Structural Analysis utilizing MSS Sets with Application to a Paper Plant

Mattias Krysander 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 the 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. The algorithm is applied to a large nonlinear industrial example, 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.

1 Introduction

When designing model-based fault-diagnosis systems, using the principle of consistency based diagnosis [5, 11, 6], a crucial step is the conflict recognition. As shown in [3], conflict recognition can be achieved by using pre-computed consistency relations (also called e.g. *analytical redundancy relations* or *parity relations*). 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 only needs as input 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 [10] 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 [2] and an extension to this work considering sensor placement is found in [12].

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 then in Section 5 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 [2] 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* [4, 1]. 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 measured water level in the tank y_h, and the measured flow from*

the tank y_f. One fault denoted f_i is assumed to be associated with 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_{yh}	f_{yf}	$\dot{f}_f$	u	y_h	y_f
e_1	X				X				X		
e_2	X	X		X							
e_3			X			X				X	
e_4		X	X								
e_5	X						X				X
e_6								X			

$$(1)$$

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 the flow measurement, and e_6 a fault model for the flow measurement fault f_{yf}.

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 **x** and **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 $c(\mathbf{y}) = 0$ is a* consistency relation for *the equations $E(\mathbf{x}, \mathbf{y})$ iff*

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

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

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.

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 exactly 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 $var_X(E) = \{x \in X | \exists e \in E : e \text{ contains } x\}$ and $equ_E(X) = \{e \in E | \exists x \in X : e \text{ contains } x\}$. 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 $\overline{var}_X(E) = \{\text{undifferentiated } x | \exists i(x^{(i)} \in var_X(E))\}$, 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 introduced to ensure that the model becomes finitely differentiated in Section 4.1.

Assumption 1 *The model M_{orig} has the property*

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

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*

$$\forall X' \subseteq var_{X_u}(H), X' \neq \varnothing : |X'| < |equ_H(X')| \qquad (4)$$

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 then 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 [9].

Definition 2 (Structurally Singular) *A finite set of equations E is* structurally singular *with respect to the set of variables X if $|E| > |var_X(E)|$.*

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 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 $c(\mathbf{y}) = 0$ for $H(\mathbf{x}, \mathbf{y})$ where $|H| < \infty$ iff $H = \bigcup_i E_i$ where for each i, E_i is an MSS set.*

For a proof, see [7].

4 Algorithm for finding and selecting MSS sets

The objective is to find all MSS sets in a differentiated version of the model M_{orig} and then choose a small subset of these MSS sets with the same diagnosability as the full set of MSS sets. 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.*

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.

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 of handling.

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 $var_{X_u}(e_1) = var_{X_u}(e_3) = \{x\}$ the derivatives of e_1 and e_3 contain different sets of variables, because $var_{X_u}(\dot{e}_1) = \{\dot{x}\} \neq var_{X_u}(\dot{e}_3) = \{x, \dot{x}\}$. 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 $y \in Y$.

Algorithm 2 consists of two parts. The first part is a modification of Pantelides' algorithm [9]. Let $M = \bigcup_{i=1}^{n} \bigcup_{j=0}^{\alpha_i} \{e_i^{(j)}\}$, then α_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 $\max(\{i | x^{(i)} \in var_{X_u}(M)\})$.

Pantelides' algorithm differentiates equation subsets, so that the original equations together with the differentiated equations have a *complete matching* [4] 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 $y \in \overline{var}_Y(M_{orig})$, $m(y) < \infty$.

(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 terminates, 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 [7].

Theorem 2 *If Assumption 1 is satisfied and for each $y \in \overline{var}_Y(M_{orig})$, $m(y) < \infty$, 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 $m(y_h) = 0$. According to Algorithm 1 the first iteration uses the fault free model, i.e. all faults are zero. 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_{yh}	f_{yf}	$\dot f_f$	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

$$(5)$$

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 include any variable that is impossible to eliminate, are removed. This can be done with Canonical Decomposition [2].

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 $1 + |X| = |equ_{M_{diff}}(X)|$, then all equations in $equ_{M_{diff}}(X)$ 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 $equ_{M_{diff}}(X)$ has to include all equations in $equ_{M_{diff}}(X)$. 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.

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 [7]. 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_{yh}	f_{yf}	$\dot f_f$	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
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 $\{e_1, \cdots, e_n\}$. 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 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 $M_{alg3} = \varnothing$.*
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.*

   (a) *Let $\tilde X = var_{X_u}(E) \backslash X$ be the unmatched variables.*

   (b) *If $\tilde X = \varnothing$, 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 $\tilde E = equ_{M_k \backslash E}(\tilde x)$ 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 [7].

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   5        2     2   3   3   5
    3 3   3   3    4   4     4   4   4   4
  1 1 1   1   1    1   1     1   1   1   1
              4
      4       3    3             5
    3 3   5   5    5       4     4
  2 2 2   2   2    2   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

When the MSS sets are found, the next step is to analyze their diagnosability. The continuation of the example in (6) will be used to illustrate how this analysis is done. The 4 MSS sets that can be found 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. Note that an X in position (i, j) is no guarantee for fault j to appear in the MSS set i. For an example of the interpretation of an incidence matrix, consider the third MSS set in Figure 1 (a). 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. For simplicity, the derivatives of the faults are omitted in Figure 1.

