PROCESS DESIGN AND CONTROL

# A Symbolic Derivation Approach for Redundancy Analysis

**Sebastián J. Ferraro,[†] Ignacio Ponzoni,[‡] Mabel C. Sánchez,[§] and Nélida B. Brignole*[,‡,§]**

*Departamentos de Matemática y de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur (UNS), Avenida Alem 1253, 8000 Bahía Blanca, Argentina, and Planta Piloto de Ingeniería Química (UNS-CONICET), Complejo CRIBABB, km 7 Camino La Carrindanga, 8000 Bahía Blanca, Argentina*

Redundancy analysis is an important stage in plant instrumentation design that consists of classifying the measurements so as to determine the redundant ones. In this work, we propose a new method based on the symbolic derivation of the equations that constitute the nonlinear steady-state model of the plant under study. The strategy succeeds in overcoming the limitations of existing techniques with respect to accuracy as well as model size and complexity. The approach is nonnumeric, thus allowing independence from the operating point. No linearization is required because the method works directly on the original model equations. Another advantage is that there is no need to solve for the observable variables. The prototype was implemented in Matlab, and its performance was assessed for several academic and industrial case studies with accurate results.

## 1. Introduction

The design of a sensor network consists of selecting the most adequate type, quantity, and location of instruments to be incorporated into the system so that the required amount of information about the process can be obtained. Sensor structure affects the estimability of state variables. For example, the subset of unmeasured variables that can be estimated from model equations is determined by the arrangement chosen for the sensors. In turn, the measurements that can be involved in a correction procedure to enhance the precision of the estimation also depend on sensor choice. Consequently, an adequate selection of measurements enables the precise estimation of certain variables even in the event of sensor failures. Thus, it is clear that sensor structure has a significant effect on both the quality and the availability of process knowledge.

The classification of process variables is a major task in the design of sensor structures for complex chemical plants as it helps in the evaluation of whether a given design will provide the required process knowledge in terms of both estimability and precision. According to the feasibility of calculation, the measurements can be classified as redundant or nonredundant by analyzing the mathematical model that represents the plant in steady state. The redundant measurements are those whose values can be computed from model equations and other measured variables. In turn, the unmeasured variables are called observable when they can be evaluated from the available measurements using model equations.

The measured values of the redundant variables can be corrected using data reconciliation procedures. The data compatibility achieved as a result of this adjustment represents a significant improvement in the precision of the values for all of the state variables. Redundancy classifications can be determined through an observability algorithm, which is applied after each individual measurement has been deleted. If the recently incorporated unmeasured variable is observable, then that measurement is redundant. The whole procedure must be repeated for each measured variable to yield the complete classification. Two main research lines have been developed to carry out this task efficiently. In the first one, called the topology-oriented approach, the variables are classified by analyzing the cycles and cutsets that appear in the undirected graph underlying the process topology. The second, known as the equation-oriented approach, makes use of different matrices associated to the system of equations employed to model the process. Two categories can be distinguished in this case: nonstructural and structural techniques. The former applies numerical procedures that make use of the model coefficients, whereas the structural methods consist of rearranging the process' occurrence matrix appropriately.

In general, the nonstructural equation-oriented techniques, as well as the topology-oriented methods, have been specifically designed to model linear and bilinear relationships. Although the procedures are efficient, their application to linearized versions of nonlinear models involves the necessity of initializing the state variables at the plant-design stage. This increases the inherent loss of rigor of the linearization because only very rough initial estimates are normally available at

* To whom all correspondence should be addressed. Phone: 5402914861700. Fax: 5402914861600. E-mail: dybrigno@criba.edu.ar.
[†] Departamento de Matemática, Universidad Nacional del Sur (UNS).
[‡] Departamento de Ciencias e Ingeniería de la Computación, Universidad Nacional del Sur (UNS).
[§] Planta Piloto de Ingeniería Química (UNS-CONICET).

10.1021/ie011046j CCC: $22.00 
Published on Web 10/17/2002

this stage. In contrast, structural algorithms provide a better alternative because they allow more independence from the degree of nonlinearity in the mathematical model.

In this work, we present an effective strategy for performing the classification of measurements that is based on a structural approach. Section 2 contains a critical literature review of the existing redundancy-classification algorithms that points out the need for improvements in redundancy techniques. The fundamentals of our proposal are explained in sections 3 and 4. Simple academic examples that illustrate the procedure are presented in section 5. Next, the performance of the new algorithm is discussed in section 6 in terms of an analysis of the results obtained for several examples of industrial interest. Finally, the main conclusions are summarized in section 7.

## 2. Measurement Classification Strategies

The classification of process variables basically comprises two main stages: the observability analysis and the determination of redundant measurements. Two major schools of thought have been developed to carry out this task, the main contributions of which and ranges of applicability for redundancy categorization are discussed below.

**2.1. Topology-Oriented Approach.** Topology-oriented methods employ the undirected graph G underlying the digraph that represents the process topology. The nodes and edges in G correspond to process units and streams. G also contains an additional node, which represents the environment.

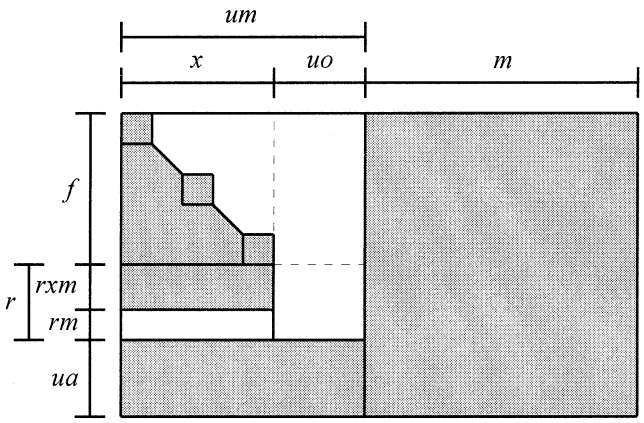
For single-component flow networks, a simple classification procedure[1,2] consists of joining all pairs of nodes connected by means of a stream that contains at least one unmeasured variable. The resulting reduced balance scheme involves only redundant measurements. This technique is adequate only for linear balance relationships. To consider more complex model formulations, it is necessary to perform the redundancy analysis by using specific tools that carefully account for nonlinearities. In this respect, Václavek and Loucka[3] extended the method mentioned above so as to take into account component balance equations, but the methodology considers only streams in which either all or none of the mass fractions are known. Although this algorithm states sufficient conditions for nonredundancy, some nonredundant measurements can be erroneously classified as redundant.[4] Later, Meyer et al.[5] presented an algorithm without assumptions about composition measurements. Although efficient, the formulation considered only component mass balances.

