or the ellipsoids is shown to offer significant advantages over the more traditional approaches with probabilistic representation of noise. While the filtered-white-Gaussian noise model can be defined on grounds of mathematical convenience, its use is necessarily coupled with a hope that an estimator with good properties in idealised noise will still perform well in real noise. With good knowledge of the plant and its environment, a sufficiently accurate approximation to the probability density function can be obtained, but shortage of prior information or excessive computing demands normally rule out this option. A more realistic approach is to match the noise representation to the extent of prior knowledge. Both interval and ellipsoidal representation of noise illustrate the principle of keeping the noise model simple while allowing for iterative refinement of the noise as we proceed.

1.1 The system model

We start by introducing the deterministic model of the system

$$g(\mathbf{x}) = \mathbf{z} \tag{1}$$

where $\mathbf{x}$ is an n-dimensional state vector and $\mathbf{z}$ is an m-dimensional measurement vector. In the deterministic equation it is assumed that the true measurement vector $\mathbf{z}^{\mathbf{t}}$ is approximated well by the observed measurement vector $\mathbf{z}^{\mathbf{o}}$**.** Conversly, in the non-deterministic or uncertain model, all that is assumed is that the true measurement vector is contained in the region bounbed by $\mathbf{z}^{\boldsymbol{l}}$ and $\mathbf{z}^{\boldsymbol{u}}$. A measurement set *M* is defined as a collection of variables in the system for which real metered values or measurement estimates (pseudomeasurements) are available. A distinction is made between the measurement set *M* and the collection of values that this set would produce for a particular operating state. This measurement set at a particular instant of time, or for a particular assumed operating state, will produce a measurement vector $\mathbf{z^o} \in \mathbf{R^m}$, where *m* is cardinality of *M.* With the measurement set *M* producing measurement vector $\mathbf{z^o}$, the set of feasible measurement vectors is given by:

$$Z(M, \mathbf{z^o}) := \{ \mathbf{z^o} \in \mathbf{R^m} \mathbf{:} \, \mathrm{z_i}^{l} \leq \mathrm{z_i} \leq \mathrm{z_i}^{u} \text{ , i=1,...,m } \} \qquad (2)$$

where *m* is the cardinality of *M* and $\mathbf{z}^{\boldsymbol{l}}$ and $\mathbf{z}^{\boldsymbol{u}}$ are defined as above. $Z(M, \mathbf{z^o})$ defines a region of $\mathbf{R^m}$ in which the true measurement vector is contained. This region is the smallest that can be obtained within the limits of accuracy of the measurement set. In this format, the system equation (1) is replaced with the following set inclusion:

$$\mathrm{g}(\mathbf{x}) \in \; Z(M, \mathbf{z^o}) \qquad (3)$$

following from the assumption that the true measurement vector is unknown but constrained in $Z(M, \mathbf{z^o})$. This gives the set of feasible state vectors, $X(M, \mathbf{z^o})$, for measurement set *M* and measurement vector $\mathbf{z^o}$, as:

$$X(M, \mathbf{z^o}) := \{ \mathbf{x} \in \mathbf{R^n} \mathbf{:} \; \mathrm{g}(\mathbf{x}) \in Z(M, \mathbf{z^o}) \} \qquad (4)$$

Equation (3) will be referred to as the **uncertain system equation** with $X(M, \mathbf{z^o})$, of (4), representing the state uncertainty set. For the uncertain system equation there is no unique operating state that can be calculated. All that can be defined is a set of possible operating states resulting from the set of possible measurement vectors. No preference is placed on these, all are assumed to be equally likely. This reflects the lack of preference for a particular measurement vector in $Z(M, \mathbf{z^o})$ Although lack of a unique estimate of $\mathbf{x}$ is at first worrying, we can reassure ourselves by noticing that engineering design is largely a matter of tolerancing for adequate performance in the worst case. Also, this method does not make any unrealistic assumptions about the probabilistic properties of the measurement data, its expected values or its probabilistic variation. The unknown-but-bounded treatment of measurement uncertainty leads to this simple and flexible presentation of state estimate uncertainty. Feasible state estimates are specified by a sharply defined set, $X(M, \mathbf{z^o})$. This fits neatly with many intended engineering uses of state estimates. Questions such as: Is the system operating in an acceptable range? Has the system failed? If so, where is the fault located?; can be answered more easily and categorically when the range of possible operating states can be clearly defined.

When faced with measurement uncertainty in state estimation the most common response of engineers and researchers has been to try and produce estimates that best fit the measurement data in some way. The attraction of providing a single ‘optimal’ point-value in n-dimensional state space must however be ballanced by the lack of indication of how accurate this estimate is. It has long been recognised that the notion of optimality of such estimates is inherently linked to the implicit assumptions about the measurement noise, which are rarely satisfied. In which case, no particular ‘optimal’ state estimate can be identified.

To make the uncertainty in state estimates more accessible to engineering practice, uncertainty intervals or confidence limits, similar to those for measurement values, can be derived in the following way. Let

$$x_i^l := \min_{\mathbf{x} \in X(M,\mathbf{z^o})} x_i, \quad i=1,...,n$$

$$x_i^u := \max_{\mathbf{x} \in X(M,\mathbf{z^o})} x_i, \quad i=1,...,n \qquad (5)$$

The vectors $\mathbf{x}^l$ and $\mathbf{x}^u$ will provide lower and upper bounds on the state vector **x** in the same way that $\mathbf{z}^l$ and $\mathbf{z}^u$ did for the measurement vector. For each individual variable, the interval (x_i^l , x_i^u) is referred to as the **uncertainty interval** for the ith variable and x_i^l and x_i^u are referred to as **confidence limits**. These uncertainty intervals or confidence limits are as tight as can be achieved with the given measurement uncertainty. Calculating these bounds - the process referred to as confidence limit analysis – is discussed in the subsequent sections. If $X^*(M, \mathbf{z^o})$ is the set defined by these bounds, ie

$$X^*(M, \mathbf{z^o}) := \{ \mathbf{x} \in \mathbf{R^n} : x_i^l \le x_i \le x_i^u , i=1,...,n \} \qquad (6)$$

then it must be noted that $X^*(M, \mathbf{z^o})$ may not be the same as $X(M, \mathbf{z^o})$. Clearly, $X(M, \mathbf{z^o}) \subseteq X^*(M, \mathbf{z^o})$, but not every combination of values that are each feasible for individual state variables are feasible for a feasible state vector. Let $Z^*(M, \mathbf{z^o})$ be the subset of $\mathbf{R^m}$ onto which $X(M, \mathbf{z^o})$ is mapped by g(.) the system function. Then

$$Z^*(M, \mathbf{z^o}) := \{ \mathbf{z} \in \mathbf{R^m} : \mathbf{z}=g(\mathbf{x}), \mathbf{x} \in X(M, \mathbf{z^o}) \} \qquad (7)$$

$Z^*(M, \mathbf{z^o}) \subseteq Z(M, \mathbf{z^o})$, but these two sets are unlikely to be equal. There may be a $\mathbf{z} \in Z(M, \mathbf{z^o})$ for which there is no **x** (neither in $X(M, \mathbf{z^o})$ nor $\mathbf{R^n}$) for which g(**x**)=**z**. In other words, there may be vectors in $Z(M, \mathbf{z^o})$ that are inconsistent for g(.). These two remarks are illustrated in Figure 1 for the 2-dimensional case.

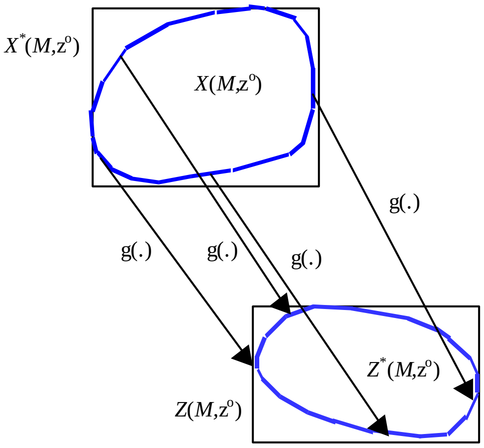
2 Confidence Limit Analysis

The process of calculating uncertainty bounds for the state estimates, which result from the measurement and pseudomeasurement uncertainty, is referred to as confidence limit analysis. Based on the model of uncertainty, described in the previous section, mathematical methods for calculating these confidence limits are now presented.

It can now be seen how the confidence limit analysis can be formulated as a series of mathematical optimisation problems. For each of the independent state variables, *i=1,...,n*

$$x_i^{l} = \min x_i \qquad \text{subject to } \mathbf{x} \in X(M, \mathbf{z^o}) \tag{8}$$

$$x_i^{u} = \max x_i \qquad \text{subject to } \mathbf{x} \in X(M, \mathbf{z^o}) \tag{9}$$



$X(M,z^o)$ – The state uncertainty set

$X^*(M,z^o)$ – The smallest box containing the state uncertainty set

$Z(M,z^o)$ – The measurement uncertainty set

$Z^*(M,z^o)$ – The image of the state uncertainty set when mapped by g(.)

Figure 1 Relationship between X^* and X, and between Z^* and Z

The nature of the system equations g(.) means that the optimisation problems of (8) and (9) are non-linear. Therefore the choice of optimisation technique to be used is not at all clear. With *n*, the number of state variables, and *m*, the number of measurements, confidence limit analysis requires 2*n* non-linear optimisations each subject to 2*m* constraints (the 2*m* constraints are supplied by the lower and upper bounds on the measurement uncertainty). For real-life systems there may be several hundred state variables and several hundred measurements and pseudomeasurements. Therefore, confidence limit analysis is a highly computationally intensive task which requires efficient optimisation techniques.

In the reminder of this section confidence algorithms are presented. These fall into two categories: non-linear and linearised methods.

2.1 Monte Carlo method

In normal use, deterministic state estimators produce one state estimate for one measurement vector. Used in this way they give no indication of how a state estimate may vary in response to variations in the measurement values. Alternatively, if a deterministic state estimator is used repeatedly for a whole range of measurement vectors then some indication of state estimate variability is provided. It is this idea that forms the basis to the Monte Carlo approach to confidence limit analysis. A large number of feasible state estimates are generated, as randomly as possible, and from these the state estimate confidence limits are estimated. The larger the number of random feasible state estimates the more reliable the confidence limits.

Let $\mathbf{z^i}$ be a measurement vector, selected randomly from the set $Z(M, \mathbf{z^o})$ of all feasible measurement vectors, and let $\mathbf{x^i}$ be a deterministic state estimate calculated from $\mathbf{z^i}$. $\mathbf{x^i}$ is a feasible state estimate if $\mathrm{g}(\mathbf{x^i}) \in Z(M, \mathbf{z^o})$. This follows from the definition of feasible vectors given in equation (4). It must be noted that $\mathrm{g}(\mathbf{x^i})$ is not necessarily equal to $\mathbf{z^i}$. If $\mathbf{z^i}$ is not a consistent vector, then there is no state vector $\mathbf{x}$ for which $\mathrm{g}(\mathbf{x^i}) = \mathbf{z^i}$. In fact, if $Z^*(M, \mathbf{z^o})$ is defined as in (7), then $Z^*(M, \mathbf{z^o}) = Z(M, \mathbf{z^o})$ only when M is a minimal measurement set (ie if M is an observable set and has no observable subset). For sequence, $\mathbf{z^1},..., \mathbf{z^k}$, of measurement vectors selected randomly from $Z(M, \mathbf{z^o})$, a sequence of sets $X^1,...,X^k$ can be defined, with

$$X^{\mathrm{j}} := \{ \mathbf{x^j} \in \mathbf{R^n} : \mathrm{g}(\mathbf{x^i}) \in Z(M, \mathbf{z^o}) \text{ for some i } \in \{1,...,\mathrm{j}\}\}, \mathrm{j}=1,...,\mathrm{k} \qquad (10)$$

where $\mathbf{x^j}$ is the state estimate calculated from $\mathbf{z^j}$. X^{j} is the set of feasible state estimates generated by the sequence of measurement vectors $\mathbf{z^1},..., \mathbf{z^j}$. This sequence of sets is such that $X^{\mathrm{j}} \subseteq X^{\mathrm{k}} \subseteq X(M, \mathbf{z^o})$ for all j=1,...,k-1, as only feasible state estimates are contained in X^{j}. For a large number, k, of randomly selected measurement vectors it can be assumed that X^{k} is approximately equal to $X(M, \mathbf{z^o})$. In other words, as $k \longrightarrow \infty$, $X^{\mathrm{k}} \longrightarrow X(M, \mathbf{z^o})$.

Before an algorithm description is given, we need to point out that the actual estimator used is of no importance, provided that it is unbiased and that it can guarantee convergence in a high proportion of cases. All state estimates are checked for feasibility before being used to update X^{j}. A sequence of random measurement vectors can be selected from $Z(M, \mathbf{z^o})$ by using a random number generator. For example, a sequence of random numbers, $r_1^{\mathrm{j}},..., r_{\mathrm{m}}^{\mathrm{j}}$, scaled to be between 0.0 and 1.0, can be generated and used to construct the measurement vector $\mathbf{z^j}$

$$\mathrm{z_i^j} = \mathrm{z_i}^{l} + r_{\mathrm{i}}^{\mathrm{j}} .(\mathrm{z_i}^{u} - \mathrm{z_i}^{l}), \mathrm{i}=1,...,\mathrm{m} \qquad (11)$$

where $\mathbf{z}^{\boldsymbol{l}}$ and $\mathbf{z}^{\boldsymbol{u}}$ are the lower and upper bounds for $Z(M, \mathbf{z^o})$. $\mathbf{z^{j+1}}$ can be constructed in a similar way from a new sequence of random numbers. Throughout the computations, for X^{j}, only two vectors need to be stored, these are

$\mathbf{x}^{\mathbf{j}l}$ and $\mathbf{x}^{\mathbf{j}u}$, the lower and upper bounding vectors for the current set of feasible state estimates, X^{j}. These vectors are updated whenever a new feasible vector, not contained in any of the X^{j} ‘s, is found.

The Monte Carlo confidence limit algorithm

1. Select a large number, k (to limit the number of simulations) and set $i = 0$.
2. Set $i = i + 1$.
3. Select a sequence of m random numbers, r_1^{i} ,..., $r_{\mathrm{m}}^{\mathrm{i}}$, and use these to construct a random measurement vector $\mathbf{z}^{\mathbf{i}}$ from $Z(M, \mathbf{z}^{\mathbf{o}})$ as described in (11)
4. Calculate a state estimate, $\mathbf{x}^{\mathbf{i}}$ from $\mathbf{z}^{\mathbf{i}}$. If $\mathrm{g}(\mathbf{x}^{\mathbf{i}}) \in Z(M, \mathbf{z}^{\mathbf{o}})$, then use $\mathbf{x}^{\mathbf{i}}$ tu update $\mathbf{x}^{\mathbf{i}l}$, $\mathbf{x}^{\mathbf{i}u}$. Otherwise reject $\mathbf{x}^{\mathbf{i}}$ as infeasible.
5. If $i < k$, go back to step 2. Otherwise stop.

