Structural Analysis for Fault Diagnosis of DAE Systems Utilizing Graph Theory and MSS Sets

Department of Electrical Engineering, Linköping University
SE-581 83 Linköping, Sweden
Phone: +46 13 282198, Email:matkr@isy.liu.se

Mattias Krysander and Mattias Nyberg

Report: LiTH-ISY-R-2410

Linköping, May 21, 2002.

Abstract

When designing model-based fault-diagnostic systems, the use of *consistency relations* (also called e.g. *parity relations*) is a common choice. Different consistency relations are sensitive to different subsets of faults, and thereby isolation can be achieved. This report 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.

Contents

Chapter 1

Structural Analysis

1.1 Introduction

When designing model-based fault-diagnostic systems, using the principle of consistency based diagnosis [7, 12, 8], a crucial step is the conflict recognition. As shown in [4], 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 report 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 report 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 report 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 diagnostic-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 been studied in other works dealing with fault diagnosis. In [11] 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 [3] and an extension to this work considering sensor placement is found in [13].

In Section 1.2 and 1.3, structural models and their usefulness in fault diagnosis are discussed. Then in Chapter 2, a complete description of the algorithm is given. The algorithm is then in Chapter 3 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.

1.2 Structural Models

The behavior of a system is described with a model. Usually the model is a set of equations. A structural model [3] contains only the information of which variables that are contained in each equation. Let $\mathsf{M}_{\mathsf{orig}}$ 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. To gain isolability it is sometimes necessary to decouple faults, i.e. to eliminate faults. A fault is decoupled by considering the fault variable as an unknown variable, i.e. include the fault 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* [2, 6]. 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 1.1 A simple example, showed in Figure 1.1, 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_{yh}	f_{yf}	$\dot{f}_{yf}$	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			

(1.1)

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_{yf}.

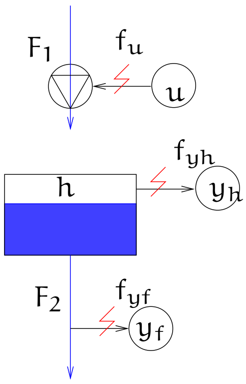

Figure 1.1: A small system with a pump, pumping water into a tank.

1.3 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, i.e. detectability and 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* is introduced that explicitly points out the submodel considered. Before consistency relation for a set of equations 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 $E(\mathbf{x}, \mathbf{y})$ denotes an equation set E that depends on variables contained in X_u and Y.

Definition 1.1 (Consistency Relation for E). A scalar equation $c(\mathbf{y}) = 0$ is a *consistency relation for* the equations $E(\mathbf{x}, \mathbf{y})$ iff

$$\exists \mathbf{x} E(\mathbf{x}, \mathbf{y}) \Leftrightarrow c(\mathbf{y}) = 0 \tag{1.2}$$

and there is no proper subset of E that has property (1.2).

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 1.1 differ from the common definition of consistency relation in two ways, the left implication in (1.2) and that there is no proper subset of E that has property (1.2). Refer to the latter as the minimality condition in Definition 1.1. The following example shows the importance of the left implication in (1.2).

Example 1.2 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 (1.2) 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 next example explains the reason why it is important that there is no proper subset of E that has property 1.2 in Definition 1.1.

Example 1.3 Let the model be $M = \{y_1 = x_1, y_2 = x_1, y_3 = x_2, x_1 = x_2\}$. Each equation explains a behavior of a part of the modeled system. Now, should there be a consistency relation for $E = \{y_1 = x_1, y_2 = x_1, y_3 = x_2\}$? It is easy to find a relation that satisfy (1.2), e.g. $y_1 - y_2 = 0$. To see why it is not good to call $y_1 - y_2 = 0$ a consistency relation for E, we suppose that $y_1 - y_2 \neq 0$ has been observed.

$$\begin{aligned} y_1 - y_2 \neq 0 \Leftrightarrow \neg\exists x_1, x_2(y_1 = x_1 \wedge y_2 = x_1 \wedge y_3 = x_2) \Leftrightarrow \\ \neg\exists x_1(y_1 = x_1 \wedge y_2 = x_1) \vee \neg\exists x_2(y_3 = x_2) \end{aligned}$$

Note that $\neg\exists x_2(y_3 = x_2)$ is false regardless of y_3 and therefore

$$y_1 - y_2 \neq 0 \Leftrightarrow \forall x_1(y_1 \neq x_1 \vee y_2 \neq x_1) \Rightarrow y_1 \neq x_1 \vee y_2 \neq x_1$$

These calculations imply that the consistency relation $y_1 - y_2 = 0$ can never infer anything about $y_3 = x_2$. The minimality condition in Definition 1.1 implies that e.g. $y_1 - y_2 = 0$ is a consistency relation only for $\{y_1 = x_1, y_2 = x_1\}$. This is the minimal set of equations that must be used to derive $y_1 - y_2 = 0$. Accepting the minimality condition in Definition 1.1 implies that there is no consistency relation for E at all.

1.3.1 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 $var_X(E) = \{x \in X | \exists e \in E : e \text{ contains } x\}$ 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 a technical requirement used to ensure that Algorithm 2 later described in Section 2.1 will terminate.

Assumption 1.1. The model M_{orig} has the property

$$\forall E \subseteq M_{orig} : |E| \leq |\overline{var}_{X_u \cup Y}(E)|. \tag{1.3}$$

The meaning of condition (1.3) is that each subset of equations include more or equally many different variables, considering derivatives as the same variable. If condition (1.1) is not fulfilled and there are no redundant equations, the model would normally be inconsistent.

The first example illustrate that there are models that does not satisfy Assumption 1.1. Furthermore, the example shows two methods that can be applied to transform a model into one or several models that satisfy Assumption 1.1.

Example 1.4 Consider the model

$$\begin{aligned} e_1 :&\quad \dot{x} = 0 \\ e_2 :&\quad x + \ddot{x} = 0 \\ e_3 :&\quad y = x. \end{aligned}$$

Since for example $|\overline{var}_{X_u \cup Y}(\{e_1, e_2\})| = 1 < 2 = |\{e_1, e_2\}|$ does not fulfill the inequality (1.3), the model does not satisfy Assumption 1.1. However, e_1 and e_2 can be merged into one equation. Differentiating e_1

and substitute the result in e_2, will produce $x = 0$. This expression is equivalent to the equations e_1 and e_2. In this way, we have transformed a model that does not satisfy Assumption 1.1 to an equivalent model that does satisfy Assumption 1.1.

A second approach to transform a model, when Assumption 1.1 fails to hold, is to remove equations that make the assumption to fail.

In this example it is possible to consider $\{e_1, e_3\}$ in the first analysis and then $\{e_2, e_3\}$. By using this method we find the consistency relations $\dot{y} = 0$ and $y + \ddot{y} = 0$ for the two subsets respectively. These two consistency relations are equivalent to $y = 0$.

