

## STRUCTURAL ANALYSIS FOR FAULT DIAGNOSIS OF DAE SYSTEMS UTILIZING MSS SETS

Mattias Krysanter and Mattias Nyberg

Department of Electrical Engineering, Linköping University  
SE-581 83 Linköping, Sweden  
Phone: +46 13 282198, Email: 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: Structural analysis; Fault isolation; Minimal structurally singular sets; Consistency relation; Graph theory; Fault diagnosis

### 1. INTRODUCTION

When designing model-based fault-diagnosis systems, the use of *consistency relations* (also called e.g. *parity relations*) is a common choice. The idea is to have a set of consistency relations, with different subsets 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 non-linear 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 a structural approach is chosen. Instead of directly manipulating the equations themselves, the proposed algorithm in this paper only deals with the structural information contained in the model, i.e. which variables that appear in each equation. Each consistency relation is derived from a submodel, i.e. a part of the complete model. The structural algorithm finds these submodels. These submodels obtained are typically much smaller than the whole model, and therefore the computational complexity of finding consistency relations decreases.

The structural information is collected in a *structural model*. By analyzing and manipulating this structural model, the proposed algorithm picks out all submodels that can be used to derive consistency relations. Moreover, of these submodels, it selects a small set 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 only needs

as input a structural model of the system. This 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 (Pulido and Alonso, 2000) 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 (Cassar and Staroswiecki, 1997) and an extension to this work considering sensor placement is found in (Travé-Massuyès *et al.*, 2001).

In Sections 2 and 3, structural models and their usefulness in fault diagnosis are discussed. Then in Section 4, a complete description of the algorithm is given. The algorithm is in (Krysanter and Nyberg, 2002) applied to a large nonlinear industrial process, a part of a paper plant. In spite of the complexity of this process, a small set of consistency relations with high diagnosis capability is successfully derived.

### 2. STRUCTURAL MODELS

The behavior of a system is described with a model. Usually the model is a set of equations. A structural model (Cassar and Staroswiecki, 1997) contains only 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* (Harary, 1969; Carpanzano and Maffezzoni, 1998). 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 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 water level in the tank  $y_h$  and the flow from the tank  $y_f$ . One fault denoted  $f_i$  is associated to each known variable. The actual flows to and from the tank are denoted  $F_i$ , and the actual water level in the tank is denoted  $h$ . Without knowing the exact physical equations describing the analytic model the structural model can be set up as follows:

equation	unknown				fault				known		
	$F_1$	$F_2$	$h$	$\dot{h}$	$f_u$	$f_h$	$f_f$	$\dot{f}_f$	$u$	$y_h$	$y_f$
$e_1$	X				X				X		
$e_2$	X	X		X							
$e_3$			X			X				X	
$e_4$		X	X								X
$e_5$		X					X				X
$e_6$								X			

Equation  $e_1$  describe 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,  $e_5$  the flow measurement and  $e_6$  a fault model for the flow measurement fault  $f_f$ .  $\diamond$

### 3. 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 diagnosability 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 the new definition of *consistency relation for an equation set* 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  $E$  that depends on variables contained in  $X_u$  and  $Y$ .

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

$$\exists \mathbf{x} E(\mathbf{x}, \mathbf{y}) \Leftrightarrow c(\mathbf{y}) = 0 \quad (2)$$

and there is no proper subset of  $E$  that has property (2).

The idea to define consistency relations for an equation set  $E$  in this way is to ensure the following: it is a sufficient explanation for an inconsistency  $c(\mathbf{y}) \neq 0$  that any equation in  $E$  is not valid.

Definition 1 differ from the common definition of consistency relation in two ways, the left implication in (2) and that there is no proper subset of  $E$  that has property (2). Refer the latter as the minimality condition in Definition 1. The following example shows the importance of the left implication in (2).

*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 it is true even if e.g.  $y_3 \neq y_1 = y_2$  and then it is impossible to find a consistent  $x$  in  $E$ . However  $y_1 - y_2 = 0$  is a consistency relation for  $\{y_1 = x, y_2 = x\}$ .

