Fault Diagnosis utilizing Structural Analysis

Mattias Krysander, Mattias Nyberg
Department of Electrical Engineering, Linköping University
SE-581 83 Linköping, Sweden
Phone: +46 13 282198, E-mail:matkr@isy.liu.se, matny@isy.liu.se

Abstract— **When designing model-based fault-diagnosis systems, the use of *consistency relations* (also called e.g. *parity relations*) is a common choice. Different subsets are sensitive to different subsets of faults, and thereby isolation can be achieved. This paper presents an algorithm for finding a small set of submodels that can be used to derive consistency relations with highest possible diagnosis capability. The algorithm handles differential algebraic models and is based on graph theoretical reasoning about structure of the model. An important step, towards finding these submodels and therefore also towards finding consistency relations, is to find all *minimal structurally singular* (MSS) sets of equations. These sets characterize the fault diagnosability.**

Keywords— **Fault diagnosis, Structural analysis, Graph theory.**

I. Introduction

When designing model-based fault-diagnosis systems, using the principle of consistency based diagnosis [1], [2], [3], a crucial step is the conflict recognition. As shown in [4], conflict recognition can be achieved by using precomputed consistency relations (also called e.g. *analytical redundancy relations* or *parity relations*). These methods are examples of precompilation techniques [5] With properly chosen consistency relations, different subsets of consistency relations are sensitive to different subsets of faults. In this way isolation between different faults can be achieved.

The systems considered in this paper are assumed to be modeled by a set of nonlinear and linear differential-algebraic equations. To find consistency relations by directly manipulating these equations is a computationally complex task, especially for large and nonlinear systems. To reduce the computational complexity of deriving consistency relations, this paper proposes a two-step approach. In the first step, the system is analyzed structurally to find overdetermined *submodels*. Each of these submodels are then in the second step transformed to consistency relations. The benefit with this two-step approach is that the submodels obtained are typically much smaller than the whole model, and therefore the computational complexity of deriving consistency relations from each submodel is substantially lower compared to directly manipulating the whole model.

The main contribution and the focus of the paper is a structural algorithm for finding these submodels. Instead of directly manipulating the equations themselves, the proposed algorithm only deals with the structural information contained in the model, i.e. which variables that appear in each equation. This structural information is collected in a *structural model*. In addition to finding all submodels that can be used to derive consistency relations, the algorithm also selects a small set of submodels that corresponds to consistency relations with the highest possible diagnosis capability.

In industry, design of diagnosis systems can be very time consuming if done manually. Therefore it is important that methods for diagnosis-system design are as systematic and automatic as possible. The algorithm presented here is fully automatic and has been implemented in Matlab. The input to the algorithm is a structural model of the system. This structural model can in turn easily be derived from for example simulation models.

Structural approaches have also been studied in other works dealing with fault diagnosis. In [6] a structural approach is investigated as an alternative to dependency-recording engines in consistency based diagnosis. Furthermore a structural approach is used in the study of supervision ability in [7] and an extension to this work considering sensor placement is found in [8].

In Sections II and III, structural models and their usefulness in fault diagnosis are discussed. Then in Section IV, a complete description of the algorithm is given. The algorithm is then in Section V applied to a large nonlinear industrial process, a part of a paper mill. In spite of the complexity of this process, a small set of consistency relations with high isolation capability is successfully derived.

II. Structural models

The behavior of a system is described with a model. Usually the model is a set of equations. A structural model [7] contains the information of which variables that are contained in each equation. Let M_{orig} denote the structural model obtained from the equations, describing the system to be diagnosed. This structural model will contain three different kinds of variables: known variables Y, e.g. sensor signals and actuators; unknown variables X_u, for example internal states of the system; and finally the faults F. If faults are decoupled then they will also be included in X_u. The differentiated and non-differentiated version of the same variable are considered to be different variables. The time shifted variables in the time discrete case are also considered to be separate variables.

A structural model can be represented by an *incidence matrix* [9], [10]. The rows correspond to equations and the columns to variables. A cross in position (i, j) tells that variable j is included in equation i.

Example 1: A simple example is a pump, pumping water into the top of a tank. The system is shown in Figure 1.

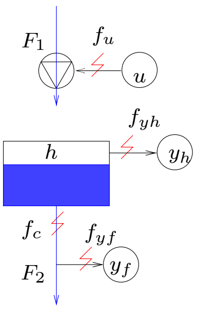


Fig. 1. The system to be diagnosed.

The water flows out of the tank through a pipe connected to the bottom of the tank. The known variables are the pump input u, the measured water level in the tank y_h, and the measured flow from the tank y_f. The fault variables are an arbitrary deviation f_u of the actuator signal, an arbitrary deviation f_{yh} of the water level sensor, a slow varying clogging fault f_c considered to be constant, and a constant bias fault f_{yf} of the water flow sensor. Note that the clogging fault change the dynamics of the system. The actual flows to and from the tank are denoted F_i, and the actual water level in the tank is denoted h. The analytical and structural model M_{orig} of the watertank system is shown in Table I. Equation e_1 describes the pump, e_2 the conservation of volume in the tank, e_3 the water level measurement, e_4 the flow from the tank caused by the gravity, e_5 a fault model for the clogging fault f_c, e_6 the flow measurement, and e_7 a fault model for the flow measurement fault f_{yf}. □

TABLE I
THE ANALYTICAL AND STRUCTURAL MODEL FOR THE WATER TANK SYSTEM SHOWN IN FIGURE 1.

Analytical model: Structural model:

expression		unknown				fault						known		
		F_1	F_2	h	$\dot{h}$	f_u	f_{yh}	f_c	$\dot{f}_c$	f_{yf}	$\dot{f}_{yf}$	u	y_h	y_f
$F_1 - f_u - u = 0$	e_1	X				X						X		
$F_1 - F_2 - \dot{h} = 0$	e_2	X	X		X									
$h + f_{yh} - y_h = 0$	e_3			X			X						X	
$(1 - f_c)\,F_2^2 - h = 0$	e_4		X	X				X						
$\dot{f}_c = 0$	e_5								X					
$F_2 + f_{yf} - y_f = 0$	e_6		X							X				X
$\dot{f}_{yf} = 0$	e_7										X			

III. Fault Diagnosis Using Structural Models

The task is to find submodels that can be used to form consistency relations. To be able to draw a correct conclusion about the isolability from the structural analysis, it is crucial that for each of these submodels there is a consistency relation that validates all equations included in the submodel. The common definition of consistency relation does not ensure this. Therefore a new definition of *consistency relation for an equation set* E is introduced that explicitly points out the submodel considered. Before consistency relation for E is defined some notation is needed.

Let $\mathbf{x}$ and $\mathbf{y}$ denote the vectors of variables contained in X_u and Y respectively. Then $E(\mathbf{x}, \mathbf{y})$ denote an equation set that depends on variables contained in X_u and Y.

Definition 1 (Consistency Relation for E) A scalar equation $c(\mathbf{y}) = \mathbf{0}$ is a *consistency relation for* the equations $E(\mathbf{x}, \mathbf{y})$ iff

$$\exists \mathbf{x} E(\mathbf{x}, \mathbf{y}) \Rightarrow c(\mathbf{y}) = 0 \tag{1}$$

and there is no proper subset of E that has property (1).

Definition 1 differ from the common definition of consistency relation in the way that there is no proper subset of E that has property (1). Refer this as the minimality condition in Definition 1. The next example illustrates Definition 1.

Example 2: Consider the model $E = \{y_1 = x, y_2 = x, y_3 = x\}$. The equation $y_1 - y_2 = 0$ is not a consistency relation for E, because $\{y_1 = x, y_2 = x\}$ also implies $y_1 - y_2 = 0$. Hence $y_1 - y_2 = 0$ is consistency relation for $\{y_1 = x, y_2 = x\}$.