The next example shows when the model is inconsistent and Assumption 1.1 is not fulfilled.

Example 1.5 A slightly modified model is

$$\begin{aligned} e_1 &: \quad \dot{x} = 1 \\ e_2 &: \quad x + \ddot{x} = 0 \\ e_3 &: \quad y = x. \end{aligned}$$

As before, in the previous example, the model does not satisfy Assumption 1.1. In this case x is over determined. The equations e_1 and e_2 are inconsistent. The equation e_1 implies that x is strictly increasing, while e_2 expresses an oscillation of x. Hence a model that does not satisfy Assumption 1.1 can be inconsistent.

From now on all models considered are assumed to satisfy Assumption 1.1.

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 1.2. There exists a consistency relation $c(\mathbf{y}) = 0$ for H iff

$$\forall X' \subseteq var_{X_u}(H), X' \neq \varnothing : |X'| < |equ_H(X')| \tag{1.4}$$

According to Assumption 1.2, 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 1.1 and 1.2 are often fulfilled. For example all subsets of equations found in the industrial example in the end of the report satisfy Assumption 1.2. Even though the "only if" direction of Assumption 1.2 is difficult to validate in an application, the results of the report can still be used to produce a lower bound of the actual detection and isolation capability.

If all subsets of the model fulfill Assumption 1.2, the structural analysis will find all subsets that can be used to find consistency relations.

Still there are sometimes inconsistencies that rely on smaller submodels than the structural analysis finds.

Example 1.6 The model

$$y = x^2 \tag{1.5}$$

satisfy Assumption 1.1 and Assumption 1.2. Nevertheless consistency for $y = x^2$ can be checked using $y \geq 0$. It is not necessary to eliminate x in this example, because (1.5) forces $y \in \mathbb{R}^+ \cup \{0\}$.

The conclusion is that the structural method in this report handle the cases when elimination of all unknown variables is possible.

Still there are inconsistencies that can not be revealed when Assumption 1.2 is not fulfilled. Here follows an illustrating example.

Example 1.7 The model

$$\begin{aligned} e_1: &\quad x_1 + x_2 = y_1 \\ e_2: &\quad x_1 + x_2 = y_2 \\ e_3: &\quad x_1 - x_2 = 0, \end{aligned} \tag{1.6}$$

satisfies Assumption 1.1 and (1.4). However, it is not possible to make a consistency relation for $\{e_1, e_2, e_3\}$. Hence, Assumption 1.2 is not fulfilled for $\{e_1, e_2, e_3\}$. The linear dependence of x_1 and x_2 in the first and second equations, makes it impossible to validate the third equation.

Furthermore $\{e_1, e_2\}$ does not satisfy (1.4) even though $y_1 - y_2 = 0$ is a consistency relation for $\{e_1, e_2\}$.

The structural analysis will find the submodel that fulfill property (1.4) in this case $\{e_1, e_2, e_3\}$ and when the consistency relation is to be calculated, it will be clear that the consistency relation found is a consistency relation only for $\{e_1, e_2\}$.

If the model consists of $\{e_1, e_2\}$, then no consistency relations are found with structural analysis.

This model does not satisfy Assumption 1.2 because of the linear dependence in e_1 and e_2.

Finally, a theorem is presented that guarantee that all sets with the structural property (1.4) have to include known variables, if the Assumptions 1.1 and 1.2 are fulfilled.

Theorem 1.1. Let $H \subseteq M_{orig}$, where M_{orig} satisfies Assumption 1.1. If H satisfies Assumption 1.2, then $var_Y(H) \neq \varnothing$.

Proof. From the fact that H satisfies Assumption 1.2 it follows that

$$|H| = |equ_H(var_{X_u}(H))| > |var_{X_u}(H)| \geq |\overline{var}_{X_u}(H)| \tag{1.7}$$

According to (1.3) it holds that

$$|H| \leq |\overline{var}_{X_u \cup Y}(H)|. \tag{1.8}$$

Suppose that $var_Y(H) = \varnothing$. This implies also that $\overline{var}_Y(H) = \varnothing$ and hence

$$\overline{var}_{X_u \cup Y}(H) = \overline{var}_{X_u}(H) \tag{1.9}$$

Finally (1.7), (1.8) and (1.9) implies a contradiction.
Hence, $var_Y(H) \neq \varnothing$. ■

1.3.2 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 (1.4). To do this, two important structural properties are defined [10].

Definition 1.2 (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 1.3 (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 if nothing else is mentioned. The next theorem tells that it is sufficient and necessary to find all MSS sets to parameterize

structural property	equation sets
structurally singular	$\{e_1, e_2\}$, $\{e_1, e_2, e_3\}$, $\{e_1, e_2, e_4\}$, $\{e_1, e_2, e_3, e_4\}$, $\{e_3, e_4\}$, $\{e_1, e_3, e_4\}$, $\{e_2, e_3, e_4\}$
MSS	$\{e_1, e_2\}$, $\{e_3, e_4\}$
condition (1.4)	$\{e_1, e_2\}$, $\{e_3, e_4\}$, $\{e_1, e_2, e_3, e_4\}$

Table 1.1: Equation sets of (1.10) and their structural properties.

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 1.2. Let $H \subseteq M_{orig}$, where M_{orig} fulfills Assumption 1.1. Further, let H and all MSS sets E_i included in H fulfill Assumption 1.2. Then there exists a consistency relation $c(\mathbf{y}) = 0$ for $H(\mathbf{x}, \mathbf{y})$ where $|H| < \infty$ iff $H = \bigcup_i E_i$.

Before Theorem 1.2 is proven, two example illustrates the importance of this theorem.

Example 1.8 The structural model

equation	x_1	x_2	y_1	y_2
e_1	X		X	
e_2	X			
e_3		X		X
e_4		X		

(1.10)

fulfills Assumption 1.1 and assume that Assumption 1.2 is fulfilled for all submodels. Equation sets of (1.10) and their structural properties are shown in Table 1.1

Note that the sets that satisfy condition (1.4) are these sets that can be used to derive consistency relations according to Assumption 1.2.

Neither all supersets to MSS sets fulfill condition (1.4), nor all structurally singular sets fulfill condition (1.4). Instead Theorem 1.2 state that the union of families of MSS sets are the sets that satisfy condition (1.4).

In this small example is

$$\{e_1, e_2\} \cup \{e_3, e_4\} = \{e_1, e_2, e_3, e_4\}$$

which satisfy condition (1.4).

To prove Theorem 1.2, a couple of definitions and Lemmas are needed. For readers who are not interested in the proofs, it is possible to directly go to Chapter 2.