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 faults that can be explained by other faults. A *fault 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 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 corresponding to the incidence matrix in Figure 1 (a) 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 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 fault matrix is not symmetric. For example fault f_{yf} can explain fault f_u but the opposite is not true. The fault matrix can more easily be analyzed after Dulmage-Mendelsohn permutations [8]. This algorithm returns a *maximal matching* [4] 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 same matrix is returned after Dulmage-Mendelsohn permutations, which usually is not the case. The diagonal blocks are the 1×1 diagonal elements.

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_{yh} can not be isolated from f_{yf}. The problem is that there is no MSS set that decouple fault f_{yf}. But there could be one if f_{yf} is eliminated. The fault f_{yf} 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_{yf} 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 Industrial example: A part of a paper plant

This example is a stock preparation and broke treatment system of a paper plant 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 2.
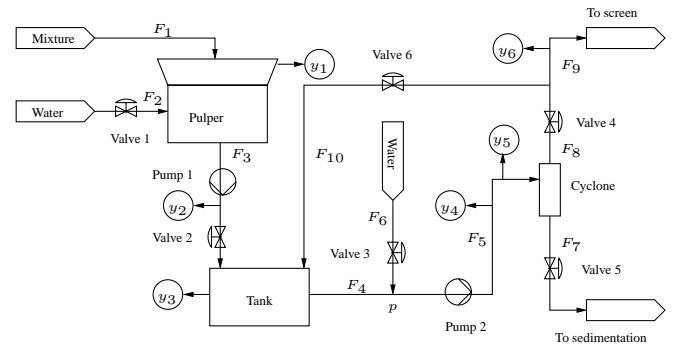
MSS	f_u	f_{yh}	f_{yf}
$\{e_1, e_2, e_3, e_4, \dot{e}_4, \dot{e}_5\}$	X	X	X
$\{e_1, e_2, e_3, \dot{e}_4, e_5, \dot{e}_5\}$	X	X	X
$\{e_1, e_2, e_4, \dot{e}_4, e_5, \dot{e}_5\}$	X		X
$\{e_3, e_4, e_5\}$		X	X

(a)

present fault	interpreted fault		
	f_u	f_{yh}	f_{yf}
f_u	X		X
f_{yh}		X	X
f_{yf}			X

(b)

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

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

Figure 2. A stock preparation and broke treatment system of a paper plant.

5.1 System Description

Most parts of the system are nonlinear and it is only the tank and the pulper that are considered to be dynamic. The model has shown to compare well to real measured data. Because of space considerations, the details of the model are omitted, but can be found in [7]. 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. 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}, 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. The structural model for these equations can be viewed in the first 29 rows in the matrices in Figure 3.

5.2 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 3. 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 3.
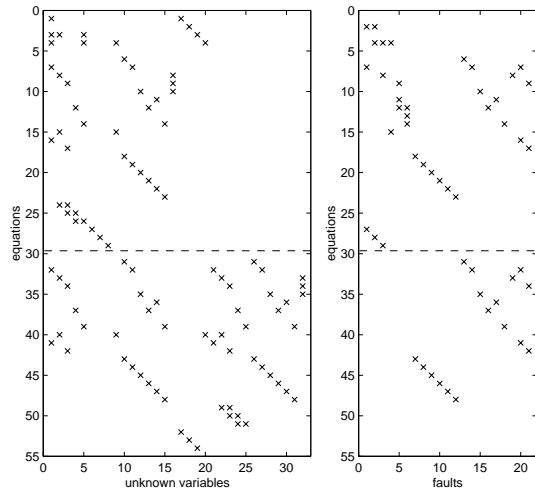

Figure 3. Structural model of the stock preparation and broke treatment system.

5.3 Simplifying the Model

In the first step of simplification applied to the left matrix in Figure 3, 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 4 (a). Note the simplification of the model by comparing Figure 3 and Figure 4 (a).
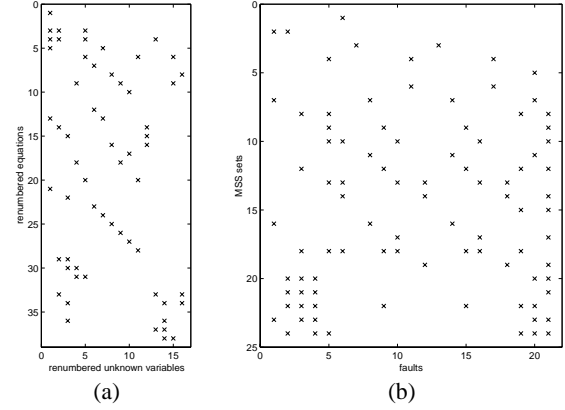

(a)   (b)

Figure 4. The simplified structural model is shown in (a). The incidence matrix of the MSS sets is shown in (b)

5.4 Finding MSS sets

Algorithm 3 is then applied to the simplified model. The algorithm returns 35770 MSS sets that are contained in the simplified model. The largest MSS set consists of 24 equations.