Monte Carlo method is obviously slow computationally, but despite this it is useful in some situations. The condition that only feasible state estimates are used to update $\mathbf{x}^{\mathbf{i}l}$ and $\mathbf{x}^{\mathbf{i}u}$ makes the procedure mathematically reliable and ensures that these bounds can be attained. The method can be used as a yardstick, against which the accuracy of all other confidence limit algorithms can be compared. Unfortunately, the method is impractical in many real-time applications.

Other more practical methods are described in subsequent sections. Firstly the problem is linearised and a linear version of the state uncertainty model is presented in section 2.2. In sections 2.3, 2.4 and 2.5, three confidence limit algorithms are presented. These are: the linear programming method; the sensitivity method and the ellipsoid method.

2.2 Linearised confidence limit analysis

Suppose that $\hat{\mathbf{x}}$ is the state estimate calculated from the measurement vector $\mathbf{z}^{\mathbf{o}}$, where $\mathbf{z}^{\mathbf{o}}$ is defined as in section 1. The non-linear system function, g(.), can be linearised around $\hat{\mathbf{x}}$ using a first order Taylor approximation to give:

$$\mathrm{g}(\mathbf{x}) \approx \mathrm{g}(\hat{\mathbf{x}}) + J.\,(\mathbf{x} - \hat{\mathbf{x}}) \tag{12}$$

for all state vectors close to $\hat{\mathbf{x}}$. In (12), J is the Jacobian matrix evaluated at $\hat{\mathbf{x}}$. g(.) can be linearised around any state vector, $\hat{\mathbf{x}}$, this need not necessarily be the state estimate for $\mathbf{z}^{\mathbf{o}}$. It is better, however, to use a value for $\hat{\mathbf{x}}$ that is in some way central to the set of feasible state vectors. This is because, the approximation used in (12) is more accurate for values of $\mathbf{x}$ for which $\| \mathbf{x} - \hat{\mathbf{x}} \|$ is small. The best estimate available for the centre of $X(M, \mathbf{z}^{\mathbf{o}})$ is the state estimate calculated from

$\mathbf{z^o}$. In section 1, a feasible state vector was described as one for which $g(\mathbf{x}) \in Z(M, \mathbf{z^o})$. This condition can be linearised using (12) to give a linear approximation, $X^1(M, \mathbf{z^o})$, of the state uncertainty set $X(M, \mathbf{z^o})$.

This is defined as follows:

$$X^1(M, \mathbf{z^o}) := \{\, \mathbf{x} \in \mathbf{R^n} : g(\hat{\mathbf{x}}) + J.\,(\mathbf{x} - \hat{\mathbf{x}}) \in Z(M, \mathbf{z^o}) \,\} \tag{13}$$

$X^1(M, \mathbf{z^o})$ will be referred to as the linearised state uncertainty set. Substituting $\mathbf{dx}$ for $\mathbf{x} - \hat{\mathbf{x}}$ and using the definition of $Z(M, \mathbf{z^o})$ given in (2), $X^1(M, \mathbf{z^o})$ can be redefined as:

$$X^1(M, \mathbf{z^o}) := \{\, \mathbf{x} \in \mathbf{R^n} : \mathbf{x} = \hat{\mathbf{x}} + \mathbf{dx},\ \mathbf{z}^{\boldsymbol{l}} - g(\hat{\mathbf{x}}) \leq J.\mathbf{dx} \leq \mathbf{z}^{\boldsymbol{u}} - g(\hat{\mathbf{x}}) \} \tag{14}$$

It is easy to see that these two definitions, (13) and (14), of $X^1(M, \mathbf{z^o})$ are equivalent. For the reasons given in section 1, the set $X^1(M, \mathbf{z^o})$ will not be calculated explicitly. Rather, the smallest 'box' or orthotope containing $X^1(M, \mathbf{z^o})$ is sought. This set will be denoted by $X^{1*}(M, \mathbf{z^o})$ and referred to as the linearised state uncertainty box. Following the definition of $\mathbf{x}^{\boldsymbol{l}}$ and $\mathbf{x}^{\boldsymbol{u}}$ in (5), lower and upper limits for $X^1(M, \mathbf{z^o})$ can be defined as follows:

$$x^{1\,\boldsymbol{l}}_{i} := \min_{\mathbf{x} \in X^1(M, \mathbf{z^o})} x_i, \qquad i=1,...,n$$

$$x^{1\,\boldsymbol{u}}_{i} := \max_{\mathbf{x} \in X^1(M, \mathbf{z^o})} x_i, \qquad i=1,...,n \tag{15}$$

The definition of state uncertainty has been linearised in this way to allow confidence limit algorithms based on linear programming methods. Calculating the bounding vectors of $X^1(M, \mathbf{z^o})$ can easily be formulated as a linear programming problem. To allow this, some new notation is introduced:

$$\mathbf{dz}^{\boldsymbol{l}} := \mathbf{z}^{\boldsymbol{l}} - g(\hat{\mathbf{x}}) \tag{16}$$

$$\mathbf{dz}^{\boldsymbol{u}} := \mathbf{z}^{\boldsymbol{u}} - g(\hat{\mathbf{x}}) \tag{17}$$

$$DZ(M, \mathbf{z^o}) := \{\, \mathbf{dz} \in \mathbf{R^m} : g(\hat{\mathbf{x}}) + \mathbf{dz} \in Z(M, \mathbf{z^o}) \,\} \tag{18}$$

$$DX^1(M, \mathbf{z^o}) := \{\, \mathbf{dx} \in \mathbf{R^n} : \hat{\mathbf{x}} + \mathbf{dx} \in X^1(M, \mathbf{z^o}) \,\} \tag{19}$$

$DX^1(M, \mathbf{z^o})$ is just the set $X^1(M, \mathbf{z^o})$ shifted by $\hat{\mathbf{x}}$, $DZ(M, \mathbf{z^o})$ is the measurement uncertainty set shifted by $g(\hat{\mathbf{x}})$ and $\mathbf{dz}^{\boldsymbol{l}}$ and $\mathbf{dz}^{\boldsymbol{u}}$ represent 'tightest' lower and upper bounds for the set $DX^1(M, \mathbf{z^o})$. Then, the i^{th} element of $\mathbf{dx}^{\boldsymbol{l}}$ can be found by solving the linear programming problem

$$\textit{minimise } dx_i \tag{20}$$
$$\textit{subject to } \mathbf{dz}^{\boldsymbol{l}} \leq J.\mathbf{dx} \leq \mathbf{dz}^{\boldsymbol{u}}$$

Similarly, the i^{th} element of $\mathbf{dx}^{\boldsymbol{u}}$ can be found by solving the corresponding linear programming problem

$$\textit{maximise } dx_i \tag{21}$$
$$\textit{subject to } \mathbf{dz}^{\boldsymbol{l}} \leq J.\mathbf{dx} \leq \mathbf{dz}^{\boldsymbol{u}}$$

Hence by performing *2n* linear programs, the vectors $\mathbf{dx}^{\boldsymbol{l}}$ and $\mathbf{dx}^{\boldsymbol{u}}$ can be constructed. Once $\mathbf{dx}^{\boldsymbol{l}}$ and $\mathbf{dx}^{\boldsymbol{u}}$ have been calculated, it is a simple matter to construct the bounds

$$\mathbf{x}^{1\boldsymbol{l}} = \hat{\mathbf{x}} + \mathbf{dx}^{\boldsymbol{l}} \tag{22}$$

$$\mathbf{x}^{1\boldsymbol{u}} = \hat{\mathbf{x}} + \mathbf{dx}^{\boldsymbol{u}} \tag{23}$$

Three special cases can be identified:

(i) g($\hat{\mathbf{x}}$) = $\mathbf{z}^{\mathbf{o}}$ where $\mathbf{z}^{\mathbf{o}}$ is the measurement vector from which $\hat{\mathbf{x}}$ was calculated as a state estimate. When this situation occurs, $\mathbf{dz}^{\boldsymbol{u}} = -\mathbf{dz}^{\boldsymbol{l}} = \mathbf{e}^{\mathbf{z}}$, in other words, $Z(M,\mathbf{z}^{\mathbf{o}})$ is symmetric about g($\hat{\mathbf{x}}$). This symmetry is carried over to the linearised state uncertainty set , hence $\mathbf{dx}^{\boldsymbol{u}} = -\mathbf{dx}^{\boldsymbol{l}}$. This means that only *n* linear programming problems need to be solved, those in (19) say. It must be noted that a general measurement vector $\mathbf{z}^{\mathbf{o}}$ will suffer from inconsistency, which means that there will be no state vector, $\mathbf{x}$, in $\mathbf{R}^{\mathbf{n}}$ for which g($\mathbf{x}$) is equal to $\mathbf{z}^{\mathbf{o}}$.

(ii) *M* is an observable set and no subset of *M* is observable. In other words, *M* is a minimal measurement set. Consequently, all $\mathbf{z}$ in $Z(M, \mathbf{z}^{\mathbf{o}})$ are consistent and, in particular, $\mathbf{z}^{\mathbf{o}}$ is consistent. Hence, there is an $\hat{\mathbf{x}} \in \mathbf{R}^{\mathbf{n}}$ for which g($\hat{\mathbf{x}}$)=$\mathbf{z}^{\mathbf{o}}$. If $\hat{\mathbf{x}}$ can be found, only *n* linear programs need be performed (this follows from case *(i)*). Furthermore, when *J* is the Jacobian matrix for *M*, evaluated at $\hat{\mathbf{X}}$, and *M* is a minimal measurement set, *J* is square and non-singular. Hence, there exists an inverse, J^{-1}, for *J*. The lower and upper bounds for $DX^1(M, \mathbf{z}^{\mathbf{o}})$, and hence for $X^1(M, \mathbf{z}^{\mathbf{o}})$, can be calculated from the individual rows of J^{-1}, without help from linear programming methods. Before this can be done, we prove two lemmas:

Lemma 1 Let *M* be a minimal measurement set, $\mathbf{z}^{\mathbf{o}}$ be an observed measurement vector from *M*, $\hat{\mathbf{x}}$ be the state estimate for $\mathbf{z}^{\mathbf{o}}$ and *J* be the non-singular Jacobian matrix defined by *M* and $\hat{\mathbf{x}}$. Then, for $DX^1(M, \mathbf{z}^{\mathbf{o}})$, defined as in (19)

$$DX^1(M, \mathbf{z}^{\mathbf{o}}) := \{ \ \mathbf{dx} \in \mathbf{R}^{\mathbf{n}} : \mathbf{dx} = J^{-1}.\mathbf{dz}, \ \mathbf{dz}^{\boldsymbol{l}} \leq J.\mathbf{dx} \leq \mathbf{dz}^{\boldsymbol{u}}) \ \} \tag{24}$$

Proof By definitions (16), (17), (18) and (19), $\mathbf{dx} \in DX^1(M, \mathbf{z}^{\mathbf{o}})$ if and only if $\mathbf{dz}^{\boldsymbol{l}} \leq J.\mathbf{dx} \leq \mathbf{dz}^{\boldsymbol{u}}$. Putting $\mathbf{dz} = J.\mathbf{dx}$, $\mathbf{dz}^{\boldsymbol{l}} \leq J.\mathbf{dx} \leq \mathbf{dz}^{\boldsymbol{u}}$ when and only when $J^{-1}.\mathbf{dz} = \mathbf{dx}$ and $\mathbf{dz}^{\boldsymbol{l}} \leq \mathbf{dz} \leq \mathbf{dz}^{\boldsymbol{u}}$. Hence, $DX^1(M, \mathbf{z}^{\mathbf{o}})$ is equal to the set { $\mathbf{dx} \in \mathbf{R^n}$ **:** $\mathbf{dx}= J^{-1}.\mathbf{dz}$**,** $\mathbf{dz}^{\boldsymbol{l}} \leq J.\mathbf{dx} \leq \mathbf{dz}^{\boldsymbol{u}}$) } and the lemma is proved.

Lemma 2 Let M be a minimal measurement set, $\mathbf{z^o}$ be an observed measurement vector from M, $\hat{\mathbf{x}}$ be the state estimate for $\mathbf{z^o}$ and let J be the non-singular Jacobian matrix defined by M and $\hat{\mathbf{x}}$. For i=1,…,n, the i^{th} element of the lower bounding vector for $DX^1(M, \mathbf{z^o})$, $\mathrm{dx}_i^{\,l}$, is given by

$\mathrm{dx}_i^{\,l} = \mathbf{a^i}.\mathbf{dz}^*$, where

$$\mathrm{dz}_j^{\,*} = \{ \begin{array}{ll} \mathrm{dz}_j^{\,u} & \text{if } a_j^{\,i} < 0.0, \\ \mathrm{dz}_j^{\,l} & \text{otherwise}\} \end{array} \qquad (25)$$

where $\mathbf{a^i}$ is the i^{th} row of the matrix J^{-1}.

Proof By definition (20), $\mathrm{dx}_i^{\,l}$ is the minimum value for dx_i subject to **dx** being a member of $DX^1(M, \mathbf{z^o})$. Using Lemma 1, this is the same as the minimum value of $(J^{-1}.\mathbf{dz})_i$ subject to $\mathbf{dz}^{\boldsymbol{l}} \leq \mathbf{dz} \leq \mathbf{dz}^{\boldsymbol{u}}$. For any **dz**, the product $(J^{-1}.\mathbf{dz})_i = \mathbf{a^i}.\mathbf{dz}$, where $\mathbf{a^i}$ is the the i^{th} row of the matrix J^{-1}. As to $\mathbf{dz}^{\boldsymbol{l}} \leq \mathbf{dz} \leq \mathbf{dz}^{\boldsymbol{u}}$ is the only constraint on the elements of the vector **dz**, the minimum value of $\mathbf{a^i}.\mathbf{dz}$ is equal to the sum of minimum values of $a_j^{\,i}.\mathrm{dz}_j$ for j=1,…,n. The minimum value of $a_j^{\,i}.\mathrm{dz}_j$, subject to of $\mathrm{dz}_j^{\,l} \leq \mathrm{dz}_j \leq \mathrm{dz}_j^{\,u}$ is just $a_j^{\,i}\,.\mathrm{dz}_j^{\,u}$ or $a_j^{\,i}\,.\mathrm{dz}_j^{\,l}$, depending whether $a_j^{\,i}$ is less than or greater than 0.0, respectively. When $a_j^{\,i}$ = 0.0, $a_j^{\,i}\,.\mathrm{dz}_j$ = 0.0 as well, so such elements do not contribute to the value of $\mathrm{dx}_i^{\,l}$. The lemma now follows.