1.3.3 Some Basic Graph Theoretic Concepts and Results

Structural models can naturally be represented also as bipartite graphs [2, 3]. This view is utilized especially when proving theorems and in Section 2.3.

Let $G = (E, X)$ be a bipartite graph with the nodes partitioned as $E \cup X$. There is an edge $\{e, x\}$ with $e \in E$ and $x \in X$ if and only if $x \in var_X(e)$. A *matching* in G is a subset of edges, such that no two edges share a common node in E or X. A *complete matching* of E into X is a matching in G such that every $x \in X$ is an endpoint of an edge. A matching in G can equally well be a complete matching of X into E. A matching in G that firstly is a complete matching of E into X and secondly is a complete matching of X into E is a *perfect matching* in G [5].

The following theorem is often refereed to as Hall's theorem [6].

Theorem 1.3 (System of Distinct Representatives). Let $V = \{V_1, V_2, \cdots, V_m\}$ be a set of objects and $S = \{S_1, S_2, \cdots, S_n\}$ a set of subsets of V. Then a complete matching of S into V exists iff $\forall S' \subseteq S : |S'| \leq |\bigcup_{S_i \in S'} S_i|$.

Note that Theorem 1.3 can be used in two ways. The following two corollaries are immediate from Theorem 1.3.

Corollary 1.4. There is a complete matching of E into X iff $\forall E' \subseteq E : |E'| \leq |var_X(E')|$.

Corollary 1.5. There is a complete matching of X into E iff $\forall X' \subseteq X : |X'| \leq |equ_E(X')|$.

1.3.4 Theory Towards Proving Theorem 1.2

This section aims to prove Theorem 1.2. The proof is divided into several Lemmas as shown in Figure 1.2.

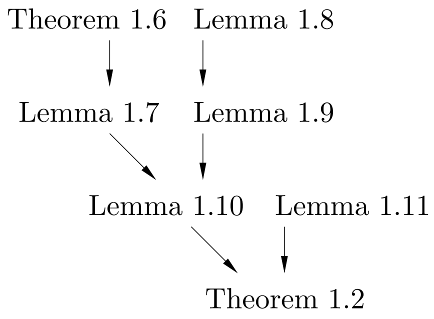

Figure 1.2: Logical dependence of Lemmas and Theorems towards proving Theorem 1.2.

Theorem 1.6. For all $e \in E$ there exists a perfect matching in $(E \backslash \{e\}, var_{X_u}(E))$ if and only if E is an MSS set.

Proof. $\Rightarrow$) From the hypothesis that there exist a perfect matching in $(E \backslash \{e\}, var_{X_u}(E))$ for any $e \in E$, it follows that E is structurally singular, because $|var_{X_u}(E)| = |E \backslash \{e\}| < |E|$.

The set E is also minimal if no subset of E is structurally singular, i.e. $\forall \hat{E} \subset E : |var_{X_u}(\hat{E})| \geq |\hat{E}|$. For each proper subset $\hat{E}$ of E, it is always possible to choose an equation e, such that $\hat{E} \subseteq E \backslash \{e\}$. Since there exists a perfect matching in $(E \backslash \{e\}, var_{X_u}(E))$, according to the hypothesis, it follows that there exists a complete matching of $\hat{E}$ into $var_{X_u}(E)$.

Let $E = \hat{E}$ and $X = var_{X_u}(E)$ in Corollary 1.4, then

$$\forall E' \subseteq \hat{E} : |E'| \leq |var_{var_{X_u}(E)}(E')| = |var_{X_u}(E')|. \tag{1.11}$$

Letting $E' = \hat{E}$, the inequality (1.11) becomes $|\hat{E}| \leq |var_{X_u}(\hat{E})|$. The conclusion is that $\hat{E}$ is not structurally singular and since $\hat{E}$ is an arbitrary chosen proper subset of E, it follows that E is an MSS set.

$\Leftarrow$) Take an arbitrary $e \in E$ and let $E' = E \backslash \{e\}$. It is sufficient to prove that there exist a perfect matching in $(E', var_X(E))$.

From the definition of MSS sets, it follows that $\forall \bar{E} \subset E : |\bar{E}| \leq |var_X(\bar{E})|$. Especially this is true for $E' \subset E$, i.e. $\forall \bar{E} \subseteq E' : |\bar{E}| \leq |var_X(\bar{E})|$. According to Corollary 1.4, there is a complete matching of E' into $var_X(E')$.

Since E is an MSS set and $var_X(E') \subseteq var_X(E)$ it follows that

$$|E'| \leq |var_X(E')| \leq |var_X(E)| < |E| = |E'| + 1 \tag{1.12}$$

This implies that $|E'| = |var_X(E')|$, hence the complete matching is a perfect matching in $(E', var_X(E'))$. The inequality (1.12) also implies that $|var_X(E')| = |var_X(E)|$, therefore $var_X(E') = var_X(E)$. The perfect matching in $(E', var_X(E'))$ is also a perfect matching in $(E', var_X(E))$. ∎

Lemma 1.7. Assume that there is a path P between a pair of equations e_1 and e_k in G where $k \neq 1$. The equations included in the path P are denoted E_P and the variables included in the path P are denoted X_P. Then for all equation sets $E \subset E_P$, it holds that

$$|E| \leq |var_{X_P}(E)|. \tag{1.13}$$

Furthermore

$$|E_P| = |X_P| + 1 \tag{1.14}$$

and

$$\forall X' \subseteq X_P, |X'| \neq 0 : |X'| < |equ_{E_P}(X')|. \tag{1.15}$$

Proof. Let the path P be defined as $e_1 - x_1 - e_2 - \cdots - x_{k-1} - e_k$. Take an arbitrary equation set $E \subset E_P$. We will find a complete matching of E into X_P. Since E is a proper subset of E_P, then there is an $e_j \in E_P$ such that $e_j \notin E$.

Assign x_i to e_i for all $e_i \in E$, where $i < j$, and assign x_{i-1} to e_i for all $e_i \in E$, where $i > j$. Corollary 1.4 implies that $\forall E' \subseteq E : |E'| \leq |var_{X_P}(E')|$ and especially $|E| \leq |var_{X_P}(E)|$. Hence (1.13) is proved.

The equation (1.14) follows from the fact that the starting point and the ending point of P are equation nodes.