The expression  $y_1 + y_2 - 2y_3 = 0$  includes  $y_3$ . The right implication in (2) holds, but the opposite direction does not hold. The conclusion is that also this expression is not a consistency relation for  $E$  or any equation subset of  $E$ .

However  $(y_1 - y_2)^2 + (y_2 - y_3)^2 = 0$  is a consistency relation for  $E$ .  $\diamond$

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

#### 3.1 Basic Assumptions

Basic assumptions are needed to guarantee that the subsets found only by analyzing structural properties are those subsets that can be used to form consistency relations. Before the basic assumptions are presented, some notation is needed. Let  $E$  be any set of equations and  $X$  any set of variables. Then define  $var_X(E) = \{x \in X \mid \exists e \in E : e \text{ contains } x\}$  and  $equ_E(X) = \{e \in E \mid \exists x \in X : e \text{ contains } x\}$ . Also, let  $var_X(e)$  and  $equ_E(x)$  be shorthand notations for  $var_X(\{e\})$  and  $equ_E(\{x\})$  respectively. If  $g$  is any equation, function or variable, let  $g^{(i)}$  denote the  $i$ :th time derivative of  $g$ . Then define  $\overline{var}_X(E) = \{\text{undifferentiated } x \mid \exists i(x^{(i)} \in var_X(E))\}$ , e.g.  $\overline{var}_{X_u \cup Y}(\{y = \dot{x}\}) = \{y, x\}$ . Finally, the number of elements in any set  $E$  is denoted  $|E|$ .

The first assumption is used to ensure that the model becomes finitely differentiated in Section 4.1.

*Assumption 1.* The model  $M_{orig}$  has the property  $\forall E \subseteq M_{orig} : |E| \leq |\overline{var}_{X_u \cup Y}(E)|$ . (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 condition (1) is not fulfilled and there are no redundant equations, the model would normally be inconsistent.

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 2.* There exists a consistency relation  $c(\mathbf{y}) = 0$  for the equation set  $H$  iff  $\forall X' \subseteq var_{X_u}(H), X' \neq \emptyset : |X'| < |equ_H(X')|$  (4)

According to Assumption 2 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 than the number of equations in  $H$  which contain some of the variables in the chosen subset.

The Assumptions 1 and 2 are often fulfilled. For example all subsets of equations found in the industrial example in the end of the paper satisfy Assumption 2. Even though the "only if" direction of Assumption 2 is difficult to validate in an application, the results of the paper can still be used to produce a lower bound of the actual detection and isolation capability. If all subsets of the model fulfill Assumption 2, the structural analysis will find all subsets that can be used to find consistency relations.

### 3.2 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 (4). To do this, two important structural properties are defined (Pantelides, 1988).

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

*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 text. The next theorem from (Krysander and Nyberg, 2002) 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.* Let  $H \subseteq M_{orig}$ , where  $M_{orig}$  fulfills Assumption 1. Further, let  $H$  and all  $E_i$  fulfill Assumption 2. Then there exists a consistency relation  $c(y) = 0$  for  $H(x, y)$  where  $|H| < \infty$  iff  $H = \bigcup_i E_i$  where for each  $i$ ,  $E_i$  is an MSS set.

## 4. ALGORITHM FOR FINDING AND SELECTING MSS SETS

The objective is to find all potential consistency relations for a given model  $M_{orig}$  and then choose a small subset of these consistency relations with the same diagnosability as the full set of the consistency relations. This is done by finding all MSS sets in a differentiated version of the model  $M_{orig}$ . The algorithm can be summarized in the following steps.

*Algorithm 1.*

- (1) Differentiating the model: Find equations that are meaningful to differentiate for finding MSS sets.

- (2) Simplifying the model: Given the original model and the additional equations found in step (1), 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.
- (3) Finding MSS sets: Search for MSS sets in the simplified model.
- (4) Analyzing Diagnosability: Examine the diagnosability of the MSS sets found in step (3).
- (5) Decoupling faults: If the diagnosability has to be improved, some faults have to be decoupled. For decoupling faults, return to step (1) and consider these faults as unknown variables in  $X_u$ .
- (6) Selecting a subset of MSS sets: Select the simplest set of MSS sets that contains the desired diagnosability.