Kretsovalis and Mah[6] incorporated energy balances into the previous model formulations, considering only a univocal relation between the temperature of a stream and its specific enthalpy. They studied process flow sheets made up of reactors, dividers, black boxes (i.e., adiabatic units with no chemical reactions), and units with pure energy flows. The strategy, which consists of two main phases, does not allow for the direct classification of all measurements. First, some measurements are categorized according to theorems based on graph theory. Then, the definition of redundancy is employed together with the observability algorithm for the classification of the remaining measurements. The technique was devised for bilinear models, so it is less rigorous for those pieces of equipment whose functionalities are strongly nonlinear, such as reactors and flashes.

**2.2. Equation-Oriented Approach.** Two main categories can be distinguished within the equation-oriented approach: numerical and structural methods. A pioneering numerical technique is Crowe's matrix projection strategy,[7] which allows the decomposition of the linear balance equations by means of the elimination of the unmeasured variables. This leads to a reduced system that contains only redundant measured variables, because the nonredundant ones automatically disappear. The same concept was later applied to redundancy classification for bilinear systems.[8,9] The procedure requires the specification of numerical values for the measurements. Nonlinear problems can be solved only after linearization around a nominal point,[10,11] which implies that nominal values for all of the variables involved should be provided. Madron[12] also dealt with the matrix of coefficients from the linear/linearized system of equations that constitutes the mathematical model. In this numerical method, the columns of the matrix are permuted so that those associated with the unmeasured variables appear first. As a result, two main blocks can be distinguished. Then, the left block is converted to its canonical form by means of the classic Gauss−Jordan procedure. In the next step, the columns corresponding to the measurements are considered. Elementary operations are performed again, this time to turn the block on the right into its canonical form. After that, the resulting matrix reveals the complete classification. In particular, the measured variables with zero entries in the redundant equations are classified as nonredundant.

For nonlinear systems, an estimation of the nominal operating point is required by numerical strategies. This requirement constitutes an important drawback at the stage of plant instrumentation design, i.e., when no plant data are available. In addition, it is well-known[12] that linearization techniques give rise to further complications that deteriorate the quality of the final results.

Structural techniques are based on rearrangements of the process occurrence matrix. The pioneering method in this field was proposed by Romagnoli and Stephanopoulos.[13] In this technique, the submatrix of unmeasured variables is permuted to block lower-triangular form so that the observable variables can be identified. Then, the unassigned equations are employed to classify the measurements according to the following rules. The redundant variables participate in unassigned equations that contain (1) all measured variables, (2) observable unmeasured variables (except unmeasured composition and temperature for bilinear balances) after they are replaced by their intervals in terms of measurements, and (3) balances around disjoint subsets with all external measured variables. An improved version of this methodology was described by Sánchez et al.[14] The classification of measurements is delayed until the set of redundant equations has been obtained. Symbolic mathematics is employed to express the observable variables in terms of the measurements so that they can be substituted into the redundant equations. Then, all of the variables contained in this reformulated set are redundant. Although the method avoids the erroneous incorporation of some nonredundant measurements into the group of redundant variables, it is only ap-




*um* = *unmeasured variables.*
*m* = *measured variables.*
*x* = *observable variables.*
*uo* = *unobservable variables.*
*f* = *assigned equations.*
*r* = *redundant equations.*
*rxm* = *redundant equations with observable variables.*
*rm* = *redundant equations without observable variables.*
*ua* = *unassigned equations.*

**Figure 1.** Occurrence matrix partitioning resulting from observability analysis.

plicable to simple models that allow for the explicit formulation of the observable variables.

In turn, Joris and Kalitventzeff[15] proposed a different structural rearrangement that is obtained without partitioning the observable variables into blocks. By means of column and row permutations, the measurements necessary to calculate a just computable subsystem of unmeasured variables are identified; those measurements are nonredundant. The main drawback of this method is that it is purely structural, thus being unable to detect algebraic singularities and cancellations that can seriously affect the classification results.

## 3. Problem Statement

In this work, a new structural technique for redundancy analysis, which we call symbolic derivation of implicit functions (SDIF), is presented. The method makes use of the results from the observability analysis. In this paper, in particular, the final pattern resulting from the nonnumeric observability algorithm of Ponzoni et al.[16] provides all of the input information about dependencies among state variables required by SDIF. The matrix rearrangement, originally proposed by Romagnoli and Stephanopoulos,[13] is shown in Figure 1, where the shaded blocks are those that contain nonzero elements. In this pattern, it is clear that the subset of redundant equations **r** comprises some that depend on the observable variables and others that are solely functions of the measurements.

All of these equations constitute the steady-state model chosen to represent the plant under analysis. The system mainly consists of balance equations that describe the conservation of mass and energy as well as thermodynamic relationships that define the significant interrelations between fluid properties. For many problems, complex formulations are required to model with the desired accuracy how the thermodynamic properties are affected by changes in the thermodynamic state of the process. Moreover, the mathematical expressions vary greatly depending on the thermodynamic model chosen to predict these properties. Therefore, for the sake of simplicity, in principle, we chose to provide complete algebraic equations solely for the balance relationships. For the thermodynamic equations, only the dependencies among variables were defined.

The fundamental problem behind redundancy analysis is to determine which measured variables have an algebraic influence on the set of redundant equations. Because the observable variables are functions of the measurements, some terms in the redundant equations might cancel out, sometimes leading to the disappearance of measured variables from the redundant equations. For an equation-oriented classification approach, a straightforward way to deal with the problem consists of solving the assigned equations for the observable variables symbolically and then substituting the observable variables into the redundant equations, canceling terms whenever possible.[17] Nevertheless, this strategy is ineffective for models containing nonlinear equations that cannot be *symbolically* solved for the observable variables. Moreover, as the problem size increases, the amount and complexity of the nesting caused by the successive substitutions becomes prohibitively expensive in terms of both memory storage and run times, thus making it practically impossible to obtain many final symbolic expressions. In view of these limitations, we designed a new approach that significantly reduces the computational burden involved in this task. The strategy proposed in this paper is based on the implicit derivation and symbolic manipulation of model equations. By the precise identification of zero/nonzero partial derivatives, the method succeeds in classifying the measurements.