However $y_1 + y_2 - 2y_3 = 0$ and $(y_1 - y_2)^2 + (y_2 - y_3)^2 = 0$ are examples of consistency relations for E.

Finally, is there a consistency relation for $\{y_1 = x\}$? Since $\exists x \; : \; y_1 = x$ is always true any $c(\mathbf{y}) = 0$ that fulfills expression (1) must be an always true expression, e.g. $0 = 0$. However the empty set of equations will also fulfill (1). Hence an always true expression is a consistency relation for the empty set of equations and there does not exist any consistency relation for $\{y_1 = x\}$ at all. □

The minimality condition in Definition 1 is important, because it guarantees that any invalid equation can infer an inconsistency.

A. Basic Assumption

A basic assumption is needed to guarantee that the subsets found only by analyzing structural properties are exactly those subsets that can be used to form consistency relations. Before the basic assumption is presented, some notation is needed. Let E be any set of equations and X any set of variables. Then define $\mathrm{var}_X(E) = \{x \in X | \exists e \in E : e$ contains $x\}$ and $\mathrm{equ}_E(X) = \{e \in E | \exists x \in X : e \text{ contains } x\}$. Also, let $\mathrm{var}_X(e)$ and $\mathrm{equ}_E(x)$ be shorthand notations for $\mathrm{var}_X(\{e\})$ and $\mathrm{equ}_E(\{x\})$ respectively. Finally, the number of elements in any set E is denoted $|E|$.

As mentioned earlier, the structural model contains less information than the analytical model. The next assumption makes it possible to draw conclusions about analytical properties from the structural properties.

Assumption 1: There exists a consistency relation $c(\mathbf{y}) = \mathbf{0}$ for the equation set H iff

$$\forall X' \subseteq \mathrm{var}_{X_u}(H), X' \neq \varnothing : |X'| < |\mathrm{equ}_H(X')| \tag{2}$$

According to Assumption 1 the unknown variables in H can be eliminated if and only if it holds that for each subset of variables in H, the number of variables is less then the number of equations in H which contain some of the variables in the chosen subset.

Assumption 1 are often fulfilled. For example all subsets of equations found in the industrial example in the end of the paper satisfy Assumption 1. Even though the "only if" direction of Assumption 1 is difficult to validate in an application, the results of the paper can still be used to produce a lower bound of the actual isolation capability.

If all subsets of the model fulfill Assumption 1, the structural analysis will find all subsets that can be used to find consistency relations.

B. Finding Consistency Relations via MSS Sets

Now, the task of finding those submodels that can be used to derive consistency relations will be transformed to the task of finding the subsets of equations that have the structural property (2). To do this, two important structural properties are defined [11].

Definition 2 (Structurally Singular) A finite set of equations E is *structurally singular* with respect to the set of variables X if $|E| > |\mathrm{var}_X(E)|$.

Definition 3 (Minimal Structurally Singular) A structurally singular set is a *minimal structurally singular* (MSS) set if none of its proper subsets are structurally singular.

For simplicity, MSS will always mean MSS with respect to X_u in the rest of the paper. The next theorem tells that it is sufficient and necessary to find all MSS sets to get all different sets that can be utilized to form consistency relations. The task of finding all submodels that can be used to derive consistency relations has thereby been transformed to the task of finding all MSS sets.

Theorem 1: Given a model M let $E_i \subseteq H \subseteq M$, where E_i is an MSS set for each i. Further, let H and all E_i fulfill Assumption 1. Then there exists a consistency relation $c(\mathbf{y}) = \mathbf{0}$ for $H(\mathbf{x}, \mathbf{y})$ where $|H| < \infty$ iff $H = \bigcup_i E_i$.
For a proof, see [12].

IV. Algorithm for finding and selecting MSS sets

The objective is to find a small set ω of MSS sets with, of the user defined desired isolability I_{des}. If full isolability is desired the returned set ω has the maximum possible isolation capability.

The algorithm can be summarized in the following steps.

Algorithm 1:

Input: A original structural model and desired isolability I_{des}.

1. Extracting the no-fault model: Neglect all fault variables and all equations containing only fault variables.
2. Differentiating the model: Find equations that are meaningful to differentiate for finding MSS sets.
3. Simplifying the model: Given the extracted model and the additional equations found in step 2, remove all equations that cannot be included in any MSS set. To simplify the next step, merge sets of equations that have to be used together in each MSS set.
4. Finding MSS sets: Search for MSS sets in the simplified model.
5. Analyzing isolability: Examine the isolability of the MSS sets found in step 4.
6. Decoupling faults: If the isolability has to be improved to fulfill the desired isolability, some faults have to be decoupled. For decoupling faults, return to step 2 and consider these faults as unknown variables in X_u.
7. Selecting a subset of MSS sets: Select a small set of MSS sets that contains the highest possible or desired isolability.

Output: A small set of MSS sets and their isolability.

Note that to avoid searching for all MSS sets decoupling all possible faults, Algorithm 1 has been organized so that first, the fault free model is analyzed. Then if it is necessary for achieving higher isolability, faults are decoupled. The following sections discuss each of the steps in Algorithm 1.

A. Extracting the No-Fault Model

To reduce the computational complexity of finding all MSS sets for all choices of decoupling faults the first step extracts the no-fault model M_{ext}, i.e. when no fault variables are decoupled. This model is the simplest because it includes the least number of unknown variables. When the isolability of the no-fault model has been calculated other decoupling alternatives are used only if the desired isolability is not fulfilled. The no-fault model is the subset of equations of M_{orig} that includes either some unknown variables or some known variables.

Example 3: Continuation of Example 1. Since no fault is decoupled in the first iteration equations e_5 and e_7 in Table I are removed. The resulting model is $M_{ext} = \{e_1, e_2, e_3, e_4, e_6\}$. □

B. Differentiating the Model

In this section an algorithm for handling derivatives is defined. This algorithm is referred to as Algorithm 2. First an example will show why differentiation has to be considered.

Example 4: Consider the model M_{ext} in Example 3. This model is a part of the model in Table I. An algorithm that is not capable of differentiating equations can obviously not eliminate $\dot{h}$ in e_2, because there is no other equation including $\dot{h}$. In general, all derivatives of a model M have to be considered. If $M^{(i)}$ denote the set of the i:th time derivative of each element, the equation set generally considered is $\cup_{i=0}^{\infty} M^{(i)}$. □

To summarize the example, Algorithm 2 must be capable of differentiating equations. The next question to answer is if it is possible to predict the structural model of a differentiated analytical model by using only the structural model of the original analytical model? An example is used to answer this question.

Example 5: Consider again the model in Example 3. The differentiated equation $\dot{e}_4$ is $2\,(1 - f_c)\;F_2\,\dot{F}_2 - F_2^2\,\dot{f}_c = \dot{h}$. The variable h is linearly dependent in e_4 and therefore $\dot{h}$ is linearly contained in equation $\dot{e}_4$. Furthermore, both F_2 and $\dot{F}_2$ are nonlinearly contained in $\dot{e}_4$ as a consequence of the fact that F_2 is nonlinearly contained in e_4. □

This example shows that variables are handled in different ways depending on if they are linearly or nonlinearly dependent. To be able to take this different treatment into account, information about which variables that are linearly contained is added to the structural model. With this additional knowledge a structural differentiation can be defined that produce a correct structural representation of differentiated equations. Structural differentiation for an arbitrary variable x and an arbitrary equation e is defined in the following way:

1. If x is linearly contained in e then $\dot{x}$ is linearly contained in $\dot{e}$.
2. If x is nonlinearly contained in e then both x and $\dot{x}$ are nonlinearly contained in $\dot{e}$.