Once the inverse of J has been calculated, Lemma 2 can be applied to each element of $\mathbf{dx}^{\boldsymbol{l}}$ in turn, providing a straightforward way of calculating this vector that does not rely on optimisation methods. Because of the symmetry in this situation, the upper bound for $DX^1(M, \mathbf{z^o})$, $\mathbf{dx}^{\boldsymbol{u}}$, is equal to the negative of the lower bound. So, Lemma 2 need only be applied n times. The bounding vectors, $\mathbf{x}^{1\boldsymbol{l}}$ and $\mathbf{x}^{1\boldsymbol{u}}$, are found by adding $\hat{\mathbf{x}}$ to $\mathbf{dx}^{\boldsymbol{l}}$ and $\mathbf{dx}^{\boldsymbol{u}}$, by equations (22) and (23).

(iii) $DX^1(M, \mathbf{z^o})$, and hence $X^1(M, \mathbf{z^o})$, may be empty, even when $Z(M, \mathbf{z^o})$ is non-empty. If this situation occurs then $Z(M, \mathbf{z^o})$ is said to be inconsistent. This is reflected in the bounding vectors, by $x^1{}_i^{\,u} \leq x^1{}_i^{\,l}$ for at least one i in 1,…,n.

The linearised state uncertainty set, $X^1(M, \mathbf{z^o})$, is only an approximation to the true state uncertainty set, $X(M, \mathbf{z^o})$. The question – how good an approximation is it? - now arises. To answer this question, an upper bound on the difference between $x^1{}_i^{\,u}$ and $x_i^{\,u}$ – the upper limits on the feasible values for the i^{th} state variable in $X^1(M, \mathbf{z^o})$ and $X(M, \mathbf{z^o})$, respectively – is derived. This is furnished by the following lemma.

Lemma 3 For a measurement uncertainty set $Z(M,\ \mathbf{z}^{\mathbf{o}})$, where M is a minimal measurement set and $\mathbf{z}^{\mathbf{o}}$ is an observed measurement vector derived from M. Let $\hat{\mathbf{x}}$ be the state estimate calculated from $\mathbf{z}^{\mathbf{o}}$ and let J be the Jacobian matrix calculated at $\hat{\mathbf{x}}$. For the i^{th} state variable, i=1,…,n,

$$| x_i^u - {x^1}_i^u | \leq \| \mathbf{a}^{\mathbf{i}} \| . \| \mathbf{e} \| \qquad (26)$$

where x_i^u and ${x^1}_i^u$ are the upper limits for the i^{th} state variable in the true and linearised state uncertainty sets respectively, $\mathbf{a}^{\mathbf{i}}$ is the i^{th} row of J^{-1} and $\mathbf{e}$ is a vector for which:

$$\| \mathbf{e} \| = O\,(\| \mathbf{x} - \hat{\mathbf{x}} \|^2\ ) \qquad (27)$$

Here, $\mathbf{x}$ is a feasible state vector in $X(M, \mathbf{z}^{\mathbf{o}})$.

Proof The first point to note is that as M is a minimal measurement set, J^{-1} and $\mathbf{a}^{\mathbf{i}}$ are well defined. The lemma is proved in two cases; for $x_i^u \geq x_i^{1u}$ and $x_i^u < {x^1}_i^u$, respectively.

Case (i): $x_i^u \geq x_i^{1u}$. Let $\mathbf{x}^*$ be a vector in $X(M,\ \mathbf{z}^{\mathbf{o}})$ for which the i^{th} state variable attains its upper bound, ie $\mathbf{x}^*$ is a vector for which $x_i^* = x_i^u$. As $\mathbf{x}^*$ is a feasible member of $X(M, \mathbf{z}^{\mathbf{o}})$ there is a measurement vector, $\mathbf{z}^*$ in $Z(M, \mathbf{z}^{\mathbf{o}})$ for which $g(\mathbf{x}^*)=\mathbf{z}^*$. A vector $\mathbf{x}^{**}$ can be defined equal to $J^{-1}.(\mathbf{z}^*\text{-}g(\hat{\mathbf{x}}))+\hat{\mathbf{x}}$. This is just the vector in $X^1(M,\ \mathbf{z}^{\mathbf{o}})$ associated with $\mathbf{z}^*$ under the relationship given in the definition of $X^1(M,\ \mathbf{z}^{\mathbf{o}})$ by (13). The vector $\mathbf{x}^{**}$ is a member of $X^1(M,\ \mathbf{z}^{\mathbf{o}})$, so $x_i^{**} \leq {x^1}_i^u$, as ${x^1}_i^u$ is the maximum value that the i^{th} state variable can take in $X^1(M, \mathbf{z}^{\mathbf{o}})$. Hence

$$| x_i^u - {x^1}_i^u | \leq | x_i^u - x_i^{**} | = | x_i^* - x_i^{**} | \qquad (28)$$

Attention is now focused on the difference $| x_i^* - x_i^{**} |$. From the definition of $\mathbf{x}^{**}$,

$$| x_i^* - x_i^{**} | = | x_i^* - [J^{-1}.(\mathbf{z}^*\text{-}g(\hat{\mathbf{x}})) + \hat{\mathbf{x}}\ ]_{\mathrm{i}} | \qquad (29)$$

Since vector $\mathbf{z}^*$ in the *RHS* of this equation is defined as equal to $g(\mathbf{x}^*)$, the Taylor approximation of (12), gives

$$\mathbf{z}^* = g(\mathbf{x}^*) = g(\hat{\mathbf{x}}) + J.\ (\mathbf{x}^* - \hat{\mathbf{x}}) + \mathbf{e} \qquad (30)$$

for a vector $\mathbf{e}$ of order $O\,(\|\ \mathbf{x} - \hat{\mathbf{x}} \|^2\ )$. Substituting for $\mathbf{z}^*$ in (29) and cancelling gives

$$| x_i^* - x_i^{**} | = | x_i^* - [\mathbf{x}^* + J^{-1}.\ \mathbf{e}]_{\mathrm{i}} | \qquad (31)$$

The i^{th} element of $J^{-1}.\ \mathbf{e}$ is simply $\mathbf{a}^{\mathbf{i}}.\ \mathbf{e}$, where $\mathbf{a}^{\mathbf{i}}$ is the i^{th} row of J^{-1}. Combining (28), (31) with this last remark, and using the *Cauchy-Schwartz* inequality it follows that | x_i^u - $x^1{}_i{}^u$| ≤ || $\mathbf{a^i}$ ||.|| $\mathbf{e}$ ||, which proves the lemma in this case.

Case (ii): $x_i^u < x^1{}_i{}^u$. Let $\mathbf{x}^+$ be a vector in $X^1(M, \mathbf{z^o})$ for which the i^{th} variable attains its upper limit. If $\mathbf{z}^+$ is defined as equal to g($\hat{\mathbf{x}}$) - $J.$ ($\mathbf{x}^+$ - $\hat{\mathbf{x}}$) and $\mathbf{x}^{++}$ is the state vector in $X(M, \mathbf{z^o})$ for which g($\mathbf{x}^{++}$)= $\mathbf{z}^+$ (this exists as M is a minimal measurement set and so $\mathbf{z}^+$ must be consistent), then a similar argument to that in case (i) gives:

$$| x^1{}_i{}^u - x_i^u| \leq | x_i{}^+ - x_i{}^{++}| \leq || \mathbf{a^i} ||.|| \mathbf{e} || \qquad (32)$$

where $\mathbf{a^i}$ is the i^{th} row of J^{-1} and || $\mathbf{e}$ || is of order O (||$\mathbf{x}^{++}$-$\hat{\mathbf{x}}$ ||^2). The lemma is therefore proven.

It can be assumed, without the loss of generality, that the Jacobian martix is scaled so that ||J|| is of order unity. When this is the case, and when J is not ill-conditioned, ||$\mathbf{a^i}$|| will also be of order unity. This means that the maximum discrepancy between the upper limits for any of the state variables in $X(M, \mathbf{z^o})$ and $X^1(M, \mathbf{z^o})$, respectively, is of order O (||$\mathbf{x}$- $\hat{\mathbf{x}}$ ||^2), where $\mathbf{x}$ is a feasible vector in $X(M, \mathbf{z^o})$. In other words, the accuracy of the linearised state uncertainty set is of the same magnitude as that of the Taylor approximation in (12) and so the discrepancy between the $\mathbf{x^u}$ and $\mathbf{x}^{1\mathbf{u}}$ will not rise significantly when the confidence limit analysis problem is treated in this linearised form. It should be noted that the results of Lemma 3 provide only an upper bound for the discrepancy between in $X(M, \mathbf{z^o})$ and $X^1(M, \mathbf{z^o})$, the true magnitude of this discrepancy will, in most cases, be much less than this bound.

Lemma 3 was stated for a minimal measurement set. This is in fact the extreme case. For an over-determined measurement set the state uncertainty sets will be smaller. So, the bounds of Lemma 3 can also be applied in the over-determined case.

2.3 Linear programming method

The general result that the confidence limits on state variables can be found by solving *2n* linear programs with *2m* constraints (equations (20), (21), (22) and (23)) is significant but a direct application of the revised simplex, or any similar linear programming algorithm, is likely to be too time consuming. In this section an alternative format for this problem is presented. It will be assumed that the measurement set M is both observable and over-determined. If M is not observable, then the uncertainty set is unbounded. If M is a minimal observable measurement set, then confidence limit analysis can be performed more efficiently using the method given in special case (ii) of the previous section.

Without loss of generality, it can be assumed that the elements of the measurement set M are ordered so that the first n elements correspond to an observable set of measurements. For any measurement vector, $\mathbf{z} \in Z(M, \mathbf{z^o})$, two new vectors can be defined. These are $\mathbf{z^n} \in \mathbf{R^n}$, the vector containing these first n elements of vector $\mathbf{z}$ and $\mathbf{z^{m\text{-}n}} \in \mathbf{R^{m\text{-}n}}$, the vector containing the remaining m-n elements. In the same way we define $\mathbf{dz^n}$, $(\mathbf{dz^n})^l$, $(\mathbf{dz^n})^u \in \mathbf{R^n}$, and $\mathbf{dz^{m\text{-}n}}$, $(\mathbf{dz^{m\text{-}n}})^l$, $(\mathbf{dz^{m\text{-}n}})^u \in \mathbf{R^{m\text{-}n}}$. New matrices J^{n} and $J^{\mathrm{m\text{-}n}}$ can also be defined, J^{n} consisting of the first n rows of J and $J^{\mathrm{m\text{-}n}}$ the remaining rows.

Lemma 4 The maximisation of (21) is equivalent to

$$\begin{array}{lll} \textit{maximise} & \mathbf{a^i}.\mathbf{dz^n} & (33) \\ \textit{subject to} & (\mathbf{dz^n})^l \leq \mathbf{dz^n} \leq (\mathbf{dz^n})^u & \\ & (\mathbf{dz^{m\text{-}n}})^l \leq J^{\mathrm{m\text{-}n}}(J^{\mathrm{n}})^{-1}\mathbf{dz^n} \leq (\mathbf{dz^{m\text{-}n}})^u & \end{array}$$

where $\mathbf{a^i}$ is the i^{th} row of $(J^{\mathrm{n}})^{-1}$.

Proof The first point to note is that the observability of the first n measurements ensures that J^{n} is non-singular. So $\mathbf{a^i}$ and $J^{\mathrm{m\text{-}n}}(J^{\mathrm{n}})^{-1}$ are well defined. Let $\mathbf{dz^n} \in \mathbf{R^n}$ be a feasible vector by the condition in (33). That is $(\mathbf{dz^n})^l \leq \mathbf{dz^n} \leq (\mathbf{dz^n})^u$ and $(\mathbf{dz^{m\text{-}n}})^l \leq J^{\mathrm{m\text{-}n}}(J^{\mathrm{n}})^{-1}\mathbf{dz^n} \leq (\mathbf{dz^{m\text{-}n}})^u$. As J^{n} is non-singular, there is a unique $\mathbf{dx}' \in \mathbf{R^n}$ such that $J^{\mathrm{n}}.\mathbf{dx}' = \mathbf{dz^n}$. So, the first constraint of (33) can be written as $(\mathbf{dz^n})^l \leq J^{\mathrm{n}}.\mathbf{dx}' \leq (\mathbf{dz^n})^u$.

Also, $J^{\mathrm{m\text{-}n}}.\mathbf{dx}' = J^{\mathrm{m\text{-}n}}(J^{\mathrm{n}})^{-1}\mathbf{dz^n}$ which means, by the second constraint of (33), that $(\mathbf{dz^{m\text{-}n}})^l \leq J^{\mathrm{m\text{-}n}}.\mathbf{dx}' \leq (\mathbf{dz^{m\text{-}n}})^u$. Consequently, $\mathbf{dx}'$ is acertained to satisfy the constraints of (21).

Conversely, let $\mathbf{dx} \in \mathbf{R^n}$ be a vector satisfying the constraints of (21), and let $(\mathbf{dz^n})' = J^{\mathrm{n}}.\mathbf{dx}$. As the vector $\mathbf{dx}$ is feasible, $(\mathbf{dz^n})^l \leq (\mathbf{dz^n})' \leq (\mathbf{dz^n})^u$. Also $(\mathbf{dz^{m\text{-}n}})^l \leq J^{\mathrm{m\text{-}n}}(J^{\mathrm{n}})^{-1}(\mathbf{dz^n})' \leq (\mathbf{dz^{m\text{-}n}})^u$. That is, $(\mathbf{dz^n})'$ is feasible in (33). It can bee seen therefore that there is one-to-one correspondence between the feasible $\mathbf{dx}$ in (21) and the feasible $\mathbf{dz^n}$ in (33). More precisely, $\mathbf{dx}$ is feasible in (21) if and only if $J^{\mathrm{n}}.\mathbf{dx}$ is feasible in (33). To complete the proof, we need to show that for all feasible $\mathbf{dx}$ by (21) the two cost functions are the same. Let $\mathbf{dz^n}$ be feasible by (33) and $\mathbf{dx}' = (J^{\mathrm{n}})^{-1}\mathbf{dz^n}$. It is easy to see that $\mathrm{dx}'_{\mathrm{i}} = \mathbf{a^i}.\mathbf{dz^n}$, where $\mathbf{a^i}$ is the i^{th} row of $(J^{\mathrm{n}})^{-1}$, which completes the proof.