Finally, choose any $X' \subseteq X_P$ where $|X'| \neq \varnothing$. Then it is clear that $\forall x_i \in X' : e_{i+1} \in equ_{E_P}(x_i)$. Furthermore, for the smallest i such that $x_i \in X'$ it follows that $e_i \in equ_{E_P}(x_i)$. All $|X'| + 1$ equations specified are different equations and hence (1.15) holds. ∎

Lemma 1.8. Consider the sets H, $E \subseteq H$, and $X \subseteq var_{X_u}(E)$. Assume that:

$$\forall X' \subseteq var_{X_u}(H), X' \neq \varnothing : |X'| < |equ_H(X')|, \tag{1.16}$$

$$\forall E' \subseteq E : |E'| \leq |var_{X_u}(E')|, \tag{1.17}$$

$$|E| = |X| + 1 \text{ ,and} \tag{1.18}$$

$$\forall X' \subseteq X, |X'| \neq 0 : |X'| < |equ_E(X')|. \tag{1.19}$$

Then there is a variable $x \in var_{X_u}(E) \backslash X$ and an equation $e \in H \backslash E$ such that $X \cup \{x\}$ and $E \cup \{e\}$ satisfy

$$\forall X' \subseteq X \cup \{x\}, |X'| \neq 0 : |X'| < |equ_{E \cup \{e\}}(X')| \tag{1.20}$$

and

$$|E \cup \{e\}| = |X \cup \{x\}| + 1. \tag{1.21}$$

Proof. First we will find an $x \notin X$ where $x \in var_{X_u}(E)$. Using (1.18) and $|E| \leq |var_{X_u}(E)|$ derived from (1.17), it is clear that $|X| < |E| \leq |var_{X_u}(E)|$. Hence there must be an $x \in var_{X_u}(E) \backslash X$. Take an arbitrary $x \in var_{X_u}(E) \backslash X$.

Let a set of variables X', where $X' \neq \varnothing$ and $X' \subseteq X \cup \{x\}$, be called a *critical set* if

$$|equ_E(X')| = |X'|. \tag{1.22}$$

Next, we will show that there is a unique minimal critical set. Suppose there are two minimal critical sets X_{c1} and X_{c2} where $X_{c1} \neq X_{c2}$. Note that $X_{ci} \neq \varnothing$ for $i \in \{1, 2\}$ to satisfy (1.22).

Suppose that $X' = X \cup \{x\}$ is a critical set such that $x \notin X'$. Then $X' \subseteq X$ and (1.19) can be used deriving $|X'| < |equ_E(X')|$. This contradicts the fact that X' is critical. Hence all critical sets include x.

Then it is possible to do the following partition of $X_{c1} \cup X_{c2}$, denoting $X_{c1} \cap X_{c2} = X_{12} \cup \{x\}$ where $x \notin X_{12}$, $X_1 = X_{c1} \backslash X_{c2}$, and $X_2 = X_{c2} \backslash X_{c1}$. Figure 1.3 visualizes the partition.

According to the partition, the critical sets X_{c1} and X_{c2} are expressed as

$$X_{c1} = X_1 \cup X_{12} \cup \{x\} \tag{1.23}$$

and

$$X_{c2} = X_2 \cup X_{12} \cup \{x\}. \tag{1.24}$$

From (1.22), (1.23), (1.24), and the fact that X_{ci} is critical it follows that

$$|equ_E(X_{ci})| = |X_{ci}| = |X_i| + |X_{12}| + 1 \quad \text{for} \quad i \in \{1, 2\}. \tag{1.25}$$

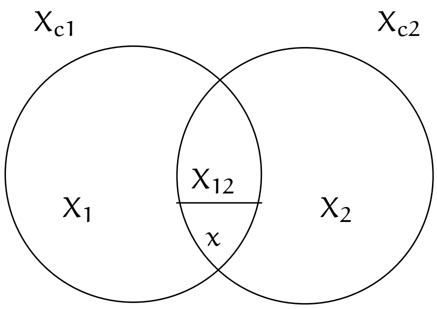

Figure 1.3: The partition of X_{c1} and X_{c2}.

Since X_{c2} is a minimal critical set it follows that $X_{c1} \not\subseteq X_{c2}$. Using (1.23) and (1.24) and the knowledge that these are partitions, imply the equivalent expression $X_1 \neq \varnothing$. This implies that

$$X_{12} \cup \{x\} \subset X_1 \cup X_{12} \cup \{x\} = X_{c1}. \tag{1.26}$$

Consider first any subset $X' \neq \varnothing$ of $X \cup \{x\}$ such that $X' \neq \{x\}$. From (1.19) it follows that

$$\forall X' \subseteq X \cup \{x\}, |X'| \leq |X' \backslash \{x\}| + 1 \leq |equ_E(X' \backslash \{x\})| \leq |equ_E(X')|. \tag{1.27}$$

Further on, if $X' = \{x\}$ then

$$|\{x\}| \leq |equ_E(\{x\})|, \tag{1.28}$$

because of the fact that x is chosen such that $x \in var_{X_u}(E)$. The inequalities (1.27) and (1.28) implies that

$$\forall X' \subseteq X \cup \{x\}, X' \neq \varnothing : |X'| \leq |equ_E(X')|. \tag{1.29}$$

Now, the minimality of X_{c1} and (1.26) imply that $X \cup \{x\}$ is not critical, i.e.

$$|equ_E(X \cup \{x\})| \neq |X \cup \{x\}|. \tag{1.30}$$

The set $X \cup \{x\}$ satisfies (1.29) and (1.30), hence

$$|equ_E(X_{12} \cup \{x\})| \geq |X_{12} \cup \{x\}| + 1 = |X_{12}| + 2. \tag{1.31}$$

From the definition of the function equ it follows that for arbitrary variable sets A and B and for an arbitrary equation set $\bar{E}$ it holds that

$$equ_{\bar{E}}(A \cup B) = equ_{\bar{E}}(A) \cup equ_{\bar{E}}(B). \tag{1.32}$$

Using (1.32) and basic set theory implies

$$\begin{aligned}|equ_E(X_1 \cup X_{12} \cup \{x\}) \cup equ_E(X_2 \cup X_{12} \cup \{x\})| = \\ |equ_E(X_1 \cup X_{12} \cup \{x\} \cup X_2 \cup X_{12} \cup \{x\})| = \\ |equ_E(X_1 \cup X_2 \cup X_{12} \cup \{x\})|.\end{aligned} \tag{1.33}$$

Further on, it holds that

$$\begin{aligned}equ_E(X_1 \cup X_{12} \cup \{x\}) \cap equ_E(X_2 \cup X_{12} \cup \{x\}) = \\ \big(equ_E(X_1) \cup equ_E(X_{12} \cup \{x\})\big) \cap \big(equ_E(X_2) \cup equ_E(X_{12} \cup \{x\})\big) = \\ \big(equ_E(X_1) \cap equ_E(X_2)\big) \cup equ_E(X_{12} \cup \{x\}).\end{aligned} \tag{1.34}$$

The last row in (1.34) can be underestimated using (1.31)

$$\begin{aligned}\big|\big(equ_E(X_1) \cap equ_E(X_2)\big) \cup equ_E(X_{12} \cup \{x\})\big| \geq \\ |equ_E(X_{12} \cup \{x\})| \geq |X_{12}| + 2.\end{aligned} \tag{1.35}$$