5.5 Analyzing Diagnosability

The two different fault matrices are seen in Figure 5. 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. To give an example of how elimination of faults is done, the attention is focused on isolating faults 4, 8, and 14.

5.6 Decoupling faults

Considering Figure 5, 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.

5.7 Selecting a subset of MSS sets

The 24 chosen MSS sets are

$$
\begin{array}{c|l}
MSS & \\
\hline
1 & 13 \\
2 & 2\ \ 53 \\
3 & 6\ \ 18 \\
4 & 11\ 22 \\
5 & 1\ \ 16\ 52 \\
6 & 22\ 36\ 47 \\
7 & 7\ \ 16\ 19 \\
8 & 8\ \ 9\ \ 17\ 24 \\
9 & 9\ \ 10\ 17\ 20 \\
10 & 12\ 17\ 21\ 25 \\
11 & 16\ 19\ 32\ 41\ 44 \\
12 & 8\ \ 10\ 17\ 20\ 24 \\
13 & 12\ 14\ 21\ 23\ 26 \\
14 & 14\ 17\ 23\ 25\ 26 \\
15 & 17\ 24\ 33\ 34\ 42\ 49 \\
26 & 7\ \ 16\ 19\ 32\ 41\ 44 \\
17 & 17\ 21\ 25\ 37\ 42\ 46\ 50 \\
18 & 8\ \ 10\ 12\ 20\ 21\ 24\ 25 \\
19 & 17\ 23\ 25\ 26\ 39\ 42\ 48\ 50\ 51 \\
20 & 3\ \ 4\ \ 15\ 16\ 17\ 24\ 40\ 42\ 49\ 54 \\
21 & 1\ \ 3\ \ 4\ \ 15\ 17\ 24\ 40\ 42\ 49\ 52\ 54 \\
22 & 3\ \ 4\ \ 8\ \ 10\ 15\ 16\ 20\ 33\ 35\ 40\ 45\ 54 \\
23 & 2\ \ 3\ \ 4\ \ 15\ 16\ 17\ 24\ 40\ 42\ 49\ 53\ 54 \\
24 & 3\ \ 4\ \ 8\ \ 9\ \ 15\ 16\ 17\ 24\ 40\ 42\ 49\ 54 \\
\end{array}
\tag{7}
$$

From these sets and the structural model in Figure 3 the incidence matrix in Figure 4 (b) is obtained.

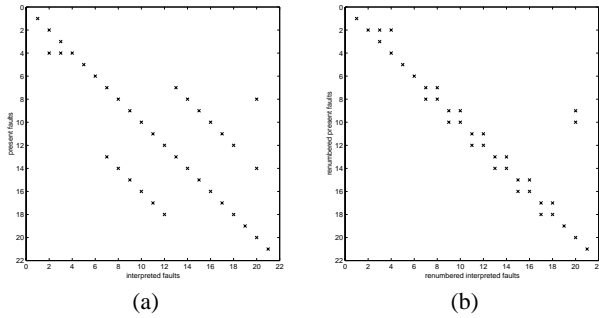(a)   (b)

Figure 5. These matrices are the fault matrices before (a) and after (b) the Dulmage-Mendelsohn permutation.

5.8 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 [7].

6 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 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 plant. 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 diagnosis capability was obtained.

REFERENCES

[1] 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*, (48), 133–150, (1998).

[2] J. P. Cassar and M. Staroswiecki, 'A structural approach for the design of failure detection and identification systems.', in *IFAC Control of Industrial Systems*, (1997).

[3] M-O. Cordier, P. Dague, M. Dumas, F. Lévy, J. Montmain, M. Staroswiecki, and L. Travé-Massuyés, 'AI and automatic control approaches of model-based diagnosis: Links and underlying hypotheses', in *4th IFAC Symosium on Fault Detection Supervision and Safety for Technical Processes*, ed., A.M.Edelmayer, volume 1, pp. 274–279, (2000).

[4] F. Harary, *Graph theory*, Addison-Wesley publishing company, ISBN 0-201-41033-8, 1969.

[5] J. De Kleer, A. K. Mackworth, and R. Reiter, 'Characterizing diagnoses and systems', *Artificial Intelligence*, (1992).

[6] J. De Kleer and B.C. Williams, 'Diagnosing multiple faults', *Artificial Intelligence*, (1987).

[7] M. Krysander and M. Nyberg, '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, (2002). URL:http://www.vehicular.isy.liu.se/Publications/.

[8] G. Meurant, *Computer Solution of Large Linear Systems*, Elsevier Science B. V., ISBN 0-444-50169-X, 1999.

[9] C C. Pantelides, 'The consistent initialization of differential-algebraic systems', *SIAM J. SCI. STAT. COMPUT.*, **9**(2), 213–231, (1988).

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

[11] R. Reiter, 'A theory of diagnosis from first principles', *Artificial Intelligence*, (1987).

[12] 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*, ed., D. T. Dupre' S. Mcilraith, pp. 205–212, (2001).