In just the same way, it can be shown that the minimisation

$$\begin{array}{lll} \textit{minimise} & \mathbf{a^i}.\mathbf{dz^n} & (34) \\ \textit{subject to} & (\mathbf{dz^n})^l \leq \mathbf{dz^n} \leq (\mathbf{dz^n})^u & \\ & (\mathbf{dz^{m\text{-}n}})^l \leq J^{\mathrm{m\text{-}n}}(J^{\mathrm{n}})^{-1}\mathbf{dz^n} \leq (\mathbf{dz^{m\text{-}n}})^u & \end{array}$$

is equivalent to the minimisation given in (20).

As before, the bounds $\mathbf{dx}^{l}$ and $\mathbf{dx}^{u}$ can be constructed for $DX^{1}(M, \mathbf{z^o})$ by performing $2n$ maximisations and minimisations of this form. Also the bounds $\mathbf{x}^{1l}$ and $\mathbf{x}^{1u}$ for $X^{1}(M, \mathbf{z^o})$ can be evaluated by adding $\hat{\mathbf{x}}$ to $\mathbf{dx}^{l}$ and $\mathbf{dx}^{u}$.

This formulation of the problem has an important advantage over the formulation of (21). While the linear program of (21) has $2m$ constraints the form of (33) has only $2(m-n)$. In many real-life systems measurement redundancy is low so $m-n<<m$. A disadvantage of the second formulation is that it requires the inversion of the martix J^{n}. However J^{n} need only to be inverted once while the maximisations and minimisations are carried out $2n$ times. So, with an efficient matrix inversion scheme this disadvantage quickly disappears.

The linear programming confidence limit algorithm

1. Select an observable subset of M containing n measurements. This is the minimal measurement set and is denoted by M'. Order M with the elements of M' appearing first.
2. Re-order $\mathbf{dz}^{l}$ and $\mathbf{dz}^{u}$ according to the new ordering of M. Assemble $(\mathbf{dz^n})^{l}$, $(\mathbf{dz^n})^{u}$, $(\mathbf{dz^{m-n}})^{l}$, $(\mathbf{dz^{m-n}})^{u}$, J^{n} and J^{m-n}.
3. Factorise J^{n} and calculate $J^{m-n}(J^{n})^{-1}$.
4. For each variable, i=1,...,n, calculate $\mathbf{a^i}$, the i^{th} row of $(J^{n})^{-1}$ and carry out the maximisation in (33) using a linear programming method. The resultant value of $\mathbf{a^i}.\mathbf{dz^n}$ is the i^{th} element of $\mathbf{dx}^{u}$. Similarly, carry out the minimisation in (34), to obtain the i^{th} element of $\mathbf{dx}^{l}$.
5. Add $\mathbf{dx}^{l}$ and $\mathbf{dx}^{u}$ to $\hat{\mathbf{x}}$ to obtain $\mathbf{dx}^{1l}$ and $\mathbf{dx}^{1u}$.

Example 1: In order to illustrate the operation of the linear programming algorithm let us consider a simple system described by the following inequalities

$$-1 < -x_1 + x_2 < 1$$

$$1 < x_1 + x_2 < 3$$

$$0.5 < x_2 < 3$$

These are depicted in Figure 2. Clearly, for the linear system the linearisation error is zero (thus there is no need for dx and dz notation). Consequently it is expected that the linear programming method will calculate the bounding box on the state uncertainty set, $X^{*}(M, \mathbf{z^o})$, that is identical to the one produced by the Monte Carlo simulations (given a sufficiently large number of Monte Carlo iterations).

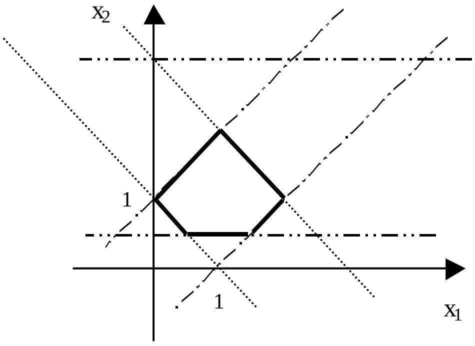


Figure 2. State uncertainty set defined by three double inequalities

We select the first two inequalities as the minimum measurement set, thus defining $(\mathbf{z^n})^l$, $(\mathbf{z^n})^u$, $(\mathbf{z^{m\text{-}n}})^l$, $(\mathbf{z^{m\text{-}n}})^u$, J^{n} and $J^{\mathrm{m\text{-}n}}$.

$$(\mathbf{z^n})^l = [-1\ 1]^{\mathrm{T}},\ (\mathbf{z^n})^u = [1\ 3]^{\mathrm{T}},\ (\mathbf{z^{m\text{-}n}})^l = [0.5]^{\mathrm{T}},\ (\mathbf{z^{m\text{-}n}})^u = [3]^{\mathrm{T}},$$

$$J^{\mathrm{n}} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \text{ and } J^{\mathrm{m\text{-}n}} = [0\ 1]$$

So, the corresponding martices $(J^{\mathrm{n}})^{-1}$ and $J^{\mathrm{m\text{-}n}}(J^{\mathrm{n}})^{-1}$ are

$$(J^{\mathrm{n}})^{-1} = \begin{bmatrix} -0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix} \quad \text{and} \quad J^{\mathrm{m\text{-}n}}(J^{\mathrm{n}})^{-1} = [0.5\ 0.5]$$

The cost function, $\mathbf{a^i}.\mathbf{z^n}$, for i=1 is $\mathbf{a^1}.\mathbf{z^n} = -0.5z_1^{\mathrm{n}} + 0.5z_2^{\mathrm{n}}$, so minimising it with respect of $\mathbf{z^n}$, subject to constraints (33), gives

$$(z_1^{\mathrm{n}})^l = 1 \text{ and } (z_2^{\mathrm{n}})^l = 1$$

and maximising the cost function gives

$$(z_1^{\mathrm{n}})^u = -1 \text{ and } (z_2^{\mathrm{n}})^u = 3$$

For i=2 we minimise $\mathbf{a^2}.\mathbf{z^n} = 0.5z_1^{\mathrm{n}} + 0.5z_2^{\mathrm{n}}$, to obtain

$$(z_1^{\mathrm{n}})^l = -1+\mathrm{s} \text{ and } (z_2^{\mathrm{n}})^l = 2-\mathrm{s}$$

(where the parameter s, $0 \le s \le 1$, signifies the degenerate LP case)

The maximisation of $\mathbf{a^2}.\mathbf{z^n}$ gives

$$(z_1^{\mathrm{n}})^u = 1 \text{ and } (z_2^{\mathrm{n}})^u = 3$$

Evaluating now the equations $x_i^l = \mathbf{a^i}.(\mathbf{z^n})^l$, and $x_i^u = \mathbf{a^i}.(\mathbf{z^n})^u$, for i=1 and i=2 we have $x_1^l = [-0.5\ 0.5][1\ 1]^{\mathrm{T}} = 0$, $x_1^u = [-0.5\ 0.5][-1\ 3]^{\mathrm{T}} = 2$, $x_2^l = [0.5\ 0.5][-1+s\ 2-s]^{\mathrm{T}} = 0$ and $x_2^u = [0.5\ 0.5][1\ 3]^{\mathrm{T}} = 2$, which is the excact solution: $0 \le x_1 \le 2$ and $0.5 \le x_2 \le 2$.

2.4 Sensitivity matrix method

A large class of applications, as exemplified by on-line decision support, requires that the confidence limit analysis procedure calculates uncertainty bounds in real-time. The linear programming method described in the previous section, while being efficient, may fall short of the real-time requirement for large-scale applications. In this section an alternative method, referred to as sensitivity matrix method, is considered.

When the measurement set is minimal (ie it is observable and contains no observable subset), the linearised uncertainty bounds can be calculated without recourse to a linear programming procedure, as discussed in case (ii) in section 2.2. So in these circumstances, confidence limit analysis can be carried out much more rapidly than in the general case when the linear programming algorithm of the previous section has to be used. The shortcut can be taken (when M is minimal) because the Jacobian matrix J is square and invertible. So, any $\mathbf{dx} \in DX^1(M, \mathbf{z^o})$ is given by $J^{-1}.\mathbf{dz}$ for some $\mathbf{dz} \in DZ^1(M, \mathbf{z^o})$. In general, M is over-determined and so J is an m by n matrix of rank n. When J is of this form it has no inverse. The lack of inverse is not due to a shortage of information, rather, there is a surfeit of measurement data. As there is no proper inverse for J, a pseudo-inverse must be used. Let $\mathbf{dx} \in \mathbf{R^n}$ and $\mathbf{dz} \in \mathbf{R^m}$, for which $J.\mathbf{dx}=\mathbf{dz}$, then $J^TJ.\mathbf{dx}=J^T.\mathbf{dz}$ and so

$$\mathbf{dx} = (J^TJ\,)^{-1}J^T.\mathbf{dz} \tag{35}$$

This equation is well defined because when J is of rank n, J^TJ is both square and invertible. The matrix $(J^TJ\,)^{-1}J^T$ is the pseudo-inverse that will be used when $(J\,)^{-1}$ is not well defined. In fact, when $(J\,)^{-1}$ is well defined $(J^TJ\,)^{-1}J^T$ is equal to J^{-1}. $(J^TJ\,)^{-1}J^T$ is referred to as the **sensitivity matrix** as its $(i,j)^{th}$ element relates the sensitivity of the i^{th} element of the state vector to changes in the j^{th} element of the measurement vector.

A new approximate linearised state uncertainty set, $X^2(M, \mathbf{z^o})$, can be defined as follows:

$$X^2(M, \mathbf{z^o}):=\{\mathbf{x} \in \mathbf{R^n}\mathbf{:}\ \mathbf{x}= \hat{\mathbf{x}} +\mathbf{dx},\ \mathbf{dx}=(J^TJ\,)^{-1}J^T.\mathbf{dz},\ \mathbf{dz} \in DZ(M, \mathbf{z^o})\} \tag{36}$$

This set is an approximation of the linearised state uncertainty set, $X^1(M, \mathbf{z^o})$. The following lemma highlights the relationship between $X^1(M, \mathbf{z^o})$ and $X^2(M, \mathbf{z^o})$.

> **Lemma 5** For $X^2(M, \mathbf{z^o})$ defined as in (36) and $X^1(M, \mathbf{z^o})$, defined as in (13),
>
> $$X^1(M, \mathbf{z^o}) \subseteq X^2(M, \mathbf{z^o}) \tag{37}$$
>
> and when M is a minimal observable measurement set
>
> $$X^1(M, \mathbf{z^o}) = X^2(M, \mathbf{z^o}) \tag{38}$$

Proof Let $\mathbf{dx}+\hat{\mathbf{x}} \in X^1(M, \mathbf{z^o})$, then $J.\mathbf{dx} \in DZ(M, \mathbf{z^o})$ by (14) and (18). Put $\mathbf{dz}=J.\mathbf{dx}$. The element of $X^2(M, \mathbf{z^o})$ generated by $\mathbf{dz}$ is just a vector $\mathbf{x}=\hat{\mathbf{x}}+(J^TJ)^{-1}J^T.\mathbf{dz} = \hat{\mathbf{x}}+(J^TJ)^{-1}J^T J.\mathbf{dx} = \hat{\mathbf{x}}+\mathbf{dx}$. Hence $\hat{\mathbf{x}}+\mathbf{dx} \subseteq X^2(M,\mathbf{z^o})$ and (37) follows. Proof of the second part, (38) follows from the observation that when M is minimal, $(J^TJ)^{-1}J^T$ symplifies to J^{-1}.

When M is over-determined there may be vectors $\mathbf{dz} \in Z(M, \mathbf{z^o})$ that are inconsistent. For such a vector there can be no $\mathbf{dx} \in \mathbf{R^n}$ with $J.\mathbf{dx}=\mathbf{dz}$. So, in particular $J(J^TJ)^{-1}J^T.\mathbf{dz}$ is not equal to $\mathbf{dz}$. It may even be that $J(J^TJ)^{-1}J^T.\mathbf{dz}$ is not contained in $DZ(M, \mathbf{z^o})$. It is these vectors that account for the difference between $X^1(M, \mathbf{z^o})$ and $X^2(M, \mathbf{z^o})$. That is, $\mathbf{dx}+\hat{\mathbf{x}} \in X^2(M, \mathbf{z^o})$-$X^1(M, \mathbf{z^o})$ if and only if the state vector $\mathbf{dx}=(J^TJ)^{-1}J^T.\mathbf{dz}$ for some $\mathbf{dz} \in DZ(M, \mathbf{z^o})$ with $J(J^TJ)^{-1}J^T.\mathbf{dz}$ not a member of $DZ(M, \mathbf{z^o})$. Although $X^2(M, \mathbf{z^o})$ is not identical to $X^1(M, \mathbf{z^o})$ it can be used to form bounds for the linearised state uncertainty set. Although these bounds are less tight than the ones obtained with the linear programming method, at least they do not rule out any feasible state vector from the uncertainty box.

Bounding vectors for $X^2(M, \mathbf{z^o})$, denoted $\mathbf{x}^{2\boldsymbol{l}}$ and $\mathbf{x}^{2\boldsymbol{u}}$, can be defined analogously to $\mathbf{x}^{1\boldsymbol{l}}$ and $\mathbf{x}^{1\boldsymbol{u}}$ for the true linearised state uncertainty set $X^1(M, \mathbf{z^o})$. The following algorithm provides a way of calculating these vectors.

Sensitivity matrix confidence limit algorithm

1. Set $i = 0$.
2. Factorise the matrix J^TJ (This can be done using an augmented matrix formulation so as to preserve the condition number of the matrix J).
3. Set $i = i + 1$.
4. Calculate $\mathbf{b^i}$, the i^{th} row of the sensitivity matrix $(J^TJ)^{-1}J^T$ (This can be done efficiently taking into account the sparsity of J and using the augmented matrix based factorisation of step 2).
5. Put $x^2{}_i{}^u = \mathbf{b^i}.\mathbf{dz}^+ + \hat{x}_i$, where

$$\mathrm{dz}_j{}^+ = \begin{cases} \mathrm{dz}_j{}^u & \text{if } \mathrm{b}_j{}^i > 0.0 \\ \mathrm{dz}_j{}^l & \textit{otherwise} \end{cases} \tag{39}$$

Put $x^2{}_i{}^l = \mathbf{b^i}.\mathbf{dz}^- + \hat{x}_i$, where

$$\mathrm{dz}_j{}^- = \begin{cases} \mathrm{dz}_j{}^l & \text{if } \mathrm{b}_j{}^i > 0.0 \\ \mathrm{dz}_j{}^u & \textit{otherwise} \end{cases} \tag{40}$$

6. If $i < n$, go back to step 3. Otherwise stop.

This algorithm bears striking resemblance to the method of calculating the bounds for $X^1(M, \mathbf{z^o})$ described in special case (ii) in section 2.2. In fact, justification of equations (39) and (40) follows as a corollary to lemma 2. Although, in this case, there is not necessarily symmetry about $\hat{\mathbf{x}}$ as g($\hat{\mathbf{x}}$) may not equal $\mathbf{z^o}$.