Now, we will apply $|A \cup B| = |A| + |B| - |A \cap B|$, where $A = equ_E(X_1 \cup X_{12} \cup \{x\})$ and $B = equ_E(X_2 \cup X_{12} \cup \{x\})$. The left hand side can be simplified using (1.33). The result is

$$\begin{aligned}|equ_E(X_1 \cup X_2 \cup X_{12} \cup \{x\})| = |equ_E(X_1 \cup X_{12} \cup \{x\})| + \\ |equ_E(X_2 \cup X_{12} \cup \{x\})| - \\ |equ_E(X_1 \cup X_{12} \cup \{x\}) \cap equ_E(X_2 \cup X_{12} \cup \{x\})|.\end{aligned} \tag{1.36}$$

Further, substitute the results in (1.23), (1.24), (1.25), and (1.34) into (1.36), then

$$\begin{aligned}|equ_E(X_1 \cup X_2 \cup X_{12} \cup \{x\})| = |X_1| + |X_{12}| + 1 + |X_2| + \\ |X_{12}| + 1 - |\big(equ_E(X_1) \cap equ_E(X_2)\big) \cup equ_E(X_{12} \cup \{x\})|.\end{aligned} \tag{1.37}$$

The last part of (1.37) is overestimated using (1.35)

$$\begin{aligned}|X_1| + |X_2| + 2|X_{12}| + 2 - \\ |\big(equ_E(X_1) \cap equ_E(X_2)\big) \cup equ_E(X_{12} \cup \{x\})| \\ \leq |X_1| + |X_2| + 2|X_{12}| + 2 - (|X_{12}| + 2) = |X_1| + |X_2| + |X_{12}|.\end{aligned} \tag{1.38}$$

The result of putting (1.37) and (1.38) together is

$$\begin{aligned}|X_1| + |X_2| + |X_{12}| \geq |equ_E(X_1 \cup X_2 \cup X_{12} \cup \{x\})| \\ \geq |equ_E(X_1 \cup X_2 \cup X_{12})|.\end{aligned} \tag{1.39}$$

Finally, $X_1 \cup X_2 \cup X_{12} \subseteq X$ and according to (1.19) is

$$|X_1| + |X_2| + |X_{12}| < |equ_E(X_1 \cup X_2 \cup X_{12})|. \tag{1.40}$$

The inequalities (1.39) and (1.40) implies a contradiction. Hence there cannot be two minimal critical sets. Let the unique minimal critical set be denoted $X_{critical}$.

Now, it is time to show that there exists an equation $e \in H \backslash E$ that fulfill (1.20). Suppose that $equ_H(X_{critical}) \subseteq E$. This together with (1.22) implies that $|X_{critical}| = |equ_E(X_{critical})| = |equ_H(X_{critical})|$.This is a contradiction according to (1.16). Hence there is an equation $e \in H \backslash E$ such that $var_{X_{critical}}(e) \neq \varnothing$. $X_{critical} \cap var_{X \cup \{x\}}(e) \neq \varnothing$.

Now, we will show that this e fulfills (1.20). Take an arbitrary $X' \subseteq X \cup \{x\}$ where $X' \neq \varnothing$. Consider first the case when $X_{critical} \subseteq X'$. Then it follows from (1.29) that $|X'| \leq |equ_E(X')| < |equ_{E \cup \{e\}}(X')|$. The last inequality follows from the fact that $equ_{\{e\}}(X_{critical}) = \{e\}$.

The opposite case is when $X_{critical} \not\subseteq X'$. From the fact that there is a unique minimal critical set it follows that X' cannot be a critical set. Hence (1.29) and $|X'| \neq ||equ_E(X')$ conclude that $|X'| < |equ_E(X')| \leq |equ_{E \cup \{e\}}(X')|$, i.e. (1.20) holds.

Finally, it remains to prove (1.21). Simple calculations using (1.18) gives $|E \cup \{e\}| = |E| + 1 = |X| + 2 = |X \cup \{x\}| + 1$. ■

Lemma 1.9. Let the equation set H have the property (1.16). Suppose that $E_j \subseteq H$ has the property (1.17), and the corresponding X_j has the properties (1.18) and (1.19). Let $E_{j+1} = E_j \cup \{e\}$ and $X_{j+1} = X_j \cup \{x\}$ where x and e are defined in Lemma 1.8. Then E_{j+1} is either an MSS set or $\forall E' \subseteq E_{j+1} : |E'| \leq |var_{X_u}(E')|$.

Proof. Note that according to Lemma 1.8, (1.21) and (1.20) hold for the set E_{j+1} and X_{j+1}. Take any $\hat{E} \subset E_{j+1}$. Then there is an $e \in E_{j+1} \backslash \hat{E}$ such that $\hat{E} \subseteq E_{j+1} \backslash \{e\}$. From (1.20) it follows that

$$\forall X' \subseteq X_{j+1}, X' \neq \varnothing : |X'| \leq |equ_{E_{j+1}}(X')| - 1 \leq |equ_{E_{j+1} \backslash \{e\}}(X')| \tag{1.41}$$

Especially, if $X' = X_{j+1}$ in (1.41) then

$$|X_{j+1}| \leq |equ_{E_{j+1} \backslash \{e\}}(X_{j+1})| \tag{1.42}$$

holds. From (1.21) it follows that

$$|equ_{E_{j+1} \backslash \{e\}}(X_{j+1})| \leq |E_{j+1} \backslash \{e\}| = |X_{j+1}|. \tag{1.43}$$

The inequalities (1.42) and (1.43) imply

$$|X_{j+1}| = |equ_{E_{j+1}\backslash\{e\}}(X_{j+1})|. \tag{1.44}$$

Now, using (1.41) in Corollary 1.5 it follows that there is a complete matching of X_{j+1} into $E_{j+1}\backslash\{e\}$. The complete matching is also a perfect matching, according to (1.44). A perfect matching is especially a complete matching of $E_{j+1}\backslash\{e\}$ into X_{j+1}. Corollary 1.4 implies that

$$\forall E' \subseteq E_{j+1}\backslash\{e\} : |E'| \leq |var_{X_{j+1}}(E')|. \tag{1.45}$$

Since $\hat{E} \subseteq E_{j+1}\backslash\{e\}$, then $E' = \hat{E}$ in (1.45) implies that

$$|\hat{E}| \leq |var_{X_{j+1}}(\hat{E})|. \tag{1.46}$$

The set $\hat{E}$ was an arbitrary proper subset to E_{j+1}. This implies that

$$\forall E' \subset E_{j+1} : |E'| \leq |var_{X_{j+1}}(E')| \leq |var_{X_u}(E')|. \tag{1.47}$$

Now, it remains to study $|var_{X_u}(E_{j+1})|$. From (1.47) it holds that

$$\begin{aligned} |E_{j+1}| = |E_j| + 1 \leq |var_{X_{j+1}}(E_j)| + 1 \leq |var_{X_{j+1}}(E_{j+1})| + 1 \leq \\ |var_{X_u}(E_{j+1})| + 1. \end{aligned} \tag{1.48}$$

There are two cases. Suppose that equality in (1.48) holds, i.e.

$$|E_{j+1}| = |var_{X_u}(E_{j+1})| + 1. \tag{1.49}$$

From (1.47), (1.49), and the definition of MSS sets it follows that E_{j+1} is an MSS set.