In this context, detailed thermodynamic formulas are unnecessary for the classification of variables by means of structural techniques because, in principle, no numeric calculations are required. The first stage of our classification procedure, i.e., structural observability analysis, makes use of only the functionalities for the thermodynamic variables involved that express the nonnegligible dependencies between thermodynamic state variables. For example, it is sufficient to indicate that the specific enthalpy of a given component depends only on the temperature, the influence of pressure being negligible, without specifying any analytic equations at all. In relation to the method for redundancy analysis proposed in this paper, these functionalities also provide enough information because it can be assumed without loss of generality that all of the partial derivatives with respect to the state variables involved are always nonzero. In the example given above, the partial derivative of the enthalpy with respect to temperature is regarded as nonzero, whereas the partial derivative with respect to pressure is considered to be zero. In this way, plant model definition is greatly simplified, and it is not necessary to choose a thermodynamic model or to fit experimental data to approximate functions. In addition, the amount of symbolic manipulation involved in the redundancy analysis is significantly reduced without affecting the quality of the classification results. Moreover, these results are independent of the thermodynamic model or individual operating points of the plant under study.

## 4. SDIF: A Derivative-Based Strategy for Redundancy Analysis

Our approach to the problem consists of taking derivatives of the redundant equations with respect to the measured variables, considering the observable variables as functions of the measurements. If a certain derivative is not zero, then the measurement participates in the equation under examination. Otherwise, it does not actually have an influence on this equation, although it might eventually take an active part on some other equation.

**4.1. Fundamentals.** The SDIF redundancy strategy analyzes the following nonlinear system

$$\begin{cases} f_i(x_1,...,x_k,m_1,...,m_l) = 0 & i = 1, ..., k \\ r_j(x_1,...,x_k,m_1,...,m_l) = 0 & j = 1, ..., n \end{cases} \quad (1)$$

where functions $f_1$, ..., $f_k$ correspond to the assigned equations and $r_1$, ..., $r_n$ represent the redundant ones. Variables $x_1$, ..., $x_k$ are observable, whereas $m_1$, ..., $m_l$ are measured.

We will only consider functions that verify the following assumptions: (1) All functions are assumed to be differentiable. (2) If a function is zero on a set with positive measure, then it is identically zero.

These assumptions include virtually every function usually found in practice, for example, all analytic functions.

In system 1, the Jacobian of the subsystem comprising the functions $f_1$, ..., $f_k$ with respect to the variables $x_1$, ..., $x_k$ is nonsingular because this was one of the requirements imposed on the occurrence matrix in the observability analysis stage. More precisely, the determinant of the Jacobian matrix vanishes only on a set with measure zero. Then, by the implicit function theorem, on a neighborhood of each point where the determinant is not zero, each $x_i$ is a function $x_i(m_1,...,m_l)$, $i = 1, ..., k$. Thus, we have

$$f_i(x_1(m_1,...,m_l),...,x_k(m_1,...,m_l),m_1,...,m_l) = 0 \quad \text{for } i = 1, ..., k \quad (2)$$

Let us denote $x = (x_1, ..., x_k)$, $m = (m_1, ..., m_l)$, $x(m) = (x_1(m_1,...,m_l), ..., x_k(m_1,...,m_l))$, $f = (f_1, ..., f_k)$, and $r = (r_1, ..., r_n)$. We define $z(m) = (x(m),m)$.

Then, eq 2 becomes

$$f(x(m),m) = (f \circ z)(m) = 0 \quad (3)$$

This last equation is simply the subsystem of assigned equations regarding $x_1$, ..., $x_k$ as functions instead of independent variables.

Let us introduce the following notation

$$D_x f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_k} \\ \vdots & \cdots & \vdots \\ \frac{\partial f_k}{\partial x_1} & \cdots & \frac{\partial f_k}{\partial x_k} \end{bmatrix},$$

$$D_{m_j} x = \begin{bmatrix} \frac{\partial x_1}{\partial m_j} \\ \vdots \\ \frac{\partial x_k}{\partial m_j} \end{bmatrix}, \quad D_{m_j} f = \begin{bmatrix} \frac{\partial f_1}{\partial m_j} \\ \vdots \\ \frac{\partial f_k}{\partial m_j} \end{bmatrix}, \quad \text{etc.}$$

It is important to note that, in general, the entries in all vector and matrices mentioned in this work are functions, not constants.

Taking derivatives in eq 3 with respect to a measured variable $m_j$, we get, by the chain rule

$$D_x f(z(m)) D_{m_j} x(m) + D_{m_j} f(z(m)) = 0 \quad (4)$$

We solve this linear system to find the vector $D_{mj}x$, because $D_x f$ and $D_{mj}f$ are known. The fact that $D_x f$ is nonsingular guarantees that the system can be solved. Solving the systems for $j = 1, ..., l$, we find $D_{mj}x$ for every $j$. Note that the coefficient matrix $D_x f$ is the same for all of the systems.

Then, we proceed to take derivatives of the functions $r_1 \circ z$, ..., $r_n \circ z$ (which correspond to redundant equations) with respect to the measured variables. The derivative of $r \circ z$ with respect to $m_j$ is the vector

$$D_x r(z(m)) D_{m_j} x(m) + D_{m_j} r(z(m)) = R_j \quad (5)$$

For example, the $i$th component of $R_j$ is

$$R_{j,i} = \frac{\partial r_i}{\partial x_1}\frac{\partial x_1}{\partial m_j} + \frac{\partial r_i}{\partial x_2}\frac{\partial x_2}{\partial m_j} + ... + \frac{\partial r_i}{\partial x_k}\frac{\partial x_k}{\partial m_j} + \frac{\partial r_i}{\partial m_j} \quad (6)$$

If the expression in eq 6 is 0, then $r_i \circ z$ does not really depend on $m_j$. Hence, if the derivatives of all of the functions $r_1 \circ z$, ..., $r_n \circ z$ with respect to $m_j$ are 0, then we classify $m_j$ as a nonredundant variable. If at least one of these derivatives is nonzero, then we classify $m_j$ as a redundant variable. That is, $m_j$ is classified as nonredundant if and only if $R_j$ is the zero vector.

Determining whether the expressions that are the components of $R_j$ are identically zero is nontrivial and is the crucial task in this method for redundancy analysis. The zero-check strategy and algorithm will be explained in sections 4.3 and 4.4, respectively.

This technique can be performed on an individual measured variable $m_j$ as above, and it can also be simultaneously applied to any number of measurements. For example, if we consider all of the measurements, we have to solve the linear system

$$D_x f D_m x + D_m f = 0 \quad (7)$$

to find the matrix $D_m x$ and then compute

$$D_x r D_m x + D_m r = R \quad (8)$$

The $j$th column of the redundancy matrix $R$ is precisely the vector $R_j$ defined above, corresponding to the measured variable $m_j$. Therefore, the entry in row $i$, column $j$ of $R$ is the partial derivative of $r_i \circ z$ with respect to $m_j$.