Example 2: We apply here the sensitivity matrix confidence limit algorithm to the system of inequalities considered in Example 1.

The pseudoinverse matrix $(J^TJ)^{-1}J^T$ is

$$(J^TJ)^{-1}J^T = \begin{bmatrix} -0.5 & 0.5 & 0 \\ 0.333 & 0.333 & 0.333 \end{bmatrix}$$

so, $\mathbf{b}^1$=[-0.5 0.5 0] and $\mathbf{b}^2$=[0.333 0.333 0.333].

Processing $\mathbf{z}$, according to (39) and (40), using the first row, $\mathbf{b}^1$, we obtain

$\mathbf{z}^-$=[1.0 1.0 0.5]^T and $\mathbf{z}^+$ = [-1.0 3.0 3.0]^T which produces

$$x_1{}^l= \mathbf{b}^1\, \mathbf{z}^- =[\ -0.5\ \ 0.5\ \ 0]\ [1.0\ 1.0\ 0.5]^T = 0$$

$$x_1{}^u= \mathbf{b}^1\, \mathbf{z}^+ =[\ -0.5\ \ 0.5\ \ 0]\ [-1.0\ 3.0\ 3.0]^T = 2$$

and using the second row, $\mathbf{b}^2$, we have $\mathbf{z}^-$ = [-1.0 1.0 0.5]^T and $\mathbf{z}^+$ = [1.0 3.0 3.0]^T which gives

$$x_2{}^l= \mathbf{b}^2\, \mathbf{z}^- =[\ 0.333\ 0.333\ 0.333]\ [-1.0\ 1.0\ 0.5]^T = 0.167$$

$$x_2{}^u= \mathbf{b}^2\, \mathbf{z}^+ =[\ 0.333\ 0.333\ 0.333]\ [1.0\ 3.0\ 3.0]^T = 2.333$$

The bounding box on the state uncertainty set, $0 \le x_1 \le 2$ and $0.167 \le x_2 \le 2.33$, as calculated here, is larger than the one obtained with the Monte Carlo and the linear programming algorithm. The widening of bounds along the $\mathbf{x}_2$ direction is caused by the inherent feature of the pseudoinverse, that of attempting to ballance the sum of distances from $x_i{}^l$ and $x_i{}^u$ to <u>all</u> upper- and lover- bound constraints respectively. By contrast, the linear programming and the Monte Carlo algorithms are concerned only with the 'active' constraints for any given value of the state vector, thus ignoring the redundant constraints.

Interval arithmetic formalism for the sensitivity matrix method

The computation of the bounds on individual state variables, implemented in Step 5 of the above algorithm (equations (39) and (40)), can be expressed using the formalism of interval arithmetic. The idea of defining algebras over the sets of intervals, rather than over the sets of real numbers, dates back to late 50's, [Warmus, 1956] and it was subsequently developed by Moore [Moore, 1966] as a framework for the quantification of the mathematically accurate results using finite precision computations. The idea being that any real number can be represented by a pair of adjacent machine numbers which define an interval containing the original number. As the computations proceed, the upper- and lower- limits of the interval containing the result are evaluated conservatively so as to ensure that all feasible results are included.

For a scalar variable, an interval [v] is defined by its upper and lower limits as $[v^l,v^u]$ and it represents a subset of **R**. The set of all intervals of **R** is denoted here as **IR**, so that $[v]\in \mathbf{IR}$. For an *n*-dimensional vector variable, an interval $[\mathbf{v}]$ is defined as a Cartesian product of intervals of the individual coordinates. That is $[\mathbf{v}]=[v_1^l,v_1^u]\mathrm{x}[v_2^l,v_2^u]\mathrm{x}\ldots\mathrm{x}[v_n^l,v_n^u]=[\mathbf{v}^l,\mathbf{v}^u]$ which defines a hyperbox in the $\mathbf{R^n}$ space. The set of all boxes of $\mathbf{R^n}$ is denoted as $\mathbf{IR^n}$, so that $[\mathbf{v}]\subseteq \mathbf{R^n}$ and $[\mathbf{v}]\in \mathbf{IR^n}$.

Here we will confine ourselves to two basic arithmetical operations on intervals. These are: addition of and multiplication of intervals. The constructive definitions of these operations are as follows:

$$[v] + [w] = [v^l,v^u] + [w^l,w^u] = [v^l+w^l,v^u+w^u] \tag{41}$$

$$[v].[w] = [v^l,v^u].[w^l,w^u] =$$

$$=[\min(v^l.w^l,v^l.w^u,v^u.w^l,v^u.w^u),\ \max(v^l.w^l,v^l.w^u,v^u.w^l,v^u.w^u)] \tag{42}$$

where [v], [w], $[v].[w]\in \mathbf{IR}$ and v^l, v^u, w^l, $w^u\in \mathbf{R}$. The multiplication of an interval, $[v]\in \mathbf{IR}$**,** by a scalar, $a\in \mathbf{R}$, is a special case of (42) with the scalar, a, represented as a point-interval [a,a].

The above formulas can be generalised to vector intervals (hyperboxes) to give

$$[\mathbf{v}] + [\mathbf{w}] = [\mathbf{v}^l,\mathbf{v}^u] + [\mathbf{w}^l,\mathbf{w}^u] = [\mathbf{v}^l+\mathbf{w}^l,\mathbf{v}^u+\mathbf{w}^u]=$$

$$([v_1^l+w_1^l,v_1^u+w_1^u], [v_2^l+w_2^l,v_2^u+w_2^u],\ldots,[v_n^l+w_n^l,v_n^u+w_n^u])^T \tag{43}$$

$$[\mathbf{v}].[\mathbf{w}] = [v_1].[w_1] + [v_2].[w_2] + \ldots + [v_n].[w_n]=$$

$$= [v_1^l,v_1^u].[w_1^l,w_1^u] + [v_2^l,v_2^u].[w_2^l,w_2^u] + \ldots + [v_n^l,v_n^u].[w_n^l,w_n^u] =$$

$$= [\min(v_1^l.w_1^l,v_1^l.w_1^u,v_1^u.w_1^l,v_1^u.w_1^u),\ \max(v_1^l.w_1^l,v_1^l.w_1^u,v_1^u.w_1^l,v_1^u.w_1^u)]\ +$$

$$+\ [\min(v_2^l.w_2^l,v_2^l.w_2^u,v_2^u.w_2^l,v_2^u.w_2^u),\ \max(v_2^l.w_2^l,v_2^l.w_2^u,v_2^u.w_2^l,v_2^u.w_2^u)]\ +\ldots$$

$$+\ [\min(v_n^l.w_n^l,v_n^l.w_n^u,v_n^u.w_n^l,v_n^u.w_n^u),\ \max(v_n^l.w_n^l,v_n^l.w_n^u,v_n^u.w_n^l,v_n^u.w_n^u)]=$$

$$[\sum_{i=1}^{n} \min(v_i^l.w_i^l,v_i^l.w_i^u,v_i^u.w_i^l,v_i^u.w_i^u),\sum_{i=1}^{n} \max(v_i^l.w_i^l,v_i^l.w_i^u,v_i^u.w_i^l,v_i^u.w_i^u)]$$

(44)

where $[\mathbf{v}]$, $[\mathbf{w}]\in \mathbf{IR^n}$, $\mathbf{v}^l,\mathbf{v}^u,\mathbf{w}^l,\mathbf{w}^u \in \mathbf{R^n}$, $[\mathbf{v}].[\mathbf{w}]\in \mathbf{IR}$ and $v_i^l,v_i^u,w_i^l,w_i^u\in \mathbf{R}$.

As before, the multiplication of an interval $[\mathbf{v}] \in \mathbf{IR^n}$, by a vector, $\mathbf{a} \in \mathbf{R^n}$ can be seen as a special case of (44) with the vector, **a**, is represented as a vector point-interval [**a**, **a**].

From the definitions (41)-(44), it is clear that the sensitivity matrix method, implemented through the formulas (39) and (40), can be written using the interval arithmetic formalism as

$$[x_i] = [\mathbf{b}^i] . [\mathbf{dz}] + [\hat{x}_i] \qquad , i=1,...,n \tag{45}$$

where $[\mathbf{b}^i]$ is a vector point-interval formed by the i^{th} row of the sensitivity matrix $(J^T J)^{-1} J^T$ and $[\hat{x}_i]$ is a point-interval $[\hat{x}_i, \hat{x}_i]$.

The hyperbox $[\mathbf{x}]=([x_i])$, i=1,...,n, is therefore identical with the set $X^2(M, \mathbf{z^o})$. This result gives an interesting insight into the interval arithmetic based estimation of confidence limits in the context of linear programming and Monte Carlo methods.

Example 3: Continuing with the system of inequalities considered in Example 1 we can formulate the interval equations for the sensitivity matrix method. The point-interval vectors corresponding to the two rows of the pseudoinverse matrix are

$$\mathbf{b}^1=[[-0.5, -0.5]\ \ [0.5, 0.5]\ \ [0, 0]]$$

$$\mathbf{b}^2=[[0.333, 0.333]\ \ [0.333, 0.333]\ [0.333, 0.333]]$$

and the interval vector **z** is

$$[\mathbf{z}]=[[-1, 1]\ \ [1, 3]\ \ [0.5, 3]]$$

Substituting the above to (45), the intervals for the state variables, x_i can be calculated as follows

$$[x_1] = [\mathbf{b}^1] . [\mathbf{z}] =[-0.5, -0.5][-1, 1] + [0.5, 0.5][1, 3] + [0, 0][0.5, 3]=$$

$$= [-0.5, 0.5] + [0.5, 1.5] + [0, 0] = [0, 2]$$

and

$$[x_2] = [\mathbf{b}^2] . [\mathbf{z}]=[0.333, 0.333][-1, 1]+[0.333, 0.333][1, 3]+[0.333, 0.333][0.5, 3]=$$

$$= [-0.333, 0.333] + [0.333, 1.0] + [0.167, 1.0] = [0.167, 2.333]$$

As expected, the result is identical to that calculated by the sensitivity matrix method (Example 2). However, it must be borne in mind that the mathematical elegance of the interval arithmetic is paid for with the increased computational burden. While the interval arythmetic calculations, listed above, required 48 multiplications, 48 logical operations and 8 additions, those in Example 2 were accomplished with 12 multiplications, 12 logical operations and 8 additions.

2.5 Ellipsoid method

An alternative to confidence limit analysis using sensitivity matrix, which historically preceeded this technique, is a method based on the iterative shrinking ellipsoids. The method has been first reported in (Schweppe, 1964) and has been considered by other researchers eg (Fogel, E., Huang, Y.F., 1982), (Norton, J.P., 1987), (Bargiela, A., 1994). The technique is referred here as **ellipsoid method**.

Mathematically, an ellipsoid, $\mathbf{E^t}$ is a region of space defined as follows:

$$\mathbf{E^t}{:=}\{\mathbf{x}\in \mathbf{R^n}\mathbf{:}\ (\mathbf{x}\text{-}\mathbf{x^t})^T P_t^{-1}(\mathbf{x}\text{-}\mathbf{x^t}) \leq 1.0\} \tag{46}$$

for some $\mathbf{x^t} \in \mathbf{R^n}$ and some symmetric and positive-definite matrix P_t , of dimension n by n. $\mathbf{E^t}$ is therefore, a region of $\mathbf{R^n}$ centered on $\mathbf{x^t}$. The aim of the ellipsoid method is to start with a large ellipsoid (usually a n-dimensional sphere) that contains the whole state uncertainty set, and then to generate a sequence of ellipsoids, decreasing in size, leading to one that fits the state uncertainty set as tightly as possible. Using an ellipsoid as an approximation to the state uncertainty set provides a simple and concise description of what can be a very complicated set. The algorithm itself, has the advantages of being sequential, mathematically and conceptually simple and can be very fast computationally.

We consider here the application of the Schweppe's ellipsoid algorithm to confidence limit analysis. The first point to note is that it is a linear method and so the state uncertainty set to be approximated is the set $X^1(M, \mathbf{z^o})$, for a measurement set M and measurement vector $\mathbf{z^o}$. An ellipsoid that is certain to contain $X^1(M, \mathbf{z^o})$ is used as the starting ellipsoid $\mathbf{E^t}$. This ellipsoid may be the n-dimensional sphere centered at the state estimate $\hat{\mathbf{x}}$ (this is the state estimate generated by $\mathbf{z^o}$) with a suitably large diameter, α. In this case $P_0 = \alpha$.I, where I is the n by n identity matrix. The 'observations' in confidence limit analysis are the linearised measurement constraints provided by (14), which can be re-written as

$$\mathbf{z}^{\boldsymbol{l}} - \mathrm{g}(\hat{\mathbf{x}}) + J.\ \hat{\mathbf{x}} \leq J.\mathbf{dx} \leq \mathbf{z}^{\boldsymbol{u}} - \mathrm{g}(\hat{\mathbf{x}}) + J.\ \hat{\mathbf{x}} \tag{47}$$

for all $\mathbf{x}$ in $X^1(M, \mathbf{z^o})$. In this equation, $\mathbf{z}^{\boldsymbol{l}} - \mathrm{g}(\hat{\mathbf{x}}) + J.\ \hat{\mathbf{x}}$ and $\mathbf{z}^{\boldsymbol{u}} - \mathrm{g}(\hat{\mathbf{x}}) + J.\ \hat{\mathbf{x}}$ are constant vectors and so can be pre-calculated. (47) represents m constraints, bounding $J.\mathbf{dx}$ above and below. Each of these is taken in turn and used to modify the current ellipsoid.