Next assume that, (1.48) is a strict inequality, i.e.

$$|E_{j+1}| \leq |var_{X_u}(E_{j+1})|. \tag{1.50}$$

Hence according to (1.47) and (1.50) it follows that

$$\forall E' \subseteq E_{j+1} : |E'| \leq |var_{X_u}(E')|. \tag{1.51}$$

■

Lemma 1.10. Let $H \subseteq M_{orig}$, where M_{orig} fulfills Assumption 1.1. Further let H and all E_i fulfill Assumption 1.2. For each $e \in H$, there exists an $E \subseteq H$ such that $e \in E$, E is an MSS set, and $var_Y(E) \neq \emptyset$.

Proof. First we will find an MSS set. This is done by adding equations to e until an MSS set is found. Then, we prove that this MSS set has to include known variables.

Suppose that $var_{X_u}(e) = \varnothing$. Then e is an MSS set.

If on the contrary $var_{X_u}(e) \neq \varnothing$, then there is an $x \in var_X(e)$. Let $X_1 = \{x\}$. According to (1.4) it follows that there is an $e' \in equ_H(X_1)\backslash\{e\}$. Define a path P as $e - x - e'$. Let the equations in the path P, be denoted E_1.

Recalling the definition of MSS, Lemma 1.7 imply according to (1.13) that

$$\forall E' \subset E_1 : |E'| \leq |var_{X_1}(E')|. \tag{1.52}$$

There are two cases, either $|E_1| = |var_{X_u}(E_1)| + 1$ or $|E_1| \leq |var_{X_u}(E_1)|$. In the first case it means that E_1 is an MSS set. The second case implies that $\forall E' \subseteq E_1 : |E'| \leq |var_{X_1}(E')|$ holds.

If the first case is present the goal is achieved. Therefore suppose that the second case is present. If it holds that $\forall E' \subseteq E_j : |E'| \leq |var_{X_u}(E')|$ then Lemma 1.9 implies that E_{j+1} is either MSS or has the property $\forall E' \subseteq E_{j+1} : |E'| \leq |var_{X_u}(E')|$. Since H is assumed to have property (1.4), $|var_{X_u}(H)| < |H| < \infty$ there is a finite number of unknown variables. Then $E_{|var_{X_u}(H)|}$ has to be an MSS set if for all j, $1 \leq j < |X_H|$, E_j is not an MSS set. Hence, it is always possible to find an MSS set including e. ■

Lemma 1.11. Let X be the unknown variables $X = var_{X_u}(E)$. If E is an MSS then $\forall \bar{X} \subseteq X, \bar{X} \neq \varnothing : |equ_E(\bar{X})| > |\bar{X}|$.

Proof. Consider the negation of the conclusion. That is, E is an MSS set and

$$\exists \bar{X} \subseteq X, \bar{X} \neq \varnothing : |equ_E(\bar{X})| \leq |\bar{X}|. \tag{1.53}$$

Let X' be an $\bar{X}$ that fulfill the requirement. From Theorem 1.6 and from the fact that E is an MSS set, it follows that $\forall e \in E : (E\backslash\{e\}, X)$, contains a perfect matching. From the definition of perfect matching it particularly follows that there is a complete matching from X into $E\backslash\{e\}$. The use of Corollary 1.5 makes it possible to write

$$\forall e \in E \forall \bar{X} \subseteq X : |\bar{X}| \leq |equ_{E\backslash\{e\}}(\bar{X})|. \tag{1.54}$$

Since $X = var_X(E)$ it means that $\forall x \in X \exists e \in E : x \in e$. Especially it holds that $\forall x \in X' \exists e \in E : x \in e$ since $\varnothing \neq X' \subseteq X$. Hence

$\mathsf{equ}_E(X') \neq \varnothing$. Now apply (1.54) to X' and an $e' \in \mathsf{equ}_E(X')$, that is

$$|X'| \leq |\mathsf{equ}_{E\setminus\{e'\}}(X')|. \tag{1.55}$$

From $e' \in \mathsf{equ}_E(X')$ follows that $e' \in \mathsf{equ}_{e'}(X')$, hence $|\mathsf{equ}_{e'}(X')| = 1$. Adding $|\mathsf{equ}_{e'}(X')| = 1$ on the right-hand side of (1.55) it becomes,

$$\begin{aligned} |X'| < |\mathsf{equ}_{E\setminus\{e'\}}(X')| + \\ +|\mathsf{equ}_{\{e'\}}(X')| = |\mathsf{equ}_E(X')|. \end{aligned} \tag{1.56}$$

This is a contradiction. Hence the theorem follows. ■

Now, it is time to prove Theorem 1.2.

Proof. ⇒) There is a consistency relation $c(\mathbf{y}) = 0$ of $H(\mathbf{x}, \mathbf{y})$. This is according to Assumption 1.2 equivalent to

$$\forall X' \subseteq var_{X_u}(H), X' \neq \varnothing : |X'| < |\mathsf{equ}_H(X')|$$

Assumption 1.1, is valid especially for H. According to Lemma 1.10 there is for each $e_i \in H$ an MSS set $E_i \subseteq H$, with $e_i \in E_i$, and $var_Y(E_i) \neq \varnothing$. Hence $H = \cup_i E_i$.

⇐) Take an arbitrary $H = \bigcup_i E_i$ where all E_i are MSS sets.

and $var_Y(E_i) \neq \varnothing$. Hence property (1.4) holds for all E_i. Lemma 1.11 applied to each E_i gives that E_i has property (1.4). From the main assumption it follows that there exist a consistency relation $c_i(Y)$ for each $E_i(Y)$. Let $c(Y) = \sum_i c_i^2(Y)$, then $c(Y) = 0$ iff $\forall i : c_i(Y) = 0$. This is equivalent to $c(Y) = 0$ iff $H(Y)$, hence $c(Y)$ is a consistency relation for $H(Y)$. ■

Chapter 2

Algorithm for Finding MSS Sets

The objective is to find all potential consistency relations for a given model $\mathsf{M}_{\mathsf{orig}}$ 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 $\mathsf{M}_{\mathsf{orig}}$. The algorithm can be summarized by the following steps.

Algorithm 1.

1. Differentiate the model: Find equations that are meaningful to differentiate for finding MSS sets.

2. Simplify 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 reduce the computational complexity of the next step, merge sets of equations that have to be used together in each MSS set.

3. Find MSS sets: Search for MSS sets in the simplified model.

4. Analyze diagnosability: Examine the diagnosability of the MSS sets found in step 3.

5. Decouple 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. Select 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 the algorithm.

2.1 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 2.1 Consider the model $E = \{e_1, e_2, e_3\} = \{y_1 = x, y_2 = \dot{x}, y_3 = x^2\}$. An algorithm that is not capable of differentiating equations can obviously not eliminate $\dot{x}$ in e_2, because there is no other equation including $\dot{x}$. 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 $\cup_{i=0}^{\infty} E^{(i)}$.

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 2.2 Consider again the three equations in Example 2.1. The differentiated equation $\dot{e}_3$ is $\dot{y}_3 = 2\,x\,\dot{x}$. The variable y_3 is linearly dependent in e_3 and therefore $\dot{y}_3$ is linearly contained in equation $\dot{e}_3$. Furthermore, both x and $\dot{x}$ are nonlinearly contained in $\dot{e}_3$ as a consequence of the fact that x is nonlinearly contained in e_3.

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. If a limit $m(y)$ for variable $y \in Y$ of the order of derivative that can be considered as possible to estimate is introduced, it is possible to find all MSS sets also in a finite subset of the differentiated model. A sufficient condition that there is a finite submodel that contains all MSS sets is that the original model M_{orig} satisfy Assumption 1.1 and all known variables have finite limitations.

Algorithm 2 is a greatly influenced of Pantelides' algorithm [10]. 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. $\beta(M, x) = \max(\{i | x^{(i)} \in var_{X_u}(M)\})$. Finally let $\widehat{var}(M)$ be the variables $var_{X_u \cup Y}(M)$ 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 $\dot{y} \in var_{X_u \cup Y}(M)$, $\forall i \in \mathbb{Z}_+ \backslash \{1\} : y^{(i)} \notin var_{X_u \cup Y}(M)$, and $m(y) = 1$, then $\dot{y} \in \widehat{var}(M)$ because $\dot{y}$ is the highest derivative of y in M and $\ddot{y}$ is unknown.

Algorithm 2.
Input: The original model M_{orig}, a description of which variables that are linearly contained in each equation, and for each $y \in \overline{var}_Y(M_{orig})$, $m(y) < \infty$.

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}. A bipartite graph can now be defined with nodes M_c^{max} and $\widehat{var}(M_c^{max})$. There is an edge between a variable node and an equation node if and only if the corresponding variable is included in the corresponding equation. Let equation i in M_c^{max} be denoted e_i. A complete matching of $M_c^{max}(i-1)$ into $\widehat{var}(M_c^{max})$ is found in previous steps. A path that have alternating matched and unmatched edges and that starts and finishes in unassigned nodes is called an augmented path. Search for an augmented path from e_i to an unassigned variable in $\widehat{var}(M_c^{max})$.

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

b) No augmented path is found. Then an MSS set with respect to $\widehat{var}(M_c^{max})$ is found as follows. Let all edges not included in the matching be directed edges from the equation nodes to the variable nodes. Then the MSS set with respect to $\widehat{var}(M_c^{max})$ 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 $\widehat{var}(M_c^{max})$ instead of MSS with respect to X_u. Differentiate E until $|\widehat{var}(E^{(i)})| \geq |E|$ 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}.

Next an example is used to describe how Algorithm 2 works.

Example 2.3 The following example is a continuation of Example 1.1 with the structural model shown in (1.1). Let $m(u) = m(y_f) = 1$ and $m(y_h) = 0$ and assume that no variable is linearly contained in any equation. Then no variable will disappear in the differentiation. Furthermore 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. The corresponding bipartite graph for (1.1) is shown in Figure 2.1.

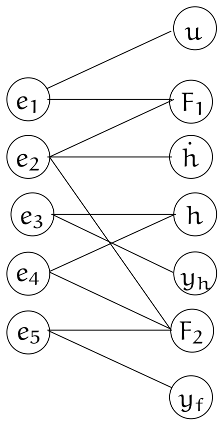

Figure 2.1: The bipartite graph corresponds to the structural model (1.1) when all faults are assumed to be zero.

Step 2 in Algorithm 2 is fed the structural model M_c shown in Figure 2.1 and the m-values. Figure 2.2 shows the graph 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 $\widehat{var}(M_c^{max})$. 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 $\widehat{var}(M_c^{max})$ in the graph showed in Figure 2.2. 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_5 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 $\widehat{var}(M_c^{max})$. When edges not contained in the matching are directed from equation nodes to variable nodes, the reachable equation nodes from e_5 are e_4 and e_5. Hence this is the equation set to be differentiated. Differentiating once implies that $\dot{y}_f$ appears in $\widehat{var}(\{\dot{e}_4, \dot{e}_5\})$. The new model consists of $\{e_1, e_2, e_3, e_4, \dot{e}_4, e_5, \dot{e}_5\}$ and the new bipartite graph showed in Figure 2.3 is extracted in step 2. Equation e_4 and e_5 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 2.2 is either unchanged or replaced

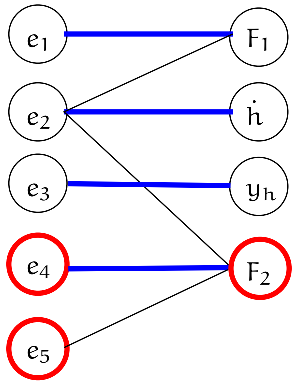

Figure 2.2: The bipartite graph M_c^{max} built in step 2. The bold edges is the matching found. The bold equation nodes are the MSS set found in step 2b).

with an edge between the replaced nodes corresponding to the differentiated equation and the differentiated variable in Figure 2.3. For example $\{e_1, F_1\}$ is unchanged and the edge $\{e_4, F_2\}$ in Figure 2.2 is replaced with $\{\dot{e}_4, \dot{F}_2\}$ in Figure 2.3. Step 2 finds an assignment for $\dot{e}_5$. The structural model M_{diff} obtained from Algorithm 2 is shown in Figure 2.4.

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 2.1. If Assumption 1.1 is satisfied and for each $y \in \overline{var}_Y(M_{orig})$, $m(y) < \infty$, then

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

Proof. The proof consists of two parts. The first part states that Algorithm 2 terminates and that the differentiated model has the property that there is a complete matching from M_{diff}^{max} into $\widehat{var}(M_{diff}^{max})$. The second part uses this complete matching and shows that $MSS_{all}(M_{orig}) = MSS(M_{diff})$.

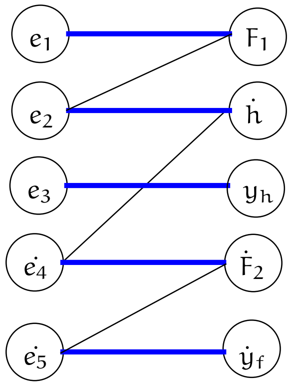

Figure 2.3: The bipartite graph M_c^{max} built in step 2 after one differentiation. The bold edges is the complete matching found in step 2.

equation	unknown					fault				known			
	F_1	F_2	$\dot{F}_2$	h	$\dot{h}$	f_u	f_{yh}	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		X		X									
$\dot{e}_4$		X	X	X	X								
e_5		X						X				X	
$\dot{e}_5$		X	X					X	X			X	X

Figure 2.4: The structural model M_{diff} obtained from Algorithm 2 when applied to the structural model (1.1).

Algorithm 2 terminates when $i = |M_{orig}|$. The variable i is increased in step 2a). Step 2a) is done when an augmented path from e_i to an unassigned variable in $\widehat{var}(M_c^{max})$ is found.

To show that Algorithm 2 terminates is equivalent to show that an augmented path from e_i to an unassigned variable in $\widehat{var}(M_c^{max})$ is always found in finitely many iterations.

First it will be shown that the differentiation in step 2b) is always terminated. Therefore assume that there is no augmented path from e_i to an unassigned variable in $\widehat{var}(M_c^{max})$ in step 2. Then step 2b) finds

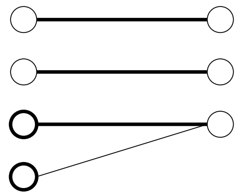

Figure 2.5: The left nodes are the equation nodes M_c^{max} and the right nodes are the variable nodes $\widehat{var}(M_c^{max})$. The bold edges represents a matching. The bold equation nodes are an MSS set E with respect to $\widehat{var}(M_c^{max})$.

an MSS set with respect to $\widehat{var}(M_c^{max})$. Let this MSS set be denoted E. In Figure 2.5 the left nodes are the equation nodes M_c^{max} and the right nodes are the variable nodes $\widehat{var}(M_c^{max})$. The bold edges represents a matching. The bold equation nodes are an MSS set E with respect to $\widehat{var}(M_c^{max})$. The model M_c and the MSS set E can be realized to have the following property.

$$\begin{aligned}&\big(\forall x \in \overline{var}_{X_u \cup Y}(E) : \beta(M_c, x) < \infty \wedge \\ &\forall y \in \overline{var}_Y(E) : m(y) < \infty\big) \quad \Rightarrow \\ &\exists m \in \mathbb{N} : \big(\forall x \in \overline{var}_{X_u}(E) : \beta(E^{(m)}, x) \geq \beta(M_c, x) \wedge \\ &\forall y \in \overline{var}_Y(E) : \beta(E^{(m)}, y) \geq \max(\beta(M_c, y), m(y))\big)\end{aligned} \tag{2.1}$$

Since both the highest derivatives of each variable and all limits on known variables are finite, it is possible to exceed those limits by differentiating E, m number of times. Assumption 1.1 guarantees that $|E| \leq |\overline{var}_{X_u \cup Y}(E)|$. According to expression (2.1) each variable in $\overline{var}_{X_u \cup Y}(E)$ will have a corresponding derivative in $\widehat{var}(E^{(m)})$. Hence $|\widehat{var}(E^{(m)})| = |\overline{var}_{X_u \cup Y}(E)| \geq |E|$ which is the stop condition of step 2b). After the redefinition of M_c^{max} in step 2 at least one new variable is included in $var_{\widehat{var}(M_c^{max})}(E^{(m)})$.

According to Lemma 2.2 the differentiation in step 2b) will not remove any corresponding edge in previous found matching.

Next to show is that the loop using step 2b) terminates, i.e. after a finite number iterations using step 2b) Algorithm 2 finds an augmented path and step 2a) is applied. Since the previous matching has a corresponding matching, the corresponding matching together with the augmented path defines a new extended matching.

As explained above the differentiation is terminated and there is at least one new variable included in $var_{\widehat{var}(M_c^{max})}(E^{(m)})$. The result of finding new variable nodes is divided into two cases.

1. All new variable nodes are already included in the matching. An example is shown in Figure 2.6 where the dashed edge is the newly appeared. Then there is a new MSS set $\hat{E}$ with respect to $\widehat{var}(M_c^{max})$ in the right graph in Figure 2.6 denoted with bold nodes. The definition of structural differentiation implies that the graphs $(E, \widehat{var}(E))$ and $(E^{(m)}, \widehat{var}(E^{(m)}))$ are isomorphic. If E is differentiated m number of times then it is clear according to how the MSS set is obtained in Algorithm 2 and the fact that the subgraphs $(E, \widehat{var}(E))$ and $(E^{(m)}, \widehat{var}(E^{(m)}))$ are isomorphic that $E^{(m)} \subset \hat{E}$. Since the new MSS set $\hat{E}$ is including E (m) then this case can only be repeated i times. Therefore it is sufficient to prove that given case 2 an augmented path will be found and hence step 2b) will be followed by step 2a).

2. There is a new variable that is not included in the matching. All nodes are reachable from $e_i^{(m)}$ when all edges not included in the matching are directed edges from the equation nodes to the variable nodes, i.e. there is an augmented path from $e_i^{(m)}$ to the new variable node in $\widehat{var}(E^{(m)})$. This augmented path defines a new complete matching including $e_i^{(m)}$. In Figure 2.7 there is a new edge to a new unassigned variable. There is an augmenting path from the last equation node to the last variable node. In the right figure the new matching is defined.

Hence the algorithm will terminate and find a complete matching of M_{diff}^{max} into $\widehat{var}(M_{diff}^{max})$.

Now it remains to prove that M_{diff} contains all MSS sets. From Lemma 2.3 it follows that $MSS_{all}(M_{orig}) \subseteq MSS(M_{diff})$. Since $M_{diff} \subset M_\infty$, it implies that $MSS(M_{diff}) \subseteq MSS(M_\infty)$
$= MSS_{all}(M_{orig})$. Hence $MSS(M_{diff}) = MSS_{all}(M_{orig})$. ■

Now the two Lemmas 2.2 and 2.3 will be discussed and proven. According to Algorithm 2 assignments for each equation in M_c are sequentially found. However it is important that the differentiation of the MSS set found in step 2b) E not removes any edge that is included in the matching. To show that differentiation not removes

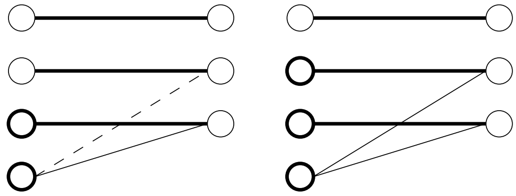

Figure 2.