**4.2. Consistency.** The method presented above treats assigned and redundant equations differently. The observability analysis, however, can determine which equations are to be considered as assigned in a nonunique way. More specifically, for a given plant model, the vector of observable variables is *always the same*, but there might be several possible outputs from the observability analysis with different subsets of assigned and redundant equations. In this section, we show that the redundancy-classification results obtained by the proposed method do not change regardless of the equation assignment. Moreover, the results do not even depend on the observability algorithm chosen, as long

as it satisfies the following conditions: (a) The equations labeled as "assigned" have a structurally nonsingular Jacobian with respect to the observable variables. (b) The nonobservable variables and the corresponding unassigned equations are consistently identified, i.e., they are the same regardless of the algorithm used. This implies that the vector of observable variables and the union of the subsets of assigned and redundant equations are unique.

These conditions are nonrestrictive and are mentioned here only for the purpose of setting the working hypotheses for the proof below. Any observability algorithm that is based on the identification of assigned/redundant/unassigned equations and observable/unobservable variables should comply with these requirements to yield a correct classification of unmeasured variables. In particular, all of the structural observability methods proposed in Ponzoni[18] satisfy these conditions.

*Theorem*. The redundancy classification obtained through SDIF is independent of the choice of the assigned equations and the observability algorithm preceding the analysis under the conditions mentioned above.

*Proof*. Let us consider the nonlinear system

$$g_i(x_1,...,x_k,m_1,...,m_l) = 0 \qquad i = 1, ..., N \tag{9}$$

First, the partition into assigned and redundant equations shown in eqs 1 is obtained using the observability algorithm $O$. Then, the method presented in this work is employed. Because the method can be applied individually to each measured variable, we will consider a certain measurement $m_j$. The linear system

$$D_x f D_{m_j} x + D_{m_j} f = 0 \tag{10}$$

is solved to obtain $D_{mj}x$. The solution exists and is unique because $D_x f$ is nonsingular.

Next, we have to analyze the vector

$$R_j = D_x r D_{m_j} x + D_{m_j} r$$

and determine whether it is zero. Suppose that $m_j$ is nonredundant, that is, $R_j = 0$. Then, $D_{mj}x$ satisfies the system

$$\begin{bmatrix} D_x f \\ D_x r \end{bmatrix} D_{m_j} x + \begin{bmatrix} D_{m_j} f \\ D_{m_j} r \end{bmatrix} = \begin{bmatrix} 0 \\ R_j \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{11}$$

Now suppose that the observability algorithm $O'$ is applied to the original system 9. We obtain another partition into assigned and redundant equations, which will be denoted by $f'$ and $r'$. By condition b, this corresponds to exchanging rows between the upper and lower blocks of eq 11. The right-hand side of eq 11 does not change because it is 0. Then, $D_{mj}x$ also satisfies

$$\begin{bmatrix} D_x f' \\ D_x r' \end{bmatrix} D_{m_j} x + \begin{bmatrix} D_{m_j} f' \\ D_{m_j} r' \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{12}$$

and in particular

$$D_x f' D_{m_j} x + D_{m_j} f' = 0 \tag{13}$$

Equation 13 has one and only one solution because $D_x f'$ is nonsingular, so the solution is the same vector $D_{m_j}x$ satisfying eq 10. Note that this holds because $R_j = 0$; in general, $D_{mj}x$ need not be the same vector as in eq 10.

Then, the lower block of eq 12 is

$$R'_j = D_x r' D_{m_j} x + D_{m_j} r' = 0$$

Therefore, if our technique is employed after the observability algorithm $O'$, then the measured variable $m_j$ is also classified as nonredundant.

It is clear that, reciprocally, if a measured variable $m_j$ is classified as nonredundant after application of observability algorithm $O'$, then it is nonredundant after $O$. Therefore

$$m_j \text{ is nonredundant after } O \Leftrightarrow m_j \text{ is nonredundant after } O'$$

This implies

$$m_j \text{ is redundant after } O \Leftrightarrow m_j \text{ is redundant after } O'$$

**4.3. The Zero-Check Strategy** The methodology for zero checking proposed in this paper is focused on the structure of the algebraic expression. The classification of the measured variable $m_j$ undergoing examination is performed by checking whether expression 6, which is a function $\varphi = \varphi(x,m)$, is identically zero. More specifically, we need to determine whether $\varphi \circ z$ is identically zero, because observable variables are to be considered as functions of the measurements.

This seemingly simple operation presents some major difficulties. The expressions obtained often require algebraic simplifications, and in some cases, the lengths of the formulas exceed the capabilities of the software package used for symbolic manipulations. In addition, the expressions sometimes involve observable variables as well as measured ones, and we do not explicitly know the functions $x_i = x_i(m_1,...,m_l)$, $i = 1, ..., k$, only their derivatives.

On the other hand, the analysis does not require complete detail. We do not need to know an exact, simplified form of the formula considered; we only need to determine whether it is identically zero.

*Definition*. Let $\varphi = \varphi(x,m)$ be an algebraic expression. We will denote by $Z(\varphi)$ the statement "$\varphi \circ z$ is identically zero". Its negation will be indicated by $\sim Z(\varphi)$. The statement $Z(\varphi)$ will also be written as $\varphi \circ z \equiv 0$.

For the sake of simplicity, we shall often write "$\varphi$ is zero", instead of "$\varphi \circ z$ is identically zero". This should cause no confusion.

*Propositions*. If $\varphi$ and $\phi$ are expressions, then the following statements are true:

(i) If for some measured variable $m_j$ we have $\sim Z(\partial(\varphi \circ z)/\partial m_j)$, then $\sim Z(\varphi)$. That is, if any derivative is not zero, considering $x$ as a function of $m$, then the expression $\varphi$ is not zero.

(ii) $Z(\varphi)$ or $Z(\phi)$ implies $Z(\varphi \cdot \phi)$. That is, if a factor is zero, then the product is zero.

(iii) $\sim Z(\varphi)$ and $\sim Z(\phi)$ implies $\sim Z(\varphi \cdot \phi)$.

(iv) If $\sim Z(\phi)$, then $Z(\varphi/\phi) \Leftrightarrow Z(\varphi)$. That is, a fraction is zero if and only if its numerator is zero.

(v) $\sim Z(\exp(\varphi))$.