Suppose that the t^{th} constraint is being used to update the $t\text{-}1^{st}$ ellipsoid, $t\in \{2,...,m\}$, and that $\mathbf{E^{t-1}}$ contains $X^1(M, \mathbf{z^o})$. The region $\mathbf{F^t}$ bounded by this constraint also contains the uncertainty set $X^1(M, \mathbf{z^o})$. So $X^1(M, \mathbf{z^o})$ is contained in the intersection of these two regions as shown in Figure 3. A new ellipsoid, $\mathbf{E^t}$, can be produced which contains the intersection of $\mathbf{E^{t-1}}$ and the region $\mathbf{F^t}$ bounded by the constraint's hyperplanes. $\mathbf{E^t}$ is the ellipsoid $\{\mathbf{x}\in \mathbf{R^n}\mathbf{:}\ (\mathbf{x}\text{-}\mathbf{x^t})^T P_t^{-1}(\mathbf{x}\text{-}\mathbf{x^t}) \leq 1.0\}$, where

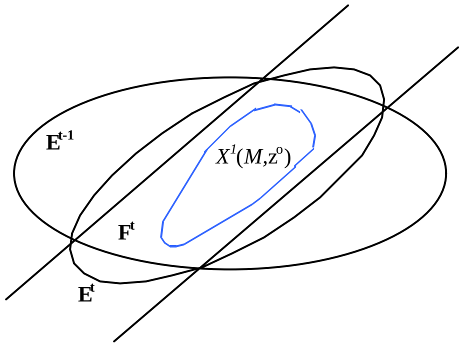
$$\mathbf{x^t} = \mathbf{x^{t-1}} + (\rho_t \nu_t /(e_t^z)^2)\, P'_{t-1}\ .\mathbf{a^t} \tag{48}$$

$$P_t = (1 + \rho_t - (\rho_t \nu_t / ((e_t^{z})^2 + \rho_t\ g_t)))\, P'_{t-1} \tag{49}$$

$$P'_{t-1} = (I + (\rho_t / ((e_t^{z})^2 + \rho_t\ g_t))\, P_{t-1}\ \mathbf{a}^{\mathbf{t}}\ (\mathbf{a}^{\mathbf{t}})^{T})\ P_{t-1} \tag{50}$$

$$g_t = (\mathbf{a}^{\mathbf{t}})^{T}\ P_{t-1}\ \mathbf{a}^{\mathbf{t}}) \tag{51}$$

$$\nu_t = 0.5(\ z_t^{u} + z_t^{l}) - (g(\mathbf{x}^{\mathbf{t-1}}))_t + (J.\,\mathbf{dx})_t + (\mathbf{a}^{\mathbf{t}})^{T}.\mathbf{x}^{\mathbf{t-1}} \tag{52}$$


$\mathbf{E}^{\mathbf{t-1}}$
$X^{1}(M,z^{o})$
$\mathbf{F}^{\mathbf{t}}$
$\mathbf{E}^{\mathbf{t}}$

F

$\mathbf{F}^{\mathbf{t}}$ - the region bounded by the hyperplaines of the constraint

$\mathbf{E}^{\mathbf{t}}$ - the updated ellipsoid

Figure 3 Ellipsoid update (in 2-dimensions)

In these equations, P_{t-1} and $\mathbf{x}^{\mathbf{t-1}}$ are the positive definite martix and centre vector, respectively, for the previous ellipsoid, $\mathbf{E}^{\mathbf{t-1}}$, and ρ_t can be any non-negative real value. The value, e_t^{z} used in these equations, is the t^{th} element of the measurement error vector, $\mathbf{e}^{\mathbf{z}} = \mathbf{z}^{\boldsymbol{u}} - \mathbf{z} = \mathbf{z} - \mathbf{z}^{\boldsymbol{l}} = 0.5(\mathbf{z}^{\boldsymbol{u}} - \mathbf{z}^{\boldsymbol{l}})$.   It should be noted that, despite the fact that (46) refers to P_t^{-1} matrix in its inverted form and (48) to (52) do not, no matrix inversion is involved in the algorithm. In fact, the matrix P_t^{-1} need never be known as all updating is performed using matrices P_{t-1} and P_t.

The parameter ρ_t, which appears in several updating formulae, has not yet been fixed. For any non-negative, real ρ_t, the updated ellipsoid, $\mathbf{E}^{\mathbf{t}}$, will contain the intersection of $\mathbf{E}^{\mathbf{t-1}}$ with the region $\mathbf{F}^{\mathbf{t}}$ bounded by the t^{th} measurement constraints. For $\rho_t = 0.0$, it is easily seen that $\mathbf{E}^{\mathbf{t}}$ is just the same as $\mathbf{E}^{\mathbf{t-1}}$. So this choice of ρ_t leads to no improvement. A more reasonable choice for ρ_t would be the value that minimises the size of $\mathbf{E}^{\mathbf{t}}$ in some way. Such a selection strategy gives the algorithm better convergence properties. In (Fogel, Huang, 1982) there are two suggestions. The first involves the solution of a quadratic equation in ρ_t and produces the ellipsoid of minimum volume. The second requires the solution of a cubic equation and minimises the sum of squares of the semi-axis in $\mathbf{E}^{\mathbf{t}}$.

On termination of the algorithm, the confidence limits for each variable are easily calculated from the final positive-definite matrix, P_t, and the final centre vector $\mathbf{x^t}$. These are:

$$x^{1}{}_{i}^{u} = x_i^t + \sqrt{\ } P_t(i,i) \tag{53}$$

$$x^{1}{}_{i}^{l} = x_i^t - \sqrt{\ } P_t(i,i) \tag{54}$$

Ellipsoidal confidence limit algorithm

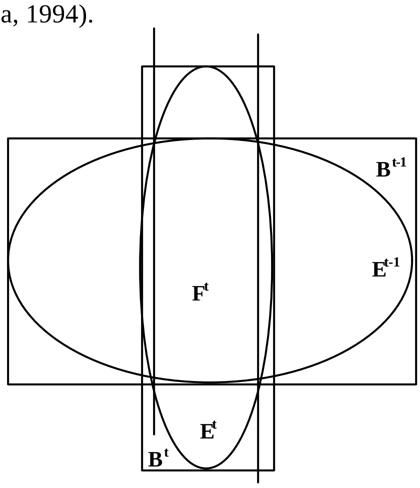
1. Set t=0 and $P_0=\alpha I$.
2. Set $t=t+1$.
3. Calculate g_t and ν_t from equations (51) and (52).
4. Find ρ_t that minimises the volume of the new ellipsoid by solving the following quadratic equation in ρ_t:

   $$(p-1)g_t^2 . \rho_t^2 + ((2p-1).(e_t^z)^2 - g_t + \nu_t)g_t . \rho_t + (e_t^z)^2(p.((e_t^z)^2-\nu_t^2)-g_t) = 0$$

5. Calculate P'_{t-1} from (50).
6. Update the state variable $\mathbf{x^t}$, as per equation (48).
7. Update P_t , equation (49).
8. If not all constraints have been processed yet than repeat from step 2.
9. If the volume of the ellipsoid has been reduced by less than a pre-specified ratio than stop, otherwise reset the constraints counter, t=0, and repeat from step 2.

In some situations, tight bounds can be found by processing each measurement constraint only once, in which case only m steps are required. However, published research suggests that further reduction in the bounds is often possible by re-processing some or all of the constraints (Belforte, Bona, 1985). Also, the variation of the order in which the constraints are processed has the effect on the rate of convergence of the algorithm.

Although the computational complexity of the ellipsoid algorithm compares favourably with other linear confidence limit routines, its effectiveness in bounding the state uncertainty set can be poor. Other approximate methods, that can bound $X^1(M, \mathbf{z^o})$ more effectively with similar computational cost, can be easily derived (Belforte, Bona, 1985), (Mo, Norton, 1988). The reason for a poor performance of the ellipsoid method when the Jacobian, J, of the uncertain system equation, g(.), is sparse is illustrated in Figure 4. Since J is sparse, each measurement constraint only bounds a few of the variables. In the ellipsoid algorithm, constraints are considered individually and so can only improve confidence limits on the few variables that they bound explicitly. Using a 2-dimensional example in which the observation hyperplanes only bound one of the variables (ig, variables are bound in the horizontal direction by the hyperplanes but not in the vertical direction). As a result, the new ellipsoid, , produces tighter confidence limits than in the horizontal direction but looser ones in the vertical direction. When this idea is extended to many dimensions, only a few of which are bound by each constraint, it is easy to see that at each iteration the majority of the variables will have their confidence limits increased and only a minority will have them reduced (Bargiela, 1994).



The new bounds, marked by $\mathbf{B}^t$, are tighter then $\mathbf{B}^{t-1}$ in the horizontal direction but are not as tight in the vertical direction

Figure 4. Ellipsoid update does not always lead to improvement in all variables

Example 4: The system of inequalities introduced in Example 1 is now processed using the ellipsoid algorithm. As in the previous three examples we note that for the linear system the **dx** variable is identical to **x** so the equation (52) simplifies to $\nu_t = 0.5(z_t^{u}+z_t^{l}) + (\mathbf{a}^{\mathbf{t}})^T.\mathbf{x}^{\mathbf{t-1}}$.

The initial ellipsoid is assumed to be a hypersphere of radius $\sqrt{10}$ centred at $\mathbf{x}^0$=[0, 0]. The measurement upper- and lower- bound vectors are $\mathbf{z}^{\boldsymbol{l}}$=[-1 1 0.5]^T and $\mathbf{z}^{\boldsymbol{u}}$=[1 3 3]^T and the matrices J and P$_0$ are

$$J=\begin{bmatrix} -1 & 1 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \text{ and } P_0=\begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

Processing the three constraints in the order t=1,2,3 we obtain, rounded to two places after decimal point

$$g_1=20.00 \quad v_1=0.00 \qquad P_1=\begin{bmatrix}7.26 & 5.09\\ 5.09 & 7.26\end{bmatrix} \qquad \mathbf{x}^{1l}=\begin{bmatrix}-2.69\\ -2.69\end{bmatrix} \quad \mathbf{x}^{1u}=\begin{bmatrix}2.69\\ 2.69\end{bmatrix}$$

$$g_2=24.70 \quad v_2=2.00 \qquad P_2=\begin{bmatrix}2.18 & -0.21\\ -0.21 & 2.18\end{bmatrix} \qquad \mathbf{x}^{2l}=\begin{bmatrix}-0.62\\ -0.62\end{bmatrix} \quad \mathbf{x}^{2u}=\begin{bmatrix}2.33\\ 2.33\end{bmatrix}$$

$$g_3=2.18 \quad v_3=0.89 \qquad P_3=\begin{bmatrix}2.48 & -0.18\\ -0.18 & 2.48\end{bmatrix} \qquad \mathbf{x}^{3l}=\begin{bmatrix}-0.74\\ -0.30\end{bmatrix} \quad \mathbf{x}^{3u}=\begin{bmatrix}2.41\\ 2.44\end{bmatrix}$$

and reversing the order of processing of the constraints to t=3,2,1 gives

$$g_3=10.00 \quad v_3=1.75 \qquad P_3=\begin{bmatrix}12.33 & 0.00\\ 0.00 & 3.12\end{bmatrix} \qquad \mathbf{x}^{3l}=\begin{bmatrix}-3.51\\ -0.46\end{bmatrix} \quad \mathbf{x}^{3u}=\begin{bmatrix}3.51\\ 3.07\end{bmatrix}$$

$$g_2=15.44 \quad v_2=0.69 \qquad P_2=\begin{bmatrix}5.46 & -2.51\\ -2.51 & 3.25\end{bmatrix} \qquad \mathbf{x}^{2l}=\begin{bmatrix}-1.89\\ -0.38\end{bmatrix} \quad \mathbf{x}^{2u}=\begin{bmatrix}2.78\\ 3.22\end{bmatrix}$$

$$g_1=13.73 \quad v_1=-0.97 \qquad P_1=\begin{bmatrix}2.17 & 0.22\\ 0.22 & 1.63\end{bmatrix} \qquad \mathbf{x}^{1l}=\begin{bmatrix}-0.57\\ -0.18\end{bmatrix} \quad \mathbf{x}^{1u}=\begin{bmatrix}2.37\\ 2.37\end{bmatrix}$$

Although both results contain the linearised state uncertainty set $X^1(M, \mathbf{z^o})$, which is defined by

$$\mathbf{x}^l=\begin{bmatrix}0.00\\ 0.17\end{bmatrix} \text{ and } \quad \mathbf{x}^u=\begin{bmatrix}2.00\\ 2.33\end{bmatrix},$$

the quality of bounding the state variables is clearly inferior to that produced by the linear programming and the sensitivity matrix methods.

This is a particularly worrying characteristic of the ellipsoid technique because it is quite difficult to decide which order of processing of the constraints will result in the most rapid convergence of the estimates. It is also worth pointing out that the criterion of non-increasing the volume of the elipse, when progressing from one iteration to the next, does not necessarily imply the tightening of the bounds on all state variables, as can be seen from the transition from t=2 to t=3 in the example above. The consecutive ellipsoids generated when processing the constraints in the order t=1,2,3 and t=3,2,1 are given in Figure 5.

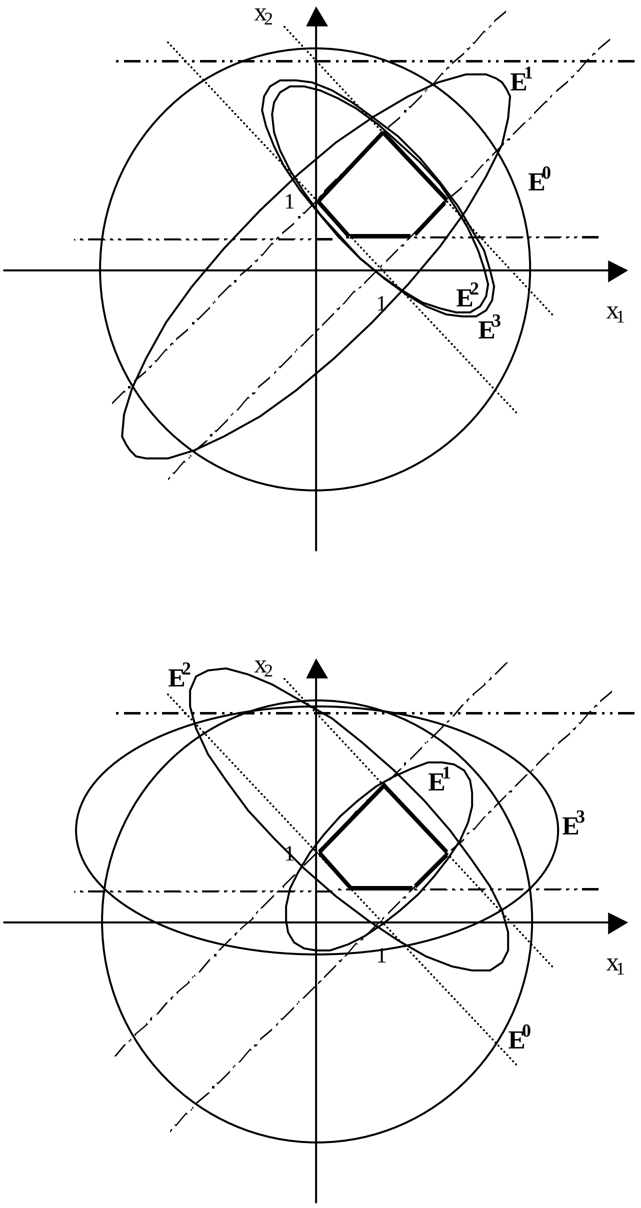


Figure 5. Confidence limits ellipsoids generated by processing the constraints in the order t=1, 2, 3 and t=3, 2, 1 respectively

3 Real-life application

The uncertainty models from the previous section are illustrated here in the context of state estimation of water distribution networks. A network represented diagrammatically in Figure 6 consists of 65 nodes, 92 pipes and 5 inflow points. The inflow points are the reservoirs at nodes 60 and 160, a pumping station at node 68 and two water supplies from a high pressure zone through pressure-reducing valves at nodes 3 and 26. Pipe data: - length [m]; - diameter [m]; and C-values (conductivity); are listed in Table 1. This data, together with the reference pressure measurement in node 160 and five inflow measurements in nodes 3, 26, 60, 68 and 160 allows calculation of the system state (a 70-dimensional vector of 65 nodal pressures and 5 inflows in fixed-pressure nodes).