Now structural differentiation can be applied to the structural model. Since all number of differentiations of each equation implies a new equation, there are infinitely many equations in the differentiated model. Let $m(y)$ be a limit for variable $y \in Y$ of the order of derivative that can be considered as possible to estimate. If these limits are introduced and if the structural assumption to be presented next is fulfilled, it is possible to find all MSS sets in a finite subset of the differentiated model.

Before the assumption is presented some notation is needed. If g is any equation, function or variable, let $g^{(i)}$ denote the i:th time derivative of g. Then define $\overline{\text{var}}_X(E) = \{\text{undifferentiated } |\exists i(x^{(i)} \in \text{var}_X(E))\}$, e.g. $\overline{\text{var}}_{X_u \cup Y}(\{y = \dot{x}\}) = \{y, x\}$.

Assumption 2: The model M_{ext} has the property

$$\forall E \subseteq M_{ext} : |E| \leq |\overline{\text{var}}_{X_u \cup Y}(E)|. \tag{3}$$

The meaning of condition (3) is that each subset of equations include more or equally many different variables, considering derivatives as the same variable. If Assumption 2 is not fulfilled and there are no redundant equations, the model would normally be inconsistent.

To present a method to verify that a model fulfills Assumption 2 some notation is needed. Let E be a set of equations, X a set of variables, and Γ a set of edges. Define Γ such that there is an edge in Γ between $x \in X$ and $e \in E$ if and only if $x \in \text{var}_X(e)$. Then the *bipartite graph* "see Appendix" (E, X, Γ) is denoted $\mathcal{G}(E, X)$. The easiest way to verify Assumption 2 is to verify the equivalent statement that there is a *complete matching* "see Appendix" of M_{ext} into $\overline{\text{var}}_{X_u \cup Y}(M_{ext})$ in the corresponding *bipartite graph* $\mathcal{G}(M_{ext}, \overline{\text{var}}_{X_u \cup Y}(M_{ext}))$.

A sufficient condition that there is a finite submodel that contains all MSS sets is that the model M_{ext} satisfy Assumption 2 and all known variables have finite limitations.

Algorithm 2 is a greatly influenced by Pantelides' algorithm [11]. Before the algorithm is presented, a few definitions are introduced. Let $M_\alpha = \bigcup_{i=1}^{n} \bigcup_{j=1}^{\alpha_i} \{e_i^{(j)}\}$ be a differentiated model of $M_{orig} = \bigcup_{i=1}^{n} \{e_i\}$. Then the highest number of differentiations in M of equation i is α_i. Let $M^{max} = \{e_i^{(\alpha_i)} | 1 \leq i \leq n\}$ be the set of *most differentiated equations* in M and $M_\infty = \{e_i^{(j)} | e_i \in M, j \in \mathbb{N}\}$. The *highest derivative* of a non-differentiated variable x in a model M is denoted $\beta(M, x)$, i.e. $\beta(M, x) = \max(\{i | x^{(i)} \in \text{var}_{X_u}(M)\})$. Finally let $\widehat{\text{var}}(M)$ be the variables $\text{var}_{X_u \cup Y}(M)$ that fulfill the following two requirements:

- It is the highest derivative of each variable that are considered.
- It is the variables, whose derivative is unknown.

For example, if $\dot{y} \in \text{var}_{X_u \cup Y}(M)$, $\forall i \in \mathbb{Z}_+ \backslash \{1\} : y^{(i)} \notin \text{var}_{X_u \cup Y}(M)$, and $m(y) = 1$, then $\dot{y} \in \widehat{\text{var}}(M)$ because $\dot{y}$ is the highest derivative of y in M and $\ddot{y}$ is unknown.

Algorithm 2:

Input: The original model M_{ext}, a description of which variables that are linearly contained in each equation, and for each $y \in \overline{\text{var}}_Y(M_{orig})$, $m(y) < \infty$.

1. Let the current model M_c be M_{orig} and let $i = 1$.
2. If $i \leq |M_{orig}|$ then let M_c^{max} be only the most differentiated equations of M_c. Let $M_c^{max}(i)$ denote the i first equations in M_c^{max} and let equation i in M_c^{max} be denoted e_i. A complete matching of $M_c^{max}(i-1)$ into $\widehat{\text{var}}(M_c^{max})$ in the bipartite graph $\mathcal{G}(M_c^{max}, \widehat{\text{var}}(M_c^{max}))$ is found in previous steps. Search for an *augmented path* "see Appendix" in $\mathcal{G}(M_c^{max}, \widehat{\text{var}}(M_c^{max}))$ from e_i to an unassigned variable in $\widehat{\text{var}}(M_c^{max})$.

a) If an augmented path is found then switch matching and nonmatching edges in the path. The new matching edges together with the previous matching now forms a complete matching of $M_c^{max}(i)$ into $\widehat{\text{var}}(M_c^{max})$. Set $i = i + 1$ and goto step 2.

b) No augmented path is found. Then an MSS set with respect to $\widehat{\text{var}}(M_c^{max})$ is found as follows. Let all nonmatching edges be directed edges from the equation nodes to the variable nodes. Then the MSS set with respect to $\widehat{\text{var}}(M_c^{max})$ is defined as all equation nodes *reachable* from e_i. Denote this MSS set E. Note the difference between this set which is MSS with respect to $\widehat{\text{var}}(M_c^{max})$ instead of MSS with respect to X_u. Differentiate E until $|\widehat{\text{var}}(E^{(i)})| \geq |E|$ using the description of which variables that are linearly contained. Let the obtained differentiated model be M_c. Goto step 2.

3. Rename the current model M_c to M_{diff}.

Output: M_{diff}.

Let $MSS(M)$ denote the set of MSS sets found in equations M and

$$MSS_{all}(M) = MSS(\cup_{i=0}^{\infty} M^{(i)}).$$

Then it is possible to state the following theorem.

Theorem 2: If Assumption 2 is satisfied and for each $y \in \overline{\text{var}}_Y(M_{ext})$, $m(y) < \infty$, then

$$MSS_{all}(M_{ext}) = MSS(M_{diff})$$

The consequence of this theorem is that all MSS sets that are possible to find if the model M_{ext} is differentiated an infinite number of times, can always be found in M_{diff}.

Next the continuation of Example 3 is presented to describe how Algorithm 2 works.

Example 6: Recall that the structural model M_{ext} is the equations e_1, e_2, e_3, e_4, e_6 in Table I. This model satisfy

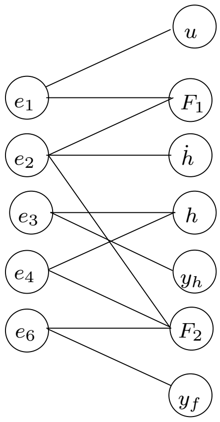


Fig. 2. The bipartite graph $\mathcal{G}(M_{ext}, \mathrm{var}_{X_u \cup Y}(M_{ext}))$.

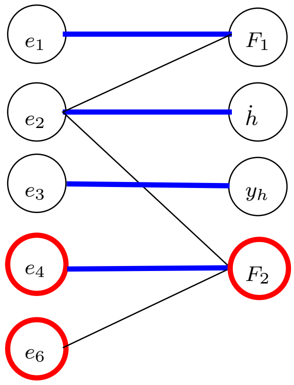


Fig. 3. The bipartite graph $\mathcal{G}(M_c^{max}, \widehat{\mathrm{var}}(M_c^{max}))$ built in step 2. The bold edges are the matching found. The bold equation nodes are the MSS set found in step 2b).