The following sections discuss each of the steps in Algorithm 1.

### 4.1 Differentiating the Model

To handle dynamic models, Algorithm 1 needs a way to deal with derivatives. In this section an algorithm for handling derivatives is defined. This algorithm is referred to as Algorithm 2. A small example will show what Algorithm 2 must be capable to handle.

*Example 3.* Consider the model  $E = \{e_1, e_2, e_3\} = \{y_1 = x, y_2 = \dot{x}, y_3 = x^2\}$ . It is obviously impossible to eliminate  $\dot{x}$  in  $e_2$  if differentiation of any equation is forbidden. In general, all derivatives of  $E$  have to be considered. If  $E^{(i)}$  denote the set of the  $i$ :th time derivative of each element, the equation set generally considered is  $\bigcup_{i=0}^{\infty} E^{(i)}$ .

Even though  $\text{var}_{X_u}(e_1) = \text{var}_{X_u}(e_3) = \{x\}$  the derivatives of  $e_1$  and  $e_3$  contain different sets of variables, because  $\text{var}_{X_u}(\dot{e}_1) = \{\dot{x}\} \neq \text{var}_{X_u}(\dot{e}_3) = \{x, \dot{x}\}$ . Since  $x$  is linearly contained in  $e_1$ , the variable  $x$  in  $\dot{e}_1$  disappears. Knowledge about which of the variables that are contained linearly in an equation determines the set of variables in the differentiated equation completely.

For all natural numbers  $j$ ,  $y_1^{(j+1)} - y_2^{(j)} = 0$  is a consistency relation. Most of these consistency relations contain high orders of derivatives of  $y_1$  and  $y_2$ . The derivatives of known variables are in general not known, but they can usually be estimated. The higher order of derivative, the more difficult it is to estimate the derivative. Thus it is reasonable to make a limitation  $m(y)$  for variable  $y$  of the order of derivative that can be considered as possible to estimate. Derivatives up to  $m(y)$  are then considered to be known and higher derivatives belong to  $X_u$ .

To summarize the example, Algorithm 2 must be capable of differentiating equations. To produce a correct structural representation of differentiated equations, the algorithm must take linearly contained variables into account. Further, it has to handle the limitation  $m(y)$  for each  $y \in Y$ .  $\diamond$

Algorithm 2 consists of two parts. The first part is a modification of Pantelides' algorithm (Pantelides,

1988). Let  $M = \bigcup_{i=1}^n \bigcup_{j=0}^{\alpha_i} \{e_i^{(j)}\}$ , then  $\alpha_i$  is the highest number of differentiations in  $M$  of equation  $i$ . Then  $M$  is a differentiated model of  $M_{orig} = \bigcup_{i=1}^n \{e_i\}$ . Let  $\{e_i^{(\alpha_i)} | 1 \leq i \leq n\}$  be the set of most differentiated equations in  $M$ . The highest derivative of a non-differentiated variable  $x$  in the model  $M$  is defined as  $\max(\{i | x^{(i)} \in \text{var}_{X_u}(M)\})$ .

Pantelides' algorithm differentiates equation subsets, so that the original equations together with the differentiated equations have a *complete matching* (Harary, 1969) of the most differentiated equations into the unknown variables with the highest derivatives.

The modification of Pantelides' algorithm is that derivatives of known variables, higher or equal to  $m(y)$ , are also allowed to be included in the matching.

*Algorithm 2.*

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

- (1) Apply the modified Pantelides' algorithm to  $M_{orig}$  and the limits  $m(y)$ . The output is the number of times each equation must be differentiated to find all MSS sets.
- (2) Differentiate the equations in  $M_{orig}$  the number of times suggested in step (1) and use the description of which variables that are linearly contained, to get the correct structural description of the differentiated structural model denoted  $M_{diff}$ .

Output:  $M_{diff}$ .

It is critical that step (1) in Algorithm 2 will terminate, i.e. no equation should be differentiated an infinite number of times. In Pantelides (1988) the condition when the algorithm terminates is stated. This condition can be written as the structural property (3). Since the model  $M_{orig}$  has this property according to Assumption 1 the algorithm will terminate.