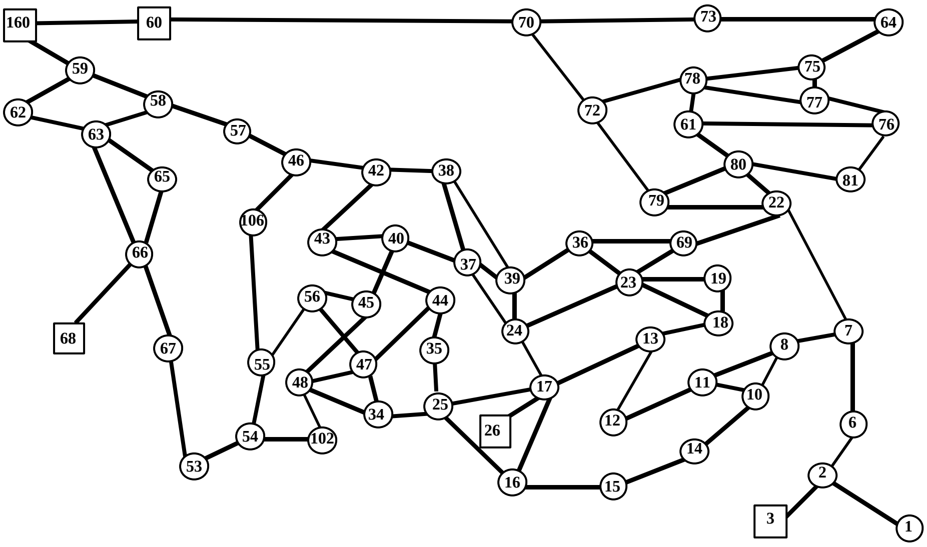


Figure 6. Water distribution network used for testing the confidence limits calculations

Two measurement sets are considered, both for the same operating state. The first measurement set, *M1*, is a minimal measurement set, consisting of nodal consumption values in all but one of the nodes, an inflow measurement for each of the inflow nodes and one reference pressure measurement at node 160. The second measurement set, *M2*, consists of all measurements contained in the set *M1* together with 4 additional pressure and 4 flow measurements.

TABLE 1. Pipe data for the test network

Pipe	*Len.*	*Dia.*	*C*	*Pipe*	*Len.*	*Dia.*	*C*	*Pipe*	*Len.*	*Dia.*	*C*
1-2	800	0.200	140	48-34	900	0.175	80	72-70	1770	0.225	47
2-3	400	0.300	165	48-102	310	0.250	70	70-73	3090	0.356	46
2-6	400	0.300	165	102-54	660	0.225	140	73-64	410	0.225	80
6-7	970	0.300	165	54-53	480	0.225	158	64-75	420	0.225	80
7-8	300	0.225	135	54-55	380	0.225	159	75-78	350	0.094	170
8-10	350	0.225	171	55-56	190	0.225	145	75-77	400	0.150	145
8-11	350	0.125	60	56-45	610	0.125	55	61-76	1470	0.150	81
11-10	360	0.150	105	45-48	1060	0.175	80	70-60	2200	0.356	100
11-12	180	0.125	60	44-43	230	0.250	80	160-59	370	0.381	50
12-13	200	0.175	115	43-40	380	0.250	80	59-58	630	0.300	50
10-14	710	0.225	110	40-45	580	0.168	120	58-57	730	0.300	118
14-15	225	0.225	110	40-37	320	0.168	120	57-46	260	0.300	85
15-16	310	0.225	90	37-24	390	0.200	145	46-42	250	0.300	85
16-17	590	0.094	80	37-39	280	0.168	120	42-38	430	0.300	145
17-13	740	0.175	115	39-24	300	0.150	90	42-43	720	0.250	80
13-18	250	0.225	158	37-38	270	0.200	145	46-106	720	0.200	145
18-19	330	0.300	145	38-39	550	0.300	229	106-55	700	0.142	137
22-7	1510	0.225	96	39-36	210	0.300	127	56-47	550	0.225	145
22-69	120	0.300	145	36-23	180	0.300	112	53-67	220	0.225	160
69-23	420	0.150	90	36-69	147	0.300	145	67-66	270	0.225	160
23-24	300	0.300	145	22-79	160	0.225	80	66-63	800	0.225	110
24-17	650	0.200	158	79-80	340	0.150	90	66-65	210	0.125	60
17-25	330	0.175	127	80-22	390	0.200	145	65-63	590	0.125	60
25-16	630	0.225	104	80-81	1220	0.150	139	63-58	2050	0.150	40
26-17	360	0.300	80	81-76	600	0.150	145	63-62	770	0.225	110
25-34	780	0.175	80	76-77	670	0.150	116	62-59	350	0.225	110
25-35	320	0.225	119	77-78	150	0.150	116	23-19	430	0.300	145
35-44	710	0.225	90	78-61	460	0.094	170	23-18	760	0.117	60
44-47	520	0.250	70	61-80	530	0.150	145	66-68	440	0.300	170
47-34	610	0.225	145	78-72	1100	0.142	105	60-160	270	0.381	32
47-48	540	0.250	70	72-79	600	0.225	60				

As it is pointed out in Section 1.1, the true measurement vector, $\mathbf{z}^{\mathbf{t}}$, rarely coincides with the observed one, $\mathbf{z}^{\mathbf{o}}$. The discrepancy is caused by meter noise affecting real measurements and the inaccuracy of estimates, which are used as pseudomeasurements. In order to reflect this reality the observed measurement values, $\mathbf{z}^{\mathbf{o}}$, are generated in the following way. Firstly, a true operating state, $\mathbf{x}^{\mathbf{t}}$, listed in column 2 of Table 3, is assumed and a true measurement vector, $\mathbf{z}^{\mathbf{t}}$, is calculated as $g(\mathbf{x}^{\mathbf{t}})$. The true measurement values, $\mathbf{z}^{\mathbf{t}}$, are listed in column 2 of Table 2. The observed measurement values, $\mathbf{z}^{\mathbf{o}}$, listed in column 3, are selected randomly from within the range [$\mathbf{z}^{\mathbf{l}}$, $\mathbf{z}^{\mathbf{u}}$], in accordance with (2). The bounds [$\mathbf{z}^{\mathbf{l}}$,$\mathbf{z}^{\mathbf{u}}$] were defined in terms of relative variability of $\mathbf{z}^{\mathbf{t}}$ as follows: 50% for $\mathbf{z}^{\mathbf{t}}$ <0.5 *l/s*; 40% for 0.5< $\mathbf{z}^{\mathbf{t}}$ <1.0 *l/s*; 30% for 1.0< $\mathbf{z}^{\mathbf{t}}$ <5.0 *l/s*; 20% for 5.0< $\mathbf{z}^{\mathbf{t}}$ <10.0 *l/s* and 10% for $\mathbf{z}^{\mathbf{t}}$ >10.0 *l/s*. This corresponds to real-life situation where measurement values are not exact but are contained within the range specified by the accuracy of meters. The state vector, $\hat{\mathbf{x}}$, calculated from the observed measurement vector, $\mathbf{z}^{\mathbf{o}}$, is shown in column 3 of Table 3. The difference between this state estimate and the true state should be noted. It is caused solely by the addition of the simulated measurement errors and shows how noise corrupted measurement data affects system state.

TABLE 2. Measurement data

Node	***True***	***Observed***	***Node***	***True***	***Observed***	***Node***	***True***	***Observed***
Reference pressure [m]			25	2.71	2.21	67	1.87	2.45
160	144.77	144.75	61	0.42	0.22	69	4.52	6.21
			34	7.40	5.96	70	2.18	1.74
Inflows [l/s]			35	2.58	2.48	72	11.51	11.70
26	65.00	65.54	36	1.92	2.44	73	2.77	2.47
3	31.00	31.43	37	2.98	2.20	75	1.32	1.39
60	34.00	33.46	38	2.36	1.64	76	5.37	4.83
160	45.00	45.85	39	0.65	1.00	77	1.16	0.93
68	31.00	30.93	40	6.77	7.48	78	1.35	1.27
			102	2.13	2.14	79	1.91	2.52
Consumptions [l/s]			42	8.03	7.68	80	2.64	2.56
1	4.85	5.00	43	3.51	4.56	81	2.79	2.21
2	6.77	5.65	44	1.89	1.97	106	1.74	2.55
6	2.09	1.94	45	1.10	1.62	26	0.26	0.31
7	1.64	0.98	46	2.73	3.19	3	7.95	8.91
8	1.16	0.78	47	10.80	12.82	60	0.58	0.79
10	9.64	8.56	48	2.95	2.47	68	2.46	2.55
11	0.34	0.29	53	0.67	0.84			
12	0.35	0.53	54	4.54	4.79	*M2 – pressure measurements*		
13	0.50	0.79	55	10.83	9.47	7	140.08	140.08
14	6.54	5.38	56	0.78	0.85	44	139.85	139.85
15	3.18	4.40	57	0.16	0.12	66	141.42	141.42
16	2.01	1.46	58	5.68	4.38	80	140.10	140.10
17	8.51	10.27	59	2.88	2.92			
18	8.71	9.69	62	2.94	3.66	*M2 - flow measurements*		
19	0.00	0.00	63	10.46	10.54	22-69	-7.13	-7.13
22	2.35	2.36	64	3.75	2.99	42-38	-0.16	-0.16
23	0.47	0.49	65	3.84	3.99	7-22	1.94	1.94
24	1.73	2.01	66	2.15	2.76	56-45	0.30	0.30

TABLE 3. True and estimated state vector

***State*Node**	***TrueState***	***Estimate***	***State*Node**	***TrueState***	***Estimate***	***State*Node**	***TrueState***	***Estimate***
(1)-1	140.11	140.04	*(25)*-38	140.33	140.15	*(49)*-69	140.31	140.14
(2)-2	140.23	140.17	*(26)*-39	140.33	140.15	*(50)*-70	143.88	143.90
(3)-6	140.20	140.14	*(27)*-40	140.06	139.84	*(51)*-72	140.25	140.10
(4)-7	140.15	140.08	*(28)*-102	140.07	139.86	*(52)*-73	141.78	141.87
(5)-8	140.02	139.96	*(29)*-42	140.33	140.15	*(53)*-75	140.73	140.76
(6)-10	139.94	139.89	*(30)*-43	140.07	139.85	*(54)*-76	139.97	139.95
(7)-11	140.02	139.95	*(31)*-44	140.06	139.85	*(55)*-77	140.36	140.34
(8)-12	140.38	140.21	*(32)*-45	140.02	139.80	*(56)*-78	140.35	140.32
(9)-13	140.41	140.23	*(33)*-46	140.45	140.27	*(57)*-79	140.27	140.11
(10)-14	139.91	139.84	*(34)*-47	139.96	139.75	*(58)*-80	140.24	140.10
(11)-15	139.93	139.85	*(35)*-48	139.99	139.78	*(59)*-81	139.97	139.95
(12)-16	140.05	139.95	*(36)*-53	140.92	140.65	*(60)*-106	140.34	140.13
(13)-17	141.81	141.58	*(37)*-54	140.21	139.99	*(61)*-26	144.37	144.18
(14)-18	140.36	140.18	*(38)*-55	140.05	139.85	*(62)*-3	140.34	140.28
(15)-19	140.36	140.18	*(39)*-56	140.03	139.83	*(63)*-60	144.82	144.81
(16)-22	140.30	140.13	*(40)*-57	140.71	140.56	*(64)*-160	144.77	144.75
(17)-23	140.36	140.18	*(41)*-58	141.11	141.00	*(65)*-68	141.88	141.59
(18)-24	140.40	140.22	*(42)*-59	143.48	143.39			
(19)-25	140.24	140.09	*(43)*-62	142.84	142.66	***Inflows [l/s]***		
(20)-61	140.23	140.10	*(44)*-63	141.79	141.53	*(66)*-26	65.00	65.54
(21)-34	139.94	139.74	*(45)*-64	141.11	141.20	*(67)*-3	31.00	31.43
(22)-35	140.17	140.00	*(46)*-65	141.32	141.01	*(68)*-60	34.00	33.46
(23)-36	140.33	140.15	*(47)*-66	141.71	141.42	*(69)*-160	45.00	45.85
(24)-37	140.32	140.14	*(48)*-67	141.25	140.97	*(70)*-68	31.00	30.95

The first set of results concerns the state uncertainty sets, $X(M1,\mathbf{z^o})$ and $X^1(M1,\mathbf{z^o})$, for the minimal measurement set, *M1*, as calculated by the Monte Carlo confidence limit algorithm and the linear programming confidence limit algorithm. Results for the sensitivity matrix algorithm, $X^2(M1,\mathbf{z^o})$, are not included because for a minimal measurement set these are identical to those using linear programming algorithm, as explained in Lemma 5.