(vi) If $\varphi$ has only one variable $v$ (observable or measured), then $\sim Z(\mathrm{d}\varphi/\mathrm{d}v) \Rightarrow \sim Z(\varphi)$. That is, if $\varphi$ actually depends on $v$, then $\varphi$ is nonzero.

(vii) If $\varphi$ does not have any observable variables and $\varphi$ is not the constant expression 0, then $\sim Z(\varphi)$.

$$\begin{array}{ccccc} \varphi & \rightarrow & \varphi_{m_1} & \rightarrow & \varphi_{m_1 m_1} \quad \cdots \\ & \searrow & & \searrow & \varphi_{m_1 m_2} \quad \cdots \\ & & \varphi_{m_2} & \rightarrow & \varphi_{m_2 m_1} \quad \cdots \\ & & & \searrow & \varphi_{m_2 m_2} \quad \cdots \end{array}$$

**Figure 2.** Partial derivatives in a tree-like representation.

*Proofs*. (i) Suppose that $Z(\varphi)$ is true. This would mean that $\varphi \circ z$ is identically zero. Then, if $m_j$ is a measured variable, we would have

$$0 \equiv \frac{\partial(\varphi \circ z)}{\partial m_j} = D_x\varphi(z(m))D_{m_j}x(m) + D_{m_j}\varphi(z(m)) = \left(\left(\frac{\partial(\varphi \circ z)}{\partial m_j}\right) \circ z\right)(m)$$

for any vector $m$. This would mean that $Z(\partial(\varphi \circ z)/\partial m_j)$ is true, contradicting the hypothesis. Therefore, our assertion holds.

(ii) Obviously, if $\varphi \circ z \equiv 0$ or $\phi \circ z \equiv 0$, then the product $(\varphi \cdot \phi) \circ z = (\varphi \circ z) \cdot (\varphi \circ z) \equiv 0$. This equality holds because the product of functions is defined pointwise.

(iii) Let Zeros $\varphi \circ z$ be the set where $\varphi \circ z$ vanishes, i.e., Zeros $\varphi \circ z = \{m \in \mathcal{R}^l: \varphi(z(m)) = 0\}$, and let Zeros $\phi \circ z$ be defined similarly. By assumption 2, Zeros $\varphi \circ z$ and Zeros $\phi \circ z$ have measure zero. Thus, the measure of the set Zeros $(\varphi \cdot \phi) \circ z$ = Zeros $(\varphi \circ z) \cdot (\phi \circ z)$ = Zeros $(\varphi \circ z) \cup$ Zeros $(\phi \circ z)$ is zero. However, this implies that $(\varphi \cdot \phi) \circ z$ is not identically zero, that is, $\sim Z(\varphi \cdot \phi)$.

(iv) If $\sim Z(\phi)$, then $\phi \circ z$ is not identically zero, and neither is $1/(\phi \circ z)$. We have $Z(\varphi/\phi)$, which means that $(\varphi/\phi) \circ z = (\varphi \circ z) \cdot 1/(\phi \circ z) \equiv 0$. Because $1/(\phi \circ z)$ is not identically zero, we must have $\varphi \circ z \equiv 0$, that is, $Z(\varphi)$. Otherwise, we would get $\sim Z(\varphi/\phi)$ by proposition iii. Thus, $Z(\varphi) \Rightarrow Z(\varphi \cdot 1/\phi) \Rightarrow Z(\varphi/\phi)$ by proposition ii.

(v) This is obvious because the exponential function never vanishes.

(vi) If $v$ is a measured variable, then proposition i implies our statement directly.

If $v$ is an observable variable, then, by definition, it depends nontrivially on a certain measured variable $m_j$. Thus, $\sim Z(\mathrm{d}\varphi/\mathrm{d}v)$ means that $\mathrm{d}\varphi/\mathrm{d}v \circ z$ is not zero. That is, $\mathrm{d}\varphi/\mathrm{d}v(v(m))$ is not identically zero. Therefore

$$\frac{\partial(\varphi \circ z)}{\partial m_j} = \frac{\partial(\varphi(v(m)))}{\partial m_j} = \frac{\mathrm{d}\varphi}{\mathrm{d}v}(v(m))\,\frac{\partial v}{\partial m_j}(m)$$

which is not identically zero by proposition iii. Then, by i, we obtain $\sim Z(\varphi)$.

(vii) In this case, $\varphi$ is an expression involving only measured variables, if any. All variables are independent, and $\varphi \circ z = \varphi$. Because $\varphi$ is a fully simplified, nonzero expression and no extra cancellations due to observable variables can take place, we conclude that $\varphi \circ z$ is not zero.

Applying proposition i repeatedly, we have, for example

$$\sim Z\left(\frac{\partial^3\varphi}{\partial m_3 \partial m_2 \partial m_1}\right) \Rightarrow \sim Z\left(\frac{\partial^2\varphi}{\partial m_2 \partial m_1}\right) \Rightarrow \sim Z\left(\frac{\partial\varphi}{\partial m_1}\right) \Rightarrow \sim Z(\varphi) \quad (14)$$

and similar statements. We then have the tree sketched in Figure 2, where the "nonzero" property is propagated backward up to the root node: $\sim Z$(some node) $\Rightarrow \sim Z$(all of the parent nodes). In this scheme, subindices mean derivatives. We sketch the tree considering only two measured variables for the sake of simplicity.

**4.4. The Zero-Check Algorithm.** This is a recursive algorithm employed to determine whether a given expression $\varphi$ is identically zero in the sense explained at the beginning of section 4.3. In this context, the partial derivative of each redundant equation with respect to each measured variable is analyzed through this program, which returns the following labels: "Zero" if the expression is identically zero, "Nonzero" if the expression is not zero, and "Unknown" if the algorithm cannot draw a definitive conclusion about the expression.

In brief, the algorithm builds a tree-like search space T whose nodes are algebraic expressions branching from $\varphi$ (root node). As it is built, T is simultaneously explored node by node following a breadth-first search order. This tree has the following two kinds of edges: (a) zero-length edges that connect the expression at each node with its *factors* and (b) unity-length edges that link the expression at each node with its *derivatives* with respect to each measured variable, with the chain rule always being applied to account for the fact that the observable variables depend on the measurements.

Only one kind of edge can branch from each node. First, the algorithm always attempts to generate zero-length edges. If the expression can be factorized, its derivatives are not calculated. Otherwise, unity-length edges are obtained.