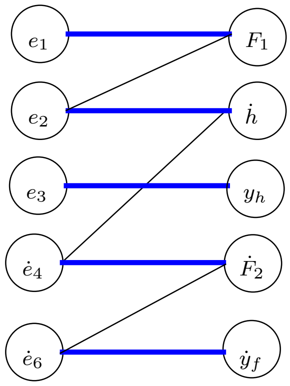


Fig. 4. The bipartite graph $\mathcal{G}(M_c^{max}, \widehat{\mathrm{var}}(M_c^{max}))$ built in step 2 after one differentiation. The bold edges are the complete matching found in step 2.

Assumption 2 since it is possible to find a complete matching of M_{ext} into $\overline{\mathrm{var}}_{X_u \cup Y}(M_{ext})$, e.g. $\{e_1, u\}$, $\{e_2, F_1\}$, $\{e_3, y_h\}$, $\{e_4, F_2\}$, and $\{e_6, y_f\}$. Let $m(u) = m(y_f) = 1$ and $m(y_h) = 0$. According to Theorem 2 it is possible to use Algorithm 2 to produce the model M_{diff}. The corresponding bipartite graph $\mathcal{G}(M_{ext}, \mathrm{var}_{X_u \cup Y}(M_{ext}))$ is shown in Figure 2.

Step 2 in Algorithm 2 is fed with the structural model M_c shown in Figure 2 and the m-values. Figure 3 shows the graph $\mathcal{G}(M_c^{max}, \widehat{\mathrm{var}}(M_c^{max}))$ built in step 2. Note that the node corresponding to h is not considered, because h is not the highest derivative of h in the model. The known variables u and y_f have known derivatives and are therefore not included in $\widehat{\mathrm{var}}(M_c^{max})$. However, the derivative of y_h is an unknown variable and y_h is therefore included.

Step 2 in Algorithm 2 searches for an augmented path from e_1 to $\widehat{\mathrm{var}}(M_c^{max})$ in the graph showed in Figure 3. The path $e_1 - F_1$ is found and this single edge becomes the first assignment. The assignments in the matching are then found in the following order $e_2 - \dot{h}$, $e_3 - y_h$, and $e_4 - F_2$. When e_6 is going to be assigned, there is no variable node left. Since no augmenting path is found, step 2b) finds an MSS set with respect to $\widehat{\mathrm{var}}(M_c^{max})$. When edges not contained in the matching are directed from equation nodes to variable nodes, the reachable equation nodes from e_6 are e_4 and e_6. Hence this is the equation set to be differentiated. The structural differentiation uses additional information about which variables that are nonlinearly included in each equation. Nonlinearly included variables are denoted with O in Table II. Differentiating once implies that $\dot{y}_f$ appears in $\widehat{\mathrm{var}}(\{\dot{e}_4, \dot{e}_6\})$. The new model consists of $\{e_1, e_2, e_3, e_4, \dot{e}_4, e_6, \dot{e}_6\}$ and the new bipartite graph showed in Figure 4 is extracted in step 2. Equation e_4 and e_6 are not anymore the most differentiated equations in the new model. Further, $\dot{y}_f$ is included, because $\ddot{y}_f$ is considered as an unknown variable. Note that an edge in the matching in Figure 3 is either unchanged or replaced with an edge between the replaced nodes corresponding to the differentiated equation and the differentiated variable in Figure 4. For example $\{e_1, F_1\}$ is unchanged and the edge $\{e_4, F_2\}$ in Figure 3 is replaced with $\{\dot{e}_4, \dot{F}_2\}$ in Figure 4. Step 2 finds an assignment for $\dot{e}_6$. The structural model M_{diff} obtained from Algorithm 2 is shown in Table II. The two differentiated equations are $\dot{e}_4 : 2\,(1 - f_c)\; F_2\, \dot{F}_2 - F_2^2\, \dot{f}_c = \dot{h}$ and $\dot{e}_6 : \dot{y}_f = \dot{f}_{yf} + \dot{F}_2$. Note the exact correspondence between the analytical differentiation and the structural differentiation in Table II. $\square$

C. Simplifying the Model

It is a complex task to find all MSS sets in a structural model. Therefore it can be of great help if it is possible to simplify the model. Here two kinds of simplifications are used.

In a first step, all equations in M_{diff} that include any variable that is impossible to eliminate, are removed. This can be done with Canonical Decomposition [7]. The decomposition divides the model in three parts: one structurally overdetermined, one structurally just-determined and one structurally underdetermined part. This is accomplished by first finding a *maximal matching* "see Appendix" in the bipartite graph $\mathcal{G}(E, \mathrm{var}_{X_u}(E))$. Denote the assigned equations and variables in the maximal matching with E_m and X_m respectively. Now, all nodes such that there is an *alternating path* "see Appendix" from $E \backslash E_m$ is

TABLE II
THE STRUCTURAL MODEL M_{diff} OBTAINED FROM ALGORITHM 2 WHEN APPLIED TO THE STRUCTURAL MODEL M_{ext}. NONLINEARLY INCLUDED VARIABLES ARE DENOTED WITH AN O.

equation	unknown					fault						known			
	F_1	F_2	$\dot{F}_2$	h	$\dot{h}$	f_u	f_{yh}	f_c	$\dot{f}_c$	f_{yf}	$\dot{f}_{yf}$	u	y_h	y_f	$\dot{y}_f$
e_1	X					X						X			
e_2	X	X			X										
e_3				X			X						X		
e_4		O		X				O							
$\dot{e}_4$		O	O		X			O	O						
e_6		X								X				X	
$\dot{e}_6$			X								X				X

the structurally overdetermined part of the model. Denote this part of the model M_{simp1}. The structurally underdetermined part of the model are the nodes such that there is an alternating path from $\text{var}_X(E)\backslash X_m$. The remaining part of the model is the structurally just-determined part. Assumption 1 implies that structural overdetermination is needed for analytical overdetermination and therefore it is only the model M_{simp1} that is considered in the continuation.

Example 7: Continuation of Example 6. It can be seen in Table II that one maximal matching in $\mathcal{G}(M_{diff}, var_{X_u}(M_{diff}))$ is $\{e_1, F_1\}$, $\{e_2, \dot{h}\}$, $\{e_3, h\}$, $\{e_4, F_2\}$, and $\{\dot{e}_4, \dot{F}_2\}$. It is easy to verify that there is an alternating path from e_6 or $\dot{e}_6$ to each equation in M_{diff}. Hence $M_{simp1} = M_{diff}$. □

In a second step, variables that can be eliminated without losing any structural information are found. The rest of this section will be devoted to a discussion about this second step.

If there is a set $X \subseteq X_u$ with the property $1 + |X| = |\text{equ}_{M_{simp1}}(X)|$, then all equations in $\text{equ}_{M_{simp1}}(X)$ have to be used to eliminate all variables in X. Since all unknown variables must be eliminated in an MSS set this means particularly that all MSS sets including any equation of $\text{equ}_{M_{simp1}}(X)$ have to include all equations in $\text{equ}_{M_{simp1}}(X)$. The idea is to find these sets. Then it is possible to eliminate internal variables, here denoted X, in these sets. Every set is replaced with one new equation denoted $\mathcal{E}(\text{equ}_{M_{simp1}}(X), X)$. For simplicity, the second argument will be omitted in the rest of this paper.