Rather than trying to visualise the state vectors themselves we focus our attention on their variability, $(x_i^u - x_i^l)/2$, around the average value, $(x_i^u + x_i^l)/2$, for each variable $i \in \{1,...,n\}$. Figure 7 depicts the following state uncertainty variability sets $DX(M1, \mathbf{z^o})$ and $DX^1(M1, \mathbf{z^o})$:

$$DX(M1, \mathbf{z^o}) := \{ \mathbf{dx} \in \mathbf{R^{+\,n}} \mathbf{:} \ \mathbf{dx=|dx|,}\ \hat{\mathbf{x}} + \mathbf{dx} \in X(M1, \mathbf{z^o}) \} \tag{50}$$

$$DX^1(M1, \mathbf{z^o}) := \{ \mathbf{dx} \in \mathbf{R^{+\,n}} \mathbf{:} \ \mathbf{dx=|dx|,}\ \hat{\mathbf{x}} + \mathbf{dx} \in X^1(M1, \mathbf{z^o}) \} \tag{51}$$

The Monte Carlo and linear programming results demonstrate the scale of the potential error in state estimates for a system with no measurement redundancy. Pressure errors are in excess of 2.0 [m] in the region of the network that is most distant from the reference pressure node. In fact, the majority of pressure errors are over 1.0 [m] with only those nodes close to node 160 having relatively tight uncertainty bounds. This indeed confirms the intuitive understanding of the relationship between the uncertainty bounds and the accuracy and location of measurements in the system.

The linear programming results correlate well with the Monte Carlo results; with the linear programming error bounds being no more than 5% off the bounds calculated by the Monte Carlo method for all but the most distant nodes from the reference pressure e.i. 1, 2, 3, 6, 7, 8, 10, 11, 14, 15, 16, 17 and 26, for which the discrepancy is less then 15% (the corresponding state variables are 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 13, 61 and 62). These observations lead to the conclusion that no significant accuracy is lost in linearising the uncertainty model and justify the use of linearised confidence limit algorithms.

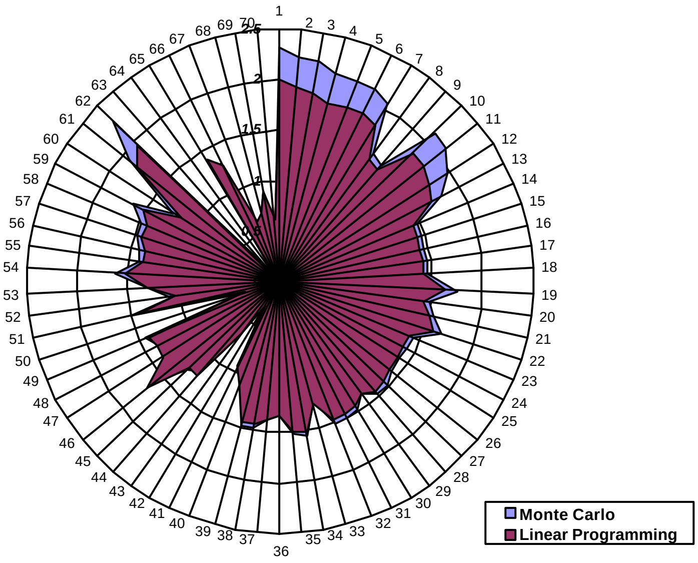
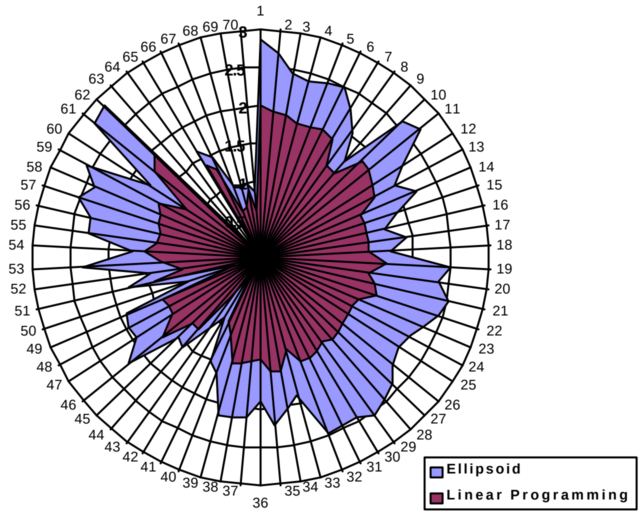


Figure 7. State uncertainty variability sets for Monte Carlo and Linear Programming methods

The results obtained with the Elipsoid method are difficult to assess objectively as they depend on the order of processing of the measurements. Figure 8 illustrates two sets of results obtained by varying the order of processing of the measurements. In either case it is clear that the Elipsoid method produces results that are inferior to those obtained with the linear programming method.



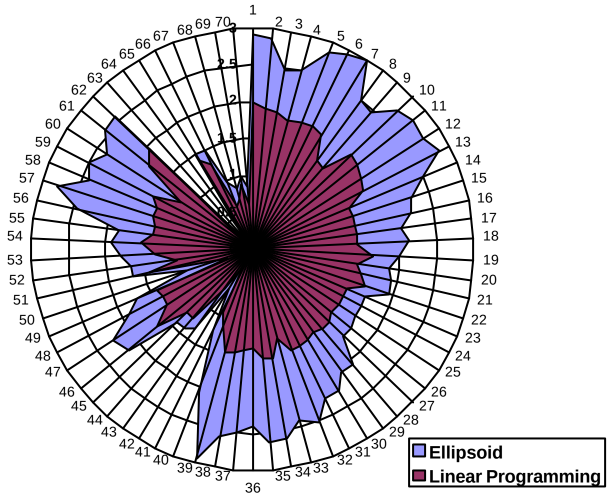


Figure 8. State uncertainty variability sets for two different runs of the Ellipsoid method compared to the result of the Linear Programming method.

The second set of results concerns the state uncertainty sets $X^1(M2,\mathbf{z^o})$ and $X^2(M2,\mathbf{z^o})$ calculated by the linear programming and the sensitivity matrix methods using the augmented measurement set, *M2*. Because of the huge computational requirements of the Monte Carlo method, when the increased number of measurements raises the chances of generating infeasible state vectors, MC method was deemed impractical and was not attempted. As with the previous measurement set, the results are analysed in terms of the state uncertainty variability sets $DX^1(M2,\mathbf{z^o})$ and $DX^2(M2,\mathbf{z^o})$. These are presented in Figure 9.

$$DX^1(M2, \mathbf{z^o}) := \{\, \mathbf{dx} \in \mathbf{R^{+\,n}} : \mathbf{dx}{=}|\mathbf{dx}|,\ \hat{\mathbf{x}} + \mathbf{dx} \in X^1(M2, \mathbf{z^o}) \,\} \tag{52}$$

$$DX^2(M2, \mathbf{z^o}) := \{\, \mathbf{dx} \in \mathbf{R^{+\,n}} : \mathbf{dx}{=}|\mathbf{dx}|,\ \hat{\mathbf{x}} + \mathbf{dx} \in X^2(M2, \mathbf{z^o}) \,\} \tag{53}$$

Since for both methods the state estimate for a given measurement vector, $\mathbf{z^o}$, is $\hat{\mathbf{x}}$, and since $DX^1(M2, \mathbf{z^o}) \subseteq DX^2(M2, \mathbf{z^o})$, the results demonstrate the important result presented in Lemma 5 that $X^1(M2, \mathbf{z^o}) \subseteq X^2(M2, \mathbf{z^o})$.

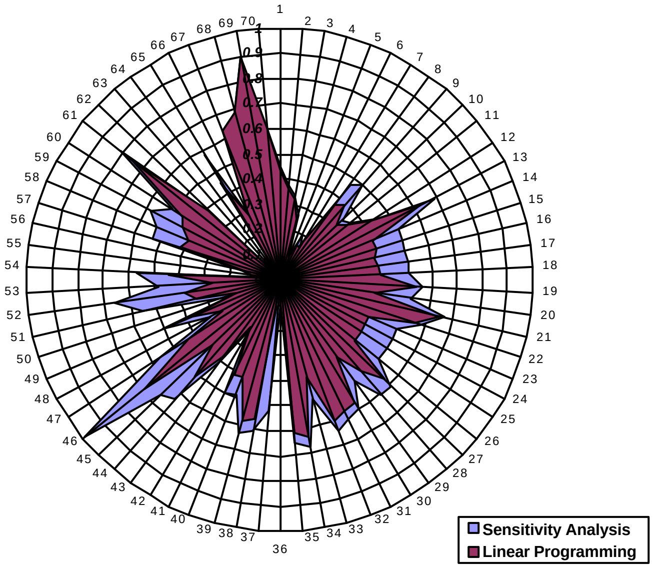


Figure 9. State uncertainty variability sets for Sensitivity Analysis and Linear Programming methods

Considering the individual variables of the state vector it is clear that the linear programming and the sensitivity matrix results are closely correlated and that the overestimation of the state uncertainty variability set with the sensitivity matrix method is confined to approx. 30% of the range of the variables. Again it is interesting to see that the uncertainty bounds have been tightened considerably in the vicinity of the additional meters.

4 Conclusions

State estimation of nonlinear systems is a challenging task that represents a large class of real-life problems. Unfortunately, traditional point-estimate solutions tend to be inadequate because of both, the sensitivity of solutions to the inaccuracy of input data and our inherently approximate knowledge of systems.

It is now widely recognised that, in order to be able to reason about systems' behaviour in presence of uncertainties, it is necessary to abstract from detailed point-estimate solutions to more general, coarser-grain interval-estimates. Such interval-based modelling represent a more credible description of reality, because the predictions or retrodictions based on such models are explicit about the limits of our understanding of systems.

Clearly, while the interval-estimation represents a positive development in terms of building abstract knowledge, the information content of interval-estimates is maximised when the bounds on the individual variables of the state vector (confidence limits) are as tight as possible. We have considered in this chapter one non-linear interval-estimation technique (Monte Carlo) and 3 linear techniques (Linear Programming, Sensitivity Matrix and Ellipsoid) which have been applied to the linearised system model. A rigorous analysis of the effect of the linearisation of the system model on the state uncertainty set has been performed. The analysis shows that the linearisation has only a second-order effect on the shape of the state uncertainty set and, as such, it is an acceptable simplification leading to much more efficient estimation techniques.

Of the three linear interval-estimators the Linear Programming technique produces tightest bounds on the state uncertainty set and is followed closely by the Sensitivity Matrix method. The Ellipsoid method is somewhat disappointing in that it produces a rather conservative bounds and it is sensitive to the order of processing of the constraints. We conclude that the Ellipsoid technique is best suited for rapid, rough approximation of the state uncertaintry set, while the other two techniques are preferable for calculation of tight and consistent bounds on the state uncertainty set. The Sensitivity Matrix method is particularly appealing because of its computational efficiency.

The results demonstrate that the concept of *granular computing* is pertinent to a broad and important class of real life systems. The recognition of information granularity is a recognition of uncertainty in systems modelling and, as such, it can be seen as a progression from *information processing* to *knowledge building*.

ACKNOWLEDGEMENTS. The research reported in this chapter is a result of several research projects funded by the Engineering and Physical Sciences Research Council (EPSRC, UK) and the industry. Recent support from the University of Alberta, Canada, in the form of Visiting Professorship, which enabled compilation of this chapter, is gratefuly acknowledged.

References

Bargiela, A., 1985, An algorithm for observability determination in water systems state estimation, *Proc. IEE*, Vol. 132, Pt. D, No. 6, pp. 245-249.

Bargiela, A., Hainsworth, G.D., 1989, Pressure and flow uncertainty in water systems, *ASCE J. Water Res. Planning and Management,* 115, 2, pp.212-229.

Bargiela, A., 1994, Ellipsoid method for quantifying the uncertainty in water system state estimation, *Proc. IEE Colloquium on Modelling Uncertain Systems*, Vol. 1994/105, pp.10/1-10/3.

Bargiela, A., 1998, Uncertainty – A key to better understanding of systems, *(Plenary lecture) Proc. European Simulation Symposium ESS'98*, ISBN 1-56555-147-8, pp.11-19.

Bargiela, A., 2000, Operational decision support through confidence limit analysis and pattern classification, *(Plenary lecture) Proc. 5th Int. Conf. Computer Simulation and AI,* Mexico City.

Belforte, G., Bona, B., 1985, An improved parameter identification algorithm for signals with unknown-but-bounded errors, *Proc. IFAC/IFORS.,* York.

Cichocki, A., Bargiela, A., 1997, Neural networks for solving linear inequality systems, *Parallel Computing*, Vol. 22, No. 11, pp. 1455-1475.

Fogel, E., Huang, Y.F., 1982, On the value of information system identification – bounded noise case, *Automatica*, Vol. 18, No. 2, pp.

Gabrys, B., Bargiela, A., 1999, Neural networks based decision support in presence of uncertainties, *ASCE J. of Water Res. Planning and Management*, Vol. 125. No. 5, pp.272-280.

Gabrys, B., Bargiela, A., 1999, Analysis of uncertainties in water systems using neural networks, *Measurement and Control*, Vol. 32, No. 5, pp.145-147.

Hainsworth, G.D., 1988, Measurement uncertainty in water distribution telemetry systems, *PhD thesis, The Nottingham Trent University.*

Hartley, J.K., Bargiela, A., 1997, Parallel state estimation with confidence limit analysis, *Parallel Algorithms and Applications*, Vol., 11, No. 1-2, pp.155-167.

Jaulin, L., Walter, E., 1996, *Guaranteed nonlinear set estimation via interval analysis*, in: *Bounding approaches to system identification* (Milanese *et al.*, eds.), Plenum Press.

Mo, S.H., Norton, J.P., 1988, Parameter bounding identification algorithms for bounded-noise records, *Proc. IEE,* Vol. 135, Pt D, No. 2, pp.

Moore, R.E., 1966, *Interval analysis*, Prentice-Hall, Englewood Cliffs, NJ.

Norton, J.P., 1986, *An Introduction to Identification*, Academic Press.

Pedrycz, W., Vukovich, G., !999, Data-based design of fuzzy sets, *Journal of Fuzzy Logic and Intelligent Systems,* Vol. 9, No. 3.

Pedrycz, W., Gomide, F., 1998, *An introduction to Fuzzy Sets. Analysis and Design*, MIT Press, Cambridge, MA.

Pedrycz, W., Smith M.H., Bargiela, A., 2000, A Granular Signature of Data, *Proc. NAFIPS'2000*.

Ratschek, A., Rokne, J., 1988, New computer methods for global optimization, Ellis Horwood Ltd., John Wiley & Sons, New York.

Rokne, J.G., *Interval arythmetic and interval analysis: An introduction*, in: *Granular Computing* (Pedrycz ed.), Elsevier.

Schweppe, F.C., 1973, *Uncertain dynamic systems,* Prentice-Hall, Englewood Cliffs, NJ.

Sterling, M.J.H., Bargiela, A., 1984, Minimum-norm state estimation for computer control of water distribution systems, *Proc. IEE*, Part D, Vol. 131

Warmus, M., 1956, *Calculus of approximations*, Bull. Acad. Polon. Sci., Cl III 4, pp. 253-259.