As the algorithm builds the tree, it simplifies the expressions using a symbolic manipulation tool. If the expression cannot be handled because of its size or complexity, it is labeled Unknown, and no further analysis is performed down that branch. The branching rules, as well as the rules for individual expression analysis and label information flow propagated upward along the branches, are based on the propositions stated in the previous section.

Each node is first checked to account for cases where an immediate conclusion can be drawn. If the straightforward analysis is inconclusive, new nodes are spawned and recursively analyzed. Further algorithmic details are given in the Appendix.

## 5. Academic Examples

**5.1. The Zero-Check Strategy.** By way of illustration, let us consider how to determine whether a particular algebraic nonlinear expression is identically zero by means of the zero-check algorithm. Consider the expression $\varphi(x_1,m_1,m_2) = (m_1 + x_1)x_1m_2^2 + m_1m_2 + m_1^2$. Also, assume that eq 7 yields $\partial x_1/\partial m_1 = 1$, $\partial x_1/\partial m_2 = 0$. The corresponding search space is shown in Figure 3. Zero-length edges are indicated by dashed arrows, whereas solid arrows represent unity-length edges. Only the significant derivatives are indicated. The expressions enclosed in boxes are those the straightforward analysis would declare Nonzero.

In level 2 of the tree, the expression $4m_2^2 + 2$ is identified as Nonzero. This label is propagated upward by the repeated application of proposition i. Then, the root node $\varphi$ is Nonzero, and the zero-check algorithm ends.

It is interesting to note that there are different ways of drawing the same conclusion. For example, in level 3, the expressions $8m_2$ are Nonzero, and so are the expressions 8 in level 4. Any one of these labels would

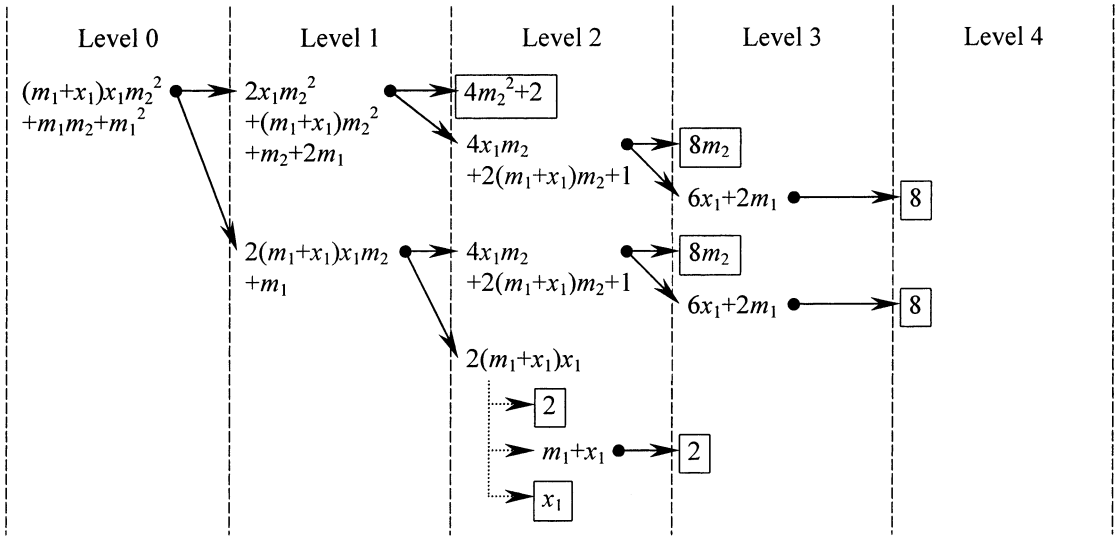



**Figure 3.** Search space T: its significant branches for zero-check purposes.

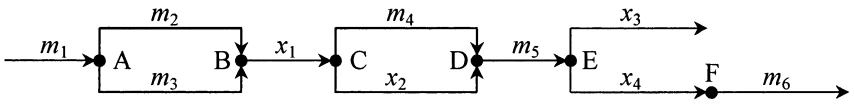



**Figure 4.** Linear example

be propagated upward as before. In level 2, two out of the three factors in $2(m_1 + x_1)x_1$ are Nonzero. The factor $m_1 + x_1$ is seen to be Nonzero by the analysis of its derivative with respect to $m_1$ in level 3. Thus, $2(m_1 + x_1)x_1$ is Nonzero, which implies that the root node is Nonzero. In this context, the zero-check algorithm tends to achieve maximum efficiency by means of a breadth-first search procedure that draws a conclusion at the lowest possible level, thus minimizing the number of derivatives required.

**5.2. General Procedure.** To illustrate the complete method, let us consider the elementary linear system shown in Figure 4, where nodes A−F represent items of equipment. All of the variables are mass flow rates: $m_1$, ..., $m_6$ are measurements, and $x_1$, ..., $x_4$ are observable variables.

The equations of the system are

$$\begin{cases} m_1 - m_2 - m_3 = 0 \ (A) \\ -x_1 + m_2 + m_3 = 0 \ (B) \\ x_1 - x_2 - m_4 = 0 \ (C) \\ x_2 + m_4 - m_5 = 0 \ (D) \\ -x_3 - x_4 + m_5 = 0 \ (E) \\ x_4 - m_6 = 0 \ (F) \end{cases} \tag{15}$$

We could solve for $x_1$, ..., $x_4$ using, for example, the equations for nodes B, C, E, and F, which then become assigned equations. In the notation employed earlier, the left-hand sides correspond to $f_1$, $f_2$, $f_3$, and $f_4$. Then, A and D remain as redundant equations, corresponding to $r_1$ and $r_2$. Thus, eq 7 becomes

$$D_xfD_mx + D_mf = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix} D_mx + \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} = 0$$

which yields

$$D_mx = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Then, we write eq 8 as

$$R = D_xrD_mx + D_mr = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} D_mx + \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 & -1 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 \end{bmatrix}$$

For each entry of $R$, the zero-check analysis is carried out, with obvious results for this example. Because $m_1$, $m_2$, $m_3$, and $m_5$ have nonzero derivatives in at least one equation, they are classified as redundant. Actually, the algorithm stops analyzing each of these measured variables as soon as the first nonzero derivative appears in the corresponding column of $R$, because this is enough to draw a definitive conclusion. In contrast, $m_4$ and $m_6$ have null derivatives in all of the redundant equations and are therefore classified as nonredundant.