This second simplification step finds subsets of variables that are included in exactly one more equation than the number of variables. To reduce the computational complexity, a complete search for such sets is in fact not performed here. Instead only a search for single variables included in two equations is done. When a variable is included in just two equations, these equations are used to eliminate the variable. If all variables are examined and some simplification was possible, then all remaining variables have to be examined once more. When no more simplifications can be made, the simplification step is finished and the resulting structural model is denoted M_{simp}. Note that with this strategy larger sets than two equations will also be found, since the algorithm can merge sets found in previous steps.

The next theorem ensures that no MSS set is lost in the simplification step.

Theorem 3: $MSS(M_{diff}) = MSS(M_{simp})$
For a proof, see [12].

Example 8: Consider Example 7 with the structural model shown in Table II. The second simplification step searches for variables which belong only to two equations. In the first search, the algorithm finds F_1 in $\{e_1, e_2\}$, $\dot{F}_2$ in $\{\dot{e}_4, \dot{e}_6\}$, and h in the equations produced by $\{e_1, e_2\}$ and $\{\dot{e}_4, \dot{e}_6\}$. This makes one group of $\{e_1, e_2, \dot{e}_4, \dot{e}_6\}$. This search made simplifications and therefore the search is performed once more. The second time no simplifications are done and the simplification step is therefore complete. The structural model is shown in Table III and a corresponding analytical model is shown in Table IV.

TABLE III
THE STRUCTURAL MODEL M_{simp} FOR THE WATER TANK SYSTEM SHOWN IN FIGURE 1

equation	unknown	fault						known			
	F_2	f_u	f_{yh}	f_c	$\dot{f}_c$	f_{yf}	$\dot{f}_{yf}$	u	y_h	y_f	$\dot{y}_f$
e_6	X					X				X	
$\mathcal{E}(\{e_3, e_4\})$	X		X	X					X		
$\mathcal{E}(\{e_1, e_2, \dot{e}_4, \dot{e}_6\})$	X	X		X	X		X	X			X

□

D. Finding MSS Sets

After the simplification step is completed, step 3 in Algorithm 1 finds all MSS sets in the simplified model M_{simp}. This section explains how the MSS sets are found.

The task is to find all MSS sets in the model M_{simp} with equations $\{e_1, \cdots, e_n\}$. Let $M_k = \{e_k, \cdots, e_n\}$ be the last $n - k + 1$ equations. Let E be the current set of equations that is examined. The set of MSS sets found is denoted ω_{alg3}. Then the following algorithm finds all MSS sets in M_{simp}.

Algorithm 3:
Input: The model M_{simp}.
1. Set $k = 1$ and $\omega_{alg3} = \varnothing$.
2. Choose equation e_k. Let $E = \{e_k\}$ and $X = \varnothing$.
3. Find all MSS sets that are subsets of M_k and include equation e_k.

TABLE IV
AN EXAMPLE OF AN ANALYTICAL MODEL AFTER THE SIMPLIFICATION STEP.

equation	expression
e_6	$y_f = F_2 + f_{yf}$
$\mathcal{E}(\{e_3, e_4\})$	$y_h = F_2^2\,(1 - f_c) + f_{yh}$
$\mathcal{E}(\{e_1, e_2, \dot{e}_4, \dot{e}_6\})$	$u = F_2(1 - F_2\,\dot{f}_c + 2\,(1 - f_c)(\dot{y}_f - \dot{f}_{yf})) - f_u$

a) Let $\tilde{X} = \text{var}_{X_u}(E) \backslash X$ be the unmatched variables.
b) If $\tilde{X} = \varnothing$, then E is an MSS set. Insert E into ω_{alg3}.
c) Else take a remaining variable $\tilde{x} \in \tilde{X}$ and let $X = X \cup \{\tilde{x}\}$. Let $\tilde{E} = \text{equ}_{M_k \backslash E}(\tilde{x})$ be the remaining equations. For all equations e in $\tilde{E}$ let $E = E \cup \{e\}$ and goto step a).
4. If $k < n$ set $k = k + 1$ and goto step number 2.
Output: The set of MSS sets found, i.e. ω_{alg3}.
Algorithm 3 finds all MSS sets in M_{simp} according to the next theorem proven in [12].

Theorem 4: $\omega_{alg3} = MSS(M_{simp})$

In [5] there is a similar algorithm for computing *minimal evaluation chain* (MEC) which is almost the same as the MSS sets. The difference is that a MEC has to include known variables. The next example not related to any previous example illustrates Algorithm 3.

Example 9: Consider the following structural model.

	x_1	x_2	x_3
1	X	X	
2		X	X
3	X	X	X
4	X		
5		X	

This model gives the following time evolution of current equations, i.e. E in Algorithm 3 is

$$
\begin{array}{ccccccccccc}
 & & & & \mathbf{2} & & & \mathbf{3} & & \mathbf{2} & \\
 & & 2 & 5 & \mathbf{5} & & 2 & \mathbf{2} & 3 & \mathbf{3} & \mathbf{5} \\
 & 3 & 3 & 3 & \mathbf{3} & 4 & 4 & \mathbf{4} & 4 & \mathbf{4} & \mathbf{4} \\
1 & 1 & 1 & 1 & \mathbf{1} & 1 & 1 & \mathbf{1} & 1 & \mathbf{1} & \mathbf{1}
\end{array}
$$

$$
\begin{array}{ccccccccccc}
 & & & & & \mathbf{4} & & & & & \\
 & & 4 & & 3 & \mathbf{3} & & & 5 & & \\
 & 3 & 3 & 5 & 5 & \mathbf{5} & & 4 & 4 & & \\
2 & 2 & 2 & 2 & 2 & \mathbf{2} & 3 & 3 & 3 & 4 & 5
\end{array}
$$

The bold columns represent the MSS sets found. This example also shows that if there are several matchings including the same equations, the algorithm finds the same subset of equations several times. □

Example 10: Continuation of Example 8. The 3 MSS sets that can be found in Table III are shown in the left column in Table V. The matrix in this table is the incidence matrix of the MSS sets. □

E. Analyzing Isolability

When the MSS sets are found, the next step is to analyze their isolability. If any equation in the MSS set i include fault j, the element (i, j) of the incidence matrix is equal to X. Note that an X in position (i, j) is no guarantee for fault j to appear in the MSS set i. Moreover it is no guarantee that a fault j that appears in the MSS set i has to make MSS set i inconsistent, i.e. exoneration is not assumed.

Example 11: To give an example of the interpretation of an incidence matrix, consider the second MSS set in Table V. This MSS set could contain f_u and f_{yf}, but it is impossible that it could contain f_{yh}, since f_{yh} is only included in equation e_3. □

If the number of different faults is large it is not easy to see which faults that can be isolated from each other. The incidence matrix of the MSS sets show which faults that could be responsible for an inconsistency of each MSS set, but it is more interesting to see which fault that can be explained by other faults. An *isolability matrix* shows the maximum isolation and detection capability of the diagnosis system. The maximum isolation capability with a diagnosis system designed with this structural method is obtained if it is assumed that each fault makes all MSS sets including this fault inconsistent. If fault j is sensitive to at least all MSS sets that fault i is sensitive to, then element (i, j) of the isolability matrix is equal to X. The interpretation of an X in position (i, j) is that fault f_i can not be isolated from fault f_j.

TABLE V

THE STRUCTURAL MODEL OF THE MSS SETS.

MSS	fault						known			
	f_u	f_{yh}	f_c	$\dot{f}_c$	f_{yf}	$\dot{f}_{yf}$	u	y_h	y_f	$\dot{y}_f$
$\{e_1, e_2, e_3, e_4, \dot{e}_4, \dot{e}_6\}$	X	X	X	X	X	X	X	X		X
$\{e_1, e_2, \dot{e}_4, e_6, \dot{e}_6\}$	X		X	X	X	X	X		X	X
$\{e_3, e_4, e_6\}$		X	X		X			X	X	

TABLE VI

THE ISOLABILITY MATRIX OF THE STRUCTURAL MODEL IN TABLE V.

present fault	interpreted fault				
	no fault	F_u	F_{yh}	F_c	F_{yf}
no fault	X	X	X	X	X
F_u		X		X	X
F_{yh}			X	X	X
F_c				X	X
F_{yf}				X	X

Example 12: The isolability matrix corresponding to the incidence matrix in Figure V is shown in Table VI. Consider the first row of the isolability matrix. Suppose that fault f_u is present. Then, the first three MSS sets are not satisfied in an ideal case. This means that f_u certainly can explain fault f_u, but also f_{yf} can explain fault f_u. Fault f_{yh} cannot explain fault f_u, since if f_{yh} is present, the third MSS set is satisfied. Note that the isolability matrix is not symmetric. For example fault f_{yf} can explain fault f_u but the opposite is not true. □

The isolability matrix can more easily be analyzed after Dulmage-Mendelsohn permutations [13]. This algorithm returns a maximal matching which is in block upper-triangular form. The faults in each diagonal block can never be distinguished with that diagnosis system.

Example 13: In Table VI, the same matrix is returned after Dulmage-Mendelsohn permutations, which usually is not the case. The diagonal blocks are the 1×1 diagonal elements {no fault}, $\{F_u\}$, $\{F_{yh}\}$ and the 2×2 diagonal element $\{F_c, F_{yf}\}$. □

F. Decoupling Faults

Let I be a set of *isolability properties*. An isolability property is that a fault becomes detectable or some fault f_1 can be isolated from some other fault f_2. The set I can be specified by the demands of the isolability capability of

TABLE VII

THE STRUCTURAL MODEL OF THE ADDITIONAL MSS SETS.

MSS	fault						known			
	f_u	f_{yh}	f_c	$\dot{f}_c$	f_{yf}	$\dot{f}_{yf}$	u	y_h	y_f	$\dot{y}_f$
$\{e_1, e_2, e_3, e_4, \dot{e}_4, e_5, e_6, \dot{e}_6\}$	X	X			X	X	X	X	X	X
$\{e_1, e_2, e_3, e_4, \dot{e}_4, \dot{e}_6, e_7\}$	X	X	X	X			X	X		X

TABLE VIII

DESIRED ISOLABILITY MATRIX OF THE WATER TANK EXAMPLE.

present	interpreted fault				
fault	no fault	F_u	F_{yh}	F_c	F_{yf}
no fault	X	X	X	X	X
F_u		X			
F_{yh}			X		
F_c				X	
F_{yf}					X

the diagnosis system or just set to full isolation capability. An isolability matrix can define the desired isolability.

Example 14: Continuation of Example 12. The desired isolability I_{des} can be chosen as full isolability. This corresponding matrix is shown in Table VIII. Note that each empty entry in the matrix define one isolability property in I_{des}. This means for example that $|I_{des}| = 16$, i.e. there are 16 desired isolability properties. □

Suppose that the element (i, j) of the isolability matrix is equal to X but the desired isolability matrix is not equal to X for some $i \neq j$. This means that the desired isolability is not fulfilled. However, it could still be possible to isolate fault i from fault j by trying to decouple fault j. Include fault j among the unknown variables X_u and search for new MSS sets by applying Algorithm 1 from step 2 to the new model obtained. An MSS set that is able to isolate fault i from fault j has to include at least one equation that includes fault i. If any such MSS set is found, it has to include an elimination of fault j. If not, this MSS would have been discovered earlier.

Example 15: Continuation of Example 14. The isolability of the MSS sets found is shown in Table VI and the desired isolability is shown in Table VIII. The lack of desired isolability fulfillment is seen in the isolability matrix as the additional X:s. The third column in the isolability matrix shows that f_u, f_{yh}, and f_{yf} can not be isolated from f_c. The problem is that there is no MSS set that decouple fault f_c. But there could be one if f_c is eliminated. The fault f_c is moved from the faults F to the unknown variables X_u. The procedure starts all over from the step 2 in Algorithm 1. The result is a new MSS set in which f_c is decoupled. Furthermore f_{yf} has also to be decoupled. Decoupling f_{yf} gives also one extra MSS set. These two additional MSS sets, shown in Table VII, together with the previous MSS sets give a possibility to detect and isolate all faults and the desired isolability is fulfilled. □

G. Selecting a Subset of MSS Sets

It is not unusual that the number of MSS sets found is very large. Many of the MSS sets probably use almost as many equations as unknown variables in the entire system. These MSS sets usually rely on too many uncertainties to be usable for fault isolation. Small MSS sets are more robust and are usually sensitive to fewer faults. Therefore the goal must be to find a set of MSS sets with the highest possible *robustness* and with the same isolation capability as the set of all MSS sets.

Assume that it is possible to calculate a real number for each MSS set that is inversely proportional to the robustness of the MSS set. Let this number for MSS set m be denoted n_m. This number can for example be as in this paper, the number of equations in each MSS set.

Now, let $\Psi(\Omega, i)$ be a function that has a set Ω of MSS sets and an isolability property i as arguments and returns the MSS sets of Ω with property i.

The set of MSS sets ω has maximum isolation capability and maximum robustness if ω fulfills

$$\begin{array}{c} \forall i \in I\big(\Psi(MSS_{all}(M_{orig}), i) \neq \varnothing \rightarrow \\ \rightarrow \min_{\psi \in \Psi(MSS_{all}(M_{orig}), i)} n_\psi = \min_{\psi \in \Psi(\omega, i)} n_\psi\big) \end{array} \tag{4}$$

Note that ω is not unique, but the minimum number for each $i \in I$ is unique.

Step 7 in Algorithm 1 start to sort the MSS sets in an increasing order of robustness. The MSS sets are examined in the rearranged order. If an MSS set increase the isolability, then select the MSS set. This means that for each detection of a fault and for each isolation between two faults, the smallest MSS sets with this isolation capability will be one of the chosen MSS sets. In this way the final output from Algorithm 1 will be a small set ω of MSS sets with highest possible isolation capability and highest possible robustness, i.e. ω fulfills expression (4). An upper bound to $|\omega|$ is $|I_{des}|$.

Example 16: Continuation of Example 15. Step 7 in Algorithm 1 rearranges the MSS sets in increasing size as shown in Table IX. It is possible to define an upper limit of the isolability requirement fulfillment as $|I|/|I_{des}|$ where I is the calculated isolability given a set of MSS sets. This number is shown in the right column of Table IX when the MSS sets are added one at a time. The third MSS set in Table IX does not increase the isolability requirement fulfillment and is therefore rejected. Hence the MSS sets 1, 2, 4, and 5 are selected. Using the function `Eliminate` in Mathematica and setting all fault variables to zero produces the consistecy relations

$$\begin{array}{l} y_f^2 - y_h = 0 \\ y_f + 2\,y_f\,\dot{y}_f - u = 0 \\ y_h + 4\,y_h\,\dot{y}_f + 4\,y_h\,\dot{y}_f^2 - u^2 = 0 \\ y_f^2 + 2\,y_h\,\dot{y}_f - u\,y_f = 0 \end{array}$$

for each MSS set respectively.

□

TABLE IX
THE STRUCTURAL MODEL OF THE MSS SETS.

MSS	f_u	f_{yh}	f_c	$\dot{f}_c$	f_{yf}	$\dot{f}_{yf}$	accumulated isolability
$\{e_3, e_4, e_6\}$		X	X		X		38%
$\{e_1, e_2, \dot{e}_4, e_6, \dot{e}_6\}$	X		X	X	X	X	63%
$\{e_1, e_2, e_3, e_4, \dot{e}_4, \dot{e}_6\}$	X	X	X	X	X	X	63%
$\{e_1, e_2, e_3, e_4, \dot{e}_4, \dot{e}_6, e_7\}$	X	X	X	X			81%
$\{e_1, e_2, e_3, e_4, \dot{e}_4, e_5, e_6, \dot{e}_6\}$	X	X			X	X	100%

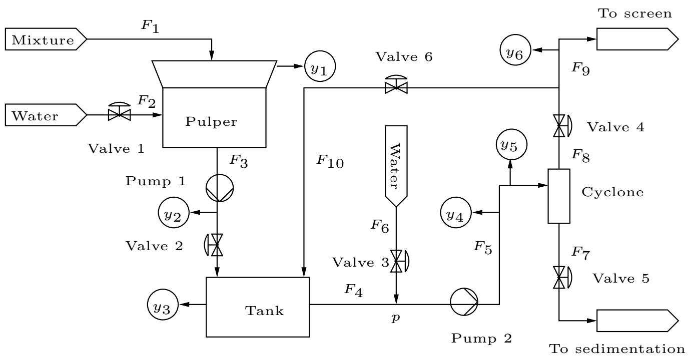


Fig. 5. A stock preparation and broke treatment system of a paper mill.

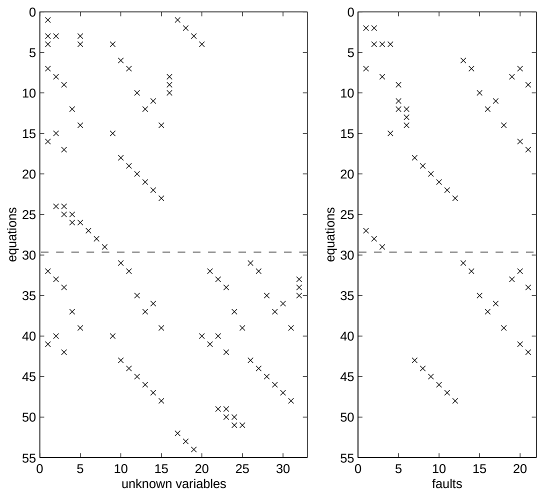


Fig. 6. Structural model of the stock preparation and broke treatment system.

V. Industrial Example: A Part of a Paper Mill

This example is a stock preparation and broke treatment system of a paper mill located in Australia. The system is used for mixing and purifying recycled paper for production of new paper. An overview of the system is shown in Figure 5.

A. System Description

Most parts of the system are nonlinear and it is only the tank and the pulper that are considered to be dynamic. Because of space considerations, the details of the model are omitted, but can be found in [12], [14]. The system has 4 states: the volume and concentration in the pulper and in the tank. There are 6 sensors in the system. Sensor y_1 and y_3 measure the water levels of the pulper and the tank respectively, y_2 and y_4 measure concentration, y_5 and y_6 measure pressure. The flows and concentrations into this system are known and the flows out from the system are also known. There are 6 valves and two pumps that are actuators with known inputs.

There are 21 faults that are considered. All sensors can have a constant offset fault $f_1, \ldots, f_6$. All valves can have a constant offset in the actuator signal $f_7, \ldots, f_{12}$. Clogging can occur in the pipes near the valves $f_{13}, \ldots, f_{18}$ and also directly after the tank f_{19}. Finally, the pumps can have a constant offset in the actuator signal f_{20}, f_{21}.

The system is described by 29 equations and additionally 21 equations describing fault models. Equations $e_1, \ldots, e_4$ describe the dynamics; $e_5, \ldots, e_{14}$ are pressure loops; e_{15} relates the concentration in the junction after the tank with the flows F_4 and F_6; e_{16} and e_{17} describe the two pumps; $e_{18}, \ldots, e_{23}$ are valve equations; $e_{24}, \ldots, e_{26}$ are flow equations; and finally $e_{27}, \ldots, e_{29}$ are sensor equations for sensor 1, 2, and 3. Finally there are the 21 equations describing each fault.

B. Extracting the No-Fault Model

The extracted model contains all equations except for the 21 equations describing fault models. The extracted structural model can be viewed in the first 29 rows in the matrices in Figure 6.

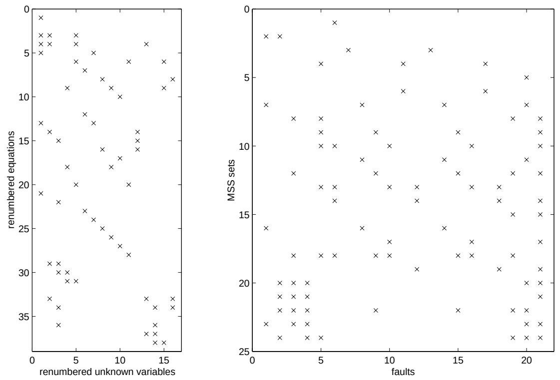
C. Differentiating the Model

The highest order of derivatives that is known for all known variables are assumed to be one. If a variable is contained linearly in an equation the variable disappears in the differentiated expression. This knowledge is used since the equations are known. Algorithm 2 is applied to the first 29 equations in Figure 6. The result is that all equations except equation 1, 2, 3, and 4 are differentiated. This results in additionally 25 differentiated equations shown in the lower part of Figure 6.

D. Simplifying the Model

In the first step of simplification applied to the left matrix in Figure 6, the equations $\{27, 28, 29\}$ include variables belonging only to one equation, i.e. they cannot be included in any MSS sets.

The second part of the simplification finds that the variables $\{9, 17, 18, 19, 20, 21, 25, 26, 27, \ 28, 29, 30, 31\}$ can be eliminated. The equations that form groups are $\{1, 52\}$, $\{2, 53\}$, $\{3, 54\}$, $\{4, 15, 40\}$, $\{32, 41, 44\}$, $\{39, 48, 51\}$, $\{31, 43\}$, $\{35, 45\}$, $\{37, 46\}$ and $\{36, 47\}$. The simplified structural model is shown in Figure 7 (a). Note the simplification of the model by comparing Figure 6 and Figure 7 (a).



(a) (b)

Fig. 7. The simplified structural model is shown in (a). The incidence matrix of the MSS sets is shown in (b)

E. Finding MSS Sets

Algorithm 3 is then applied to the simplified model. The algorithm returns 35 770 MSS sets that are contained in the simplified model. The largest MSS set consists of 24 equations.

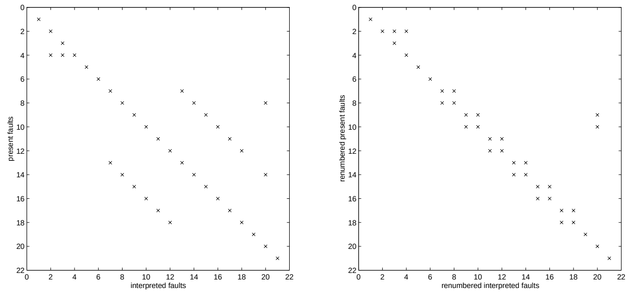
F. Analyzing Isolability

The isolability matrices before and after the Dulmage-Mendelsohn permutations are seen in Figure 8. The Dulmage-Mendelsohn permutations gives that the faults $\{7, 13\}$, $\{8, 14\}$, $\{9, 15\}$, $\{10, 16\}$,$\{11, 17\}$ and $\{12, 18\}$ are never distinguishable. These pairs of faults all belong pairwise to the same valve. This isolation performance for faults concerning valves is in this case acceptable. Hence the desired isolability I_{des} does not include isolation between pairs of faults belonging pairwise to the same valve.

The result of the isolability analysis is now that it is not possible to isolate faults 4, 8, and 14. This follows from the difference between the isolability shown in Figure 8 (a) and the desired isolability I_{des}.

G. Decoupling Faults

Considering Figure 8, it is still important to discover if any MSS set can decouple fault 2 or 3 and be sensitive to fault 4. It is also necessary to decouple fault 20. Apply Algorithm 1 to the original model, but where fault 2 now is considered to be an unknown variable. Then apply the Algorithm 1 to the model where faults 3 is decoupled and finally also when fault 20 is decoupled. The algorithm finds thereby additional MSS sets that isolate fault 4, 8, and 14.



(a) (b)

Fig. 8. These matrices are the isolability matrices before (a) and after (b) the Dulmage-Mendelsohn permutation.

H. Selecting a Subset of MSS Sets

The 24 chosen MSS sets are

MSS	
1	13
2	2 53
3	6 18
4	11 22
5	1 16 52
6	22 36 47
7	7 16 19
8	8 9 17 24
9	9 10 17 20
10	12 17 21 25
11	16 19 32 41 44
12	8 10 17 20 24
13	12 14 21 23 26
14	14 17 23 25 26
15	17 24 33 34 42 49
26	7 16 19 32 41 44
17	17 21 25 37 42 46 50
18	8 10 12 20 21 24 25
19	17 23 25 26 39 42 48 50 51
20	3 4 15 16 17 24 40 42 49 54
21	1 3 4 15 17 24 40 42 49 52 54
22	3 4 8 10 15 16 20 33 35 40 45 54
23	2 3 4 15 16 17 24 40 42 49 53 54
24	3 4 8 9 15 16 17 24 40 42 49 54

(5)

From these sets and the structural model in Figure 6 the incidence matrix in Figure 7 (b) is obtained.

I. Generating Consistency Relations

Consistency relations corresponding to the 24 MSS sets are calculated by using the function `Eliminate` in Mathematica. Most of the equations in the model are polynomial equations. For polynomial equation-systems, the function `Eliminate` uses Gröbner Basis techniques for elimination. Each MSS set with 7 or less equations was easily eliminated to a consistency relation. The consistency relations from the MSS set 17 and 18 were obtained from the `Eliminate` function, but were to complex to be numerically reliable. Elimination of the unknown variables in MSS sets with 8 or more equations was computational intractable with the `Eliminate` function. Therefore, by using only consistency relations obtained from the 15 first MSS sets, the isolation capability was reduced slightly. Some further results of the investigation can be found in [12].

VI. Conclusion

This paper has presented a systematic and automatic method for finding a small set of submodels that can be used to derive consistency relations with highest possible isolation capability. The method is based on graph theoretical reasoning about the structure of the model. It is assumed that a condition on algebraic independency is fulfilled.

An important idea, towards finding these submodels, is to use the mathematical concept *minimal structurally singular* sets. These sets have in Theorem 1 been shown to characterize these submodels, i.e. the consistency relations, which give the fault detection and the fault isolation capability.

The method is capable of handling general differential-algebraic non-causal equations. Further, the method is not limited to any special type of fault model. Algorithm 1 finds all submodels that can be used to derive consistency relations and this is proven in Theorem 2, 3, and 4. The key step in Algorithm 1 is step 3 that finds all MSS sets in the model it is applied to.

Finally the method has been applied to a large nonlinear industrial example, a part of a paper mill. The algorithm successfully manage to derive a small set of submodels. In spite of the complexity of this process, a sufficient number of submodels could be transformed to consistency relations so that high isolation capability was obtained.

References

[1] J. De Kleer, A. K. Mackworth, and R. Reiter, "Characterizing diagnoses and systems," *Artificial Intelligence*, vol. 56, 1992.

[2] R. Reiter, "A theory of diagnosis from first principles," *Artificial Intelligence*, vol. 32, pp. 57–95, 1987.

[3] J. De Kleer and B.C. Williams, "Diagnosing multiple faults," *Artificial Intelligence*, vol. 32, pp. 97–130, 1987.

[4] M-O. Cordier, P. Dague, M. Dumas, F. Levý, J. Montmain, M. Staroswiecki, and L. Travé-Massuyés, "AI and automatic control approaches of model-based diagnosis: Links and underlying hypotheses," in *4th IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes*, 2000, vol. 1, pp. 274–279.

[5] B. Pulido and C. Alonso, "Possible conflicts, ARRs, and conflicts," in *Proceedings of the Thirteenth International Workshop on Principles of Diagnosis*, 2002, pp. 122–128.

[6] B. Pulido and C. Alonso, "An alternative approach to dependency-recording engines in consistency-based diagnosis," in *Lecture Notes in Artificial Intelligence.* Artificial Intelligence: Methodology, Systems, and Applications. 9th International Conference, AIMSA 2000, 2000, vol. 1904, pp. 111–121, Springer-Verlag, Berlin, Germany; 2000.

[7] J. P. Cassar and M. Staroswiecki, "A structural approach for the design of failure detection and identification systems.," in *IFAC Control of Industrial Systems*, Belford, France, 1997.

[8] L. Travé-Massuyès, T. Escobet, and R. Milne, "Model-based diagnosability and sensor placement. application to a frame 6 gas turbine subsystem," in *DX01 twelfth international workshop on principals of diagnosis*, 2001, pp. 205–212.

[9] F. Harary, *Graph theory*, Addison-Wesley publishing company, Reading, Mass, 1969.

[10] E. Carpanzano and C. Maffezzoni, "Symbolic manipulation techniques for model simplification in object-oriented modeling of large scale continuous systems," *Mathematics and Computers in Simulations*, vol. 48, pp. 133–150, 1998.

[11] C C. Pantelides, "The consistent initialization of differential-algebraic systems," *SIAM J. SCI. STAT. COMPUT.*, vol. 9, no. 2, pp. 213–231, March 1988.

[12] M. Krysander and M. Nyberg, "Structural analysis for fault diagnosis of DAE systems utilizing graph theory and MSS sets," Tech. Rep. LiTH-ISY-R-2410, Dept. of Electrical Engineering Linköpings universitet, 2002, Available: http://www.vehicular.isy.liu.se/Publications/.

[13] G. Meurant, *Computer Solution of Large Linear Systems*, Number ISBN 0-444-50169-X. Elsevier Science B. V., Amsterdam : North Holland, 1999.

[14] Jonas Biteus, "Diagnosis of fluid systems utilizing gröbner bases and filtering of consistency relations," Tech. Rep. LiTH-ISY-EX-3237, Department of Electrical Engineering, Linköpings universitet, SE-581 83 Linköping, Sweden, 2001.

Appendix

Bipartite Graph. A graph with edges Γ is bipartite if its nodes can be partitioned into two sets E and X such that no two nodes in the same set have an edge in common. This graph may be written as (E, X, Γ).

Matching. Given a graph with edges Γ a matching is a set of edges $\Gamma_0 \subseteq \Gamma$ such that no two edges have a node in common. A matching Γ_0 is maximal if $|\Gamma_1| > |\Gamma_0|$ implies that Γ_1 is not a matching.

Complete Matching. Given a bipartite graph (E, X, Γ), a complete matching of E into X is a matching such that all nodes in E is an endpoint of an edge.

Alternating Path. Consider a matching Γ_0 in a graph with edges Γ. An alternating path is defined as a path whose edges are alternately in Γ_0 and in $\Gamma\backslash\Gamma_0$.

Augmented path. An alternating path in a matching Γ_0 is an augmented path in Γ_0 if it begins and ends at two distinct unmatched nodes.