Let now  $MSS(M)$  denote the set of MSS sets found in equations  $M$  and  $MSS_{all}(M) = MSS(\bigcup_{i=0}^{\infty} M^{(i)})$ . Then it is possible to state the following theorem proven in (Krysander and Nyberg, 2002).

*Theorem 2.* If Assumption 1 is satisfied and for each  $y \in \overline{\text{var}}_Y(M_{orig})$ ,  $m(y) < \infty$ , then

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

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

*Example 4.* The following example is a continuation of Example 1 with the structural model shown in (1). Let  $m(u) = m(y_f) = 1$  and  $m(y_h) = 0$ . Assume that all faults are zero, i.e. the system is fault free. The equation  $e_6$  contains only a fault. Since all faults are at the moment assumed to be zero, then  $e_6$  is not considered. Further, assume that no variable is

linearly contained in any equation. Then no variable will disappear in the differentiation. The structural model  $M_{diff}$  obtained from Algorithm 2 is

equation	unknown					fault				known			
	$F_1$	$F_2$	$\dot{F}_2$	$h$	$\dot{h}$	$f_u$	$f_h$	$f_f$	$\dot{f}_f$	$u$	$y_h$	$y_f$	$\dot{y}_f$
$e_1$	X					X				X			
$e_2$	X	X			X								
$e_3$				X		X					X		
$e_4$		X		X									
$\dot{e}_4$		X	X	X	X								
$e_5$		X					X					X	
$\dot{e}_5$		X	X				X	X			X	X	X

(5)

## 4.2 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 fulfill the following condition are removed. The condition is that the equation includes a variable in a subset of unknown variables that do not fulfill (4) where  $H = M_{diff}$ . This can be done with Canonical Decomposition (Cassar and Staroswiecki, 1997).

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  $1 + |X| = |\text{equ}_{M_{diff}}(X)|$ , then all equations in  $\text{equ}_{M_{diff}}(X)$  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  $\text{equ}_{M_{diff}}(X)$  has to include all equations in  $\text{equ}_{M_{diff}}(X)$ . The idea is to find these sets. Then it is possible to eliminate internal variables, in the previous discussion denoted  $X$ , in these sets. Every set is replaced with one new equation.

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. However, with this strategy larger sets than two equations will also be found, since the algorithm can merge previously found sets. 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  $M_{simp}$ .

The next theorem from (Krysander and Nyberg, 2002) ensures that no MSS set is lost in the simplification step.

*Theorem 3.*  $MSS(M_{diff}) = MSS(M_{simp})$

Consider again Example 4 and the output (5) from the differentiation step. No equations can be removed in the first simplification step.

The second 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  $\{\dot{e}_4, \dot{e}_5\}$  and  $\dot{h}$  in the equations produced by  $\{e_1, e_2\}$  and  $\{\dot{e}_4, \dot{e}_5\}$ . This makes one group of  $\{e_1, e_2, \dot{e}_4, \dot{e}_5\}$ . This search made simplifications and therefore the search is performed once more. The second time no simplifications have been done and the simplification step is therefore complete. The remaining system is

equation	unknown		fault				known			
	$F_2$	$h$	$f_u$	$f_h$	$f_f$	$\dot{f}_f$	$u$	$y_h$	$y_f$	$\dot{y}_f$
$\{e_1, e_2, \dot{e}_4, \dot{e}_5\}$	X	X	X	X	X	X	X	X	X	X
$e_3$		X		X				X		
$e_4$	X	X								
$e_5$	X				X			X		

(6)

#### 4.3 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  $\{e_1, \dots, e_n\}$ . Let  $M_k = \{e_k, \dots, e_n\}$  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  $M_{alg3}$ . Then the following algorithm finds all MSS sets in  $M_{simp}$ .

*Algorithm 3.*

Input: The model  $M_{simp}$ .

- (1) Set  $k = 1$  and  $M_{alg3} = \emptyset$ .
- (2) Choose equation  $e_k$ . Let  $E = \{e_k\}$  and  $X = \emptyset$ .
- (3) Find all MSS sets that are subsets of  $M_k$  and include equation  $e_k$ .
  - (a) Let  $\tilde{X} = var_{X_u}(E) \setminus X$  be the unmatched variables.
  - (b) If  $\tilde{X} = \emptyset$ , then  $E$  is an MSS set. Insert  $E$  into  $M_{alg3}$ .
  - (c) Else take a remaining variable  $\tilde{x} \in \tilde{X}$  and let  $X = X \cup \{\tilde{x}\}$ . Let  $\tilde{E} = equ_{M_k \setminus E}(\tilde{x})$  be the remaining equations. For all equations  $e$  in  $\tilde{E}$  let  $E = E \cup \{e\}$  and goto (a).
- (4) If  $k < n$  set  $k = k + 1$  and goto number (2).

Output: The set of MSS sets found, i.e.  $M_{alg3}$ .

Algorithm 3 finds all MSS sets in  $M_{orig}$  according to the next theorem proven in (Krysander and Nyberg, 2002).

*Theorem 4.*  $M_{alg3} = MSS(M_{simp})$

The following small example with five equations shows how the algorithm works.

	$x_1$	$x_2$	$x_3$
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

	2		3			2			
	2	5	<b>5</b>	2	<b>2</b>	3	<b>3</b>	<b>5</b>	
3	3	3	<b>3</b>	4	4	<b>4</b>	4	<b>4</b>	<b>4</b>
1	1	1	1	<b>1</b>	1	1	<b>1</b>	1	<b>1</b>
				<b>4</b>					
	4		3	<b>3</b>				5	
	3	3	5	5	<b>5</b>	4	4		
2	2	2	2	2	<b>2</b>	3	3	3	4
									5

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.

#### 4.4 Analyzing Diagnosability

In many cases in real applications, it is difficult to draw conclusions when a consistency relation is fulfilled. Therefore it is assumed that conclusions are drawn only when consistency relations are not fulfilled.

Now follows the continuation of the example in (6). The 4 MSS sets in (6) are shown in the left column in Figure 1 (a). The matrix in this figure is the incidence matrix of the MSS sets in (6). If any equation in the MSS set  $i$  include fault  $j$ , the element  $(i, j)$  of the incidence matrix is equal to  $X$ . However, an  $X$  in position  $(i, j)$  is no guarantee for fault  $j$  to make a consistency relation derived from MSS set  $i$  invalid. The derivatives of the faults are omitted in Figure 1. The third MSS set in Figure 1 (a) could contain  $f_u$  and  $f_f$ , but it is impossible that it could contain  $f_h$ , since  $f_h$  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 set shows how the consistency relations react on the faults, but it is more interesting to see which faults that can be explained by other faults. The *fault matrix* shows the maximum isolation and detection capability of the diagnosis system. If fault  $j$  is sensitive to at least all consistency relations as fault  $i$  is sensitive to, then element  $(i, j)$  of fault 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  $f_j$ .

The fault matrix in the example is shown in Figure 1 (b). Consider the first row of the fault matrix. Suppose that fault  $f_u$  is present. Then, the first three consistency relations are not satisfied in an ideal case. This means that  $f_u$  certainly can explain fault  $f_u$ , but also  $f_f$  can explain fault  $f_u$ . Fault  $f_h$  cannot explain fault  $f_u$ , since if  $f_h$  is present, the third consistency relation is satisfied. Note that the fault matrix is not symmetric. For example fault  $f_f$  can explain fault  $f_u$  but the opposite is not true. The fault matrix can more easily be analyzed after Dulmage-Mendelsohn permutations (Meurant, 1999). This algorithm returns a *maximal matching* (Harary, 1969) which is in block upper-triangular form. The diagonal blocks corresponds to strong Hall components of the adjacency graph of the fault matrix. The interpretation is that faults in a diagonal block can never be distinguished with that diagnosis system. In the small example in Figure 1 (b), the

MSS	$f_u$	$f_h$	$f_f$
$\{e_1, e_2, e_3, e_4, e_4, e_5\}$	X	X	X
$\{e_1, e_2, e_3, e_4, e_5, e_5\}$	X	X	X
$\{e_1, e_2, e_4, e_4, e_5, e_5\}$	X	X	X
$\{e_3, e_4, e_5\}$		X	X

(a)

present fault	$f_u$	$f_h$	$f_f$
$f_u$	X		X
$f_h$		X	X
$f_f$			X

(b)

Fig. 1. The incidence matrix of MSS sets is shown in (a). The fault matrix of (a) is shown in (b).

same matrix is returned after Dulmage-Mendelsohn permutations, which usually is not the case. The diagonal blocks are the  $1 \times 1$  diagonal elements.

#### 4.5 Decoupling faults

Suppose that the element  $(i, j)$  of the fault matrix is equal to  $X$  for some  $i \neq j$ . 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 step (1) 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.

In the example in Figure 1, the fault matrix shows that  $f_u$  and  $f_h$  can not be isolated from  $f_f$ . The problem is that there is no consistency relation that decouple fault  $f_f$ . But there could be one if  $f_f$  is eliminated. The fault  $f_f$  is moved from the faults  $F$  to the unknown variables  $X_u$ . The procedure starts all over from the step (1) in Algorithm 1. The result is a new MSS set in which  $f_f$  is decoupled. This gives a possibility to detect and isolate all faults.

#### 4.6 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 the set of most robust MSS sets but with the same diagnosis capability as the set of all MSS sets.

Start to sort the MSS sets in an ascending order of complexity. The complexity measure is here the number of equations, even though more informative measures are also a possibility. The MSS sets are examined in the rearranged order. If an MSS set increase the diagnosability, then select the MSS set. The diagnosability is increased if some fault becomes detectable or some fault  $i$  can be isolated from some other fault  $j$ . This means that for each detection of a fault and for each isolation between two faults, the smallest MSS sets with this diagnosis ability will be one of the chosen MSS sets. In this way the final output from Algorithm 1 will be the most robust set of MSS sets with highest possible diagnosis capability.

## 5. 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 diagnosis capability. The method is based on

graph theoretical reasoning about the structure of the model. It is assumed that a condition on algebraic interdependency is fulfilled, i.e. Assumption 2. 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 it 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.

In this paper consistency relations are used to validate the MSS sets. However, there are also other methods that can be used to validate the MSS sets, e.g. observers. Finally in (Krysander and Nyberg, 2002) the method has been applied to a large nonlinear industrial example, a paper plant. The algorithm successfully manage to derive a small set submodels. In spite of the complexity of this process, most of these submodels can be used to derive consistency relations with high diagnosis capability.

## 6. REFERENCES

- Carpanzano, E. and C. Maffezzoni (1998). Symbolic manipulation techniques for model simplification in object-oriented modeling of large scale continuous systems. *Mathematics and Computers in Simulations* (48), 133–150.
- Cassar, J. P. and M. Staroswiecki (1997). A structural approach for the design of failure detection and identification systems.. In: *IFAC Control of Industrial Systems*.
- Harary, F. (1969). *Graph theory*. Addison-Wesley publishing company. ISBN 0-201-41033-8.
- Krysander, M. and M. Nyberg (2002). Structural analysis for fault diagnosis of DAE systems utilizing graph theory and MSS sets. Technical Report LiTH-ISY-R-2410. Dept. of Electrical Engineering Linköping University. URL:<http://www.vehicular.isy.liu.se/Publications/>
- Meurant, G. (1999). *Computer Solution of Large Linear Systems*. Elsevier Science B. V.. ISBN 0-444-50169-X.
- Pantelides, C C. (1988). The consistent initialization of differential-algebraic systems. *SIAM J. SCI. STAT. COMPUT.* **9**(2), 213–231.
- Pulido, B. and C. Alonso (2000). An alternative approach to dependency-recording engines in consistency-based diagnosis. In: *Lecture Notes in Artificial Intelligence*. Vol. 1904. Artificial Intelligence: Methodology, Systems, and Applications. 9th International Conference, AIMS 2000.. Springer-Verlag, Berlin, Germany; 2000. pp. 111–121.
- Travé-Massuyès, L., T. Escobet and R. Milne (2001). Model-based diagnosability and sensor placement. application to a frame 6 gas turbine subsystem. In: *DX01 twelfth international workshop on principals of diagnosis* (D. T. Dupre' S. McIlraith, Ed.). pp. 205–212.