It is interesting to note that any other method that examines the occurrence matrix from a purely structural viewpoint, i.e., without considering the actual algebraic form of the equations, would fail in the analysis of this simple example. For instance, let us

consider a method that automatically classifies all of the measured variables that appear in redundant equations as redundant variables. In this example, equations D and A, which contain measurements $m_1-m_5$, are redundant. Then, $m_4$ would be erroneously classified as a redundant variable. The key issue is the fact that one equation contains both direct and indirect dependencies that produce cancellation. By writing down $x_2$ explicitly, one would obtain $x_2 = m_2 + m_3 - m_4$. Then, the substitution of $x_2$ into eq D would lead to the cancellation of $m_4$. Obviously, this effect cannot be predicted just by analyzing the occurrence matrix.

It is clear that the occurrence matrix does not always contain all of the information required for an accurate redundancy analysis. In other words, the correct classification might differ for two systems with the same occurrence matrix. For example, equation D in system 15 could be modified without structural changes by introducing slight alterations that avoid cancellation. For instance, if $m_4$ is multiplied by 2 in equation D, it does not cancel, thus becoming genuinely redundant.

## 6. Prototype Implementation and Industrial Results

A Matlab prototype of the new method was developed to assess the algorithmic performance on realistic problems. The implementation corresponds to the algorithms detailed in the Appendix. The input information was provided by the GS−FLCN observability algorithm by Ponzoni et al.[16] Because this procedure rearranges equations and variables so that a block lower-triangular pattern is generated for the submatrix of size $f \times x$ (see Figure 1), the corresponding Jacobian matrix $D_x f$ keeps the same sparsity pattern. The SDIF implementation takes advantage of this structure when solving eq 4. For the subroutines for symbolic simplification and factorization, built-in procedures from Maple were used.

It is important to note that rigorous models of industrial plants consist of a large number of equations, some of which are strongly nonlinear. The occurrence matrices are always sparse, but the equations are usually coupled, and long dependence chains normally arise. In view of the characteristics of the expressions to be processed, several implementation details should be taken into account. In the first stage of the zero-check algorithm, the nodes are factorized because the factorization reduces the length and complexity of the individual expressions, thus leading to a more efficient analysis. Checking the length of the expressions before processing them with the symbolic manipulation tool is also necessary to avoid the limitations of the symbolic solver.

The algorithmic performance was first tested by analyzing a classical example from Kretsovalis and Mah[6] that consists of 75 equations and 72 variables, involving only linear and bilinear relations. Our classification procedure involves two successive stages: observability and redundancy analyses. All of the final results coincided with those reported by Kretsovalis and Mah[6] (see Table 1), revealing the compatibility of the two approaches.

Next, an industrial example adapted from the plant sector presented by Joris and Kaliventzeff [15] was considered. This problem includes mass and energy balances, as well as strongly nonlinear equations for the two reactors involved in the process. Because of the complexity of the formulation, it was not convenient to choose classical techniques. The application of GS−FLCN's method for structural observability analysis,[16] followed by the redundancy strategy presented in this work, yielded the results shown in Table 1. Our approach provided an efficient way to solve this difficult problem. Therefore, we can infer that it constitutes a promising alternative with a wide applicability range.

**Table 1. Classification Results**

| | equations | | | variables | | | redundant |
|---|---|---|---|---|---|---|---|
| example | a | r | ua | o | uo | m | measurements |
| Mah | 43 | 32 | 0 | 43 | 0 | 29 | 24 |
| Joris | 51 | 27 | 0 | 51 | 0 | 21 | 18 |

## 7. Conclusions

A new approach for redundancy analysis that is based on the symbolic derivation of implicit functions is proposed in this work. The method overcomes deficiencies of existing techniques in terms of applicability range and classification accuracy. Its performance was assessed for a variety of academic and industrial case studies with satisfactory results. One of the main features of the strategy is that it works directly on the original model equations, thus allowing the use of rigorous nonlinear formulations. Another advantage is that, because of its nonnumeric nature, the method does not require the definition of an operating point.

## Nomenclature

$D_x f$ = Jacobian matrix of $f$ with respect to $x$
$f = (f_1, ..., f_k)$ = assigned equations
$k$ = number of assigned equations = number of observable variables
$l$ = number of measured variables
$m = (m_1, ..., m_l)$ = measured variables
$n$ = number of redundant equations
$O$ = observability algorithm
$R$ = redundancy matrix
$R_j$ = $j$th column of $R$
$r = (r_1, ..., r_n)$ = redundant equations
rm = redundant equations without observable variables
rxm = redundant equations with observable variables
ua = unassigned equations
um = unmeasured variables
uo = unobservable variables
$x = (x_1, ..., x_k)$ = observable variables
$Z(\varphi) = \varphi \circ z$ is identically zero
$z(m) = (x(m),m)$
$\varphi$ = algebraic expression
$\sim$ = negation

## Appendix: Zero-Check Algorithms

**1. *Zero Check*($\varphi$, *maxdepth*, *result*).** The purpose of this algorithm is to check whether an expression is identically zero, considering the observable variables as functions of the measurements. The input parameters are $\varphi$, the symbolic expression under study, and *maxdepth*, the maximum number of levels allowed for the breadth-first search procedure, starting from the current node. The output parameter is *result*, a logical variable valued as Zero if $\varphi$ is identically zero, as Nonzero if $\varphi$ is definitely not zero, or as Unknown if no conclusion can be drawn. Finally, the

main internal datum is *depth*, the number of generations of descendants to be explored from the current node.

begin algorithm
*Check Length*($\varphi$, *success*)
if *success* = false then
*result* = Unknown
*Return*
end if
Simplify $\varphi$ using Maple
and assign the simplified expression to *simple*
Extract the numerator
of *simple* using Maple and assign it to *num*
*Perform Straightforward Analysis*(*num*, *result*)
if *result* = Unknown then
for *depth* = 1, *maxdepth*
*Analyze Factors* (*num*, *depth*, *result*)
if *result* = Unknown then
*Analyze Derivatives*(*num*, *depth*, *result*)
if *result* = Nonzero then *Return*
else *Return*
end if
end for
else *Return*
end if
end algorithm

**2. *Check Length*($\varphi$, *success*).** The purpose of this algorithm is to verify that an expression is short enough for Maple to handle. The input parameter is $\varphi$, the symbolic expression under study, and the output parameter is *success*, a logical variable valued as false if $\varphi$ is too long and true otherwise.

**3. *Perform Straightforward Analysis*($\varphi$, *result*).** The purpose of this algorithm is to test whether an immediate conclusion can be drawn for a given expression. The input parameter is $\varphi$, the symbolic expression under study, and the output parameter is *result*, a logical variable valued as Zero if $\varphi$ is identically zero, as Nonzero if $\varphi$ is definitely not zero, or as Unknown if no conclusion can be drawn.

begin algorithm
if $\varphi$ is the number 0
then *result* = Zero
else if ($\varphi$ is the expression $\partial Y/\partial Z$, where $Y$ is a
thermodynamic property and $Z$ is a
state variable)
or (there are no observable variables in $\varphi$)
or (there is only one observable
variable and there are no measured variables in $\varphi$)
or ($\varphi$ is an exponential function)
then *result* = Nonzero
else
*result* = Unknown
end if
end algorithm

**4. *Analyze Factors*($\varphi$, *maxdepth*, *result*).** The purpose of this algorithm is to build zero-length edges to determine whether $\varphi$ is Nonzero through the analysis of its factors. The input parameters are $\varphi$, the symbolic expression under study, and *maxdepth*, the maximum number of levels allowed for the breadth-first search procedure, starting from the current node. The output parameter is *result*, a logical variable valued as Zero if $\varphi$ is identically zero, as Nonzero if $\varphi$ is definitely not zero, or as Unknown if no conclusion can be drawn.

begin algorithm
Factorize $\varphi$ into $\varphi_1$, $\varphi_2$, ..., $\varphi_k$ using Maple
If $k = 1$ then
*result* = Unknown
else
$i = 1$
*result* = Nonzero
while (*result* $\neq$ Zero) and ($i \leq k$) do
*Zero Check*($\varphi_i$, *maxdepth*, *ithresult*)
if *ithresult* = Unknown then *result* = Unknown
else if *ithresult* = Zero then *result* = Zero
end if
$i = i + 1$
end while
end if
end algorithm

**5. *Analyze Derivatives*($\varphi$, *maxdepth*, *result*).** The purpose of this algorithm is to build unity-length edges to determine whether $\varphi$ is nonzero or unknown through the analysis of its derivatives. The input parameters are $\varphi$, the symbolic expression under study, and *maxdepth*, the maximum number of levels allowed for the breadth-first search procedure, starting from the current node. The output parameter is *result*, a logical variable valued as Nonzero if $\varphi$ is definitely not zero or as Unknown if no conclusion can be drawn.

begin algorithm
$i = 1$
*result* = Unknown
while (*result* = Unknown) and ($i \leq$
number of measured variables) do
Calculate the partial derivative of $\varphi$ wrt $m_i$,
using the chain rule, and assign it to $\varphi_{mi}$
*Zero Check*($\varphi_{mi}$, *maxdepth* $-$ 1, *ithresult*)
if *ithresult* = Nonzero then *result* = Nonzero
end while
end algorithm

## Literature Cited

(1) Václavek, V. Studies on System Engineering. III: Optimal Choice of the Balance Measurements in Complicated Chemical Systems. *Chem. Eng. Sci.* **1969**, *24*, 947−955.

(2) Mah, R. S. H.; Stanley, G. M.; Downing, D. M. Reconciliation and Rectification of Process Flow and Inventory Data. *Ind. Eng. Chem. Process Des. Dev.* **1976**, *15*, 175.

(3) Václavek, V.; Loucka, M. Selection of Measurements Necessary to Achieve Multicomponent Mass Balances in Chemical Plants. *Chem. Eng. Sci.* **1976**, *31*, 1199−1205.

(4) Kretsovalis, A.; Mah, R. S. H. Observability and Redundancy Classification in Multicomponent Process Networks. *AIChE J.* **1987**, *33*, 70−82.

(5) Meyer, M.; Koehret, B.; Enjalbert, M. Validation de données sur des systémes incomplétement observés. Récents *Prog.* Génie *Procedés* **1988**, *2*, 6.

(6) Kretsovalis, A.; Mah, R. S. H. Observability and Redundancy Classification in Generalized Process Networks: I. Theorems and II. Algorithms. *Comput. Chem. Eng.* **1988**, *12*, 671−703.

(7) Crowe, C. M.; García Campos, Y. A.; Hrymak, A. Reconciliation of Process Flow Rates by Matrix Projection Part I: Linear Case. *AIChE J.* **1983**, *29*, 881−888.

(8) Crowe, C. M. Reconciliation of Process Flow Rates by Matrix Projection Part II: The Non-Linear Case. *AIChE J.* **1986**, *32*, 616−623.

(9) Sánchez, M. C.; Romagnoli, J. A. Use of Orthogonal Transformations in Classification/Data Reconciliation. *Comput. Chem. Eng.* **1996**, *20*, 483−493.

(10) Swartz, C. L. E. Data Reconciliation for Generalized Flowsheet Applications. Presented at the 197th American Chemical Society National Meeting, Dallas, TX, Apr 9−14, 1989.

(11) Albuquerque, J. S.; Biegler, L. T. Data Reconciliation and Gross Error Detection for Dynamic Systems. *AIChE J.* **1996**, *42*, 2841−2856.

(12) Madron, F. Process Plant Performance. *Measurement and Data Processing for Optimization and Retrofits*; Series in Chemical Engineering; Ellis Horwood: Chichester, U.K., 1992.

(13) Romagnoli, J. A.; Stephanopoulos, G. On the Rectification of Measurement Errors for Complex Chemical Plants. *Chem. Eng. Sci.* **1980**, *35*, 1067−1081.

(14) Sánchez, M. C.; Bandoni, A. J.; Romagnoli, J. A. PLADAT: A Package for Process Variable Classification and Plant Data Reconciliation. *Comput. Chem. Eng.* **1992**, *S*, 499−506.

(15) Joris, P.; Kalitventzeff, B. Process Measurement Analysis and Validation. Presented at the XVIII Congress on the Use of Computers in Chemical Engineering, CEF'87, Taormina, Italy, Apr 26−30, 1987.

(16) Ponzoni, I.; Sánchez, M. C.; Brignole, N. B. A New Structural Algorithm for Observability Classification. *Ind. Eng. Chem. Res.* **1999**, *38*, 3027−3035.

(17) Sánchez, M. C. Monitoreo de Procesos: Análisis de la Instrumentación y Reconciliación de Datos de Planta. Ph.D. Thesis, Universidad Nacional del Sur, Bahía Blanca, Argentina, 1996.

(18) Ponzoni, I. Aplicación de Teoría de Grafos al Desarrollo de Algoritmos para Clasificación de Variables. Ph.D. Thesis, Universidad Nacional del Sur, Bahía Blanca, Argentina, 2001.

*Received for review* December 30, 2001
*Revised manuscript received* July 22, 2002
*Accepted* August 9, 2002

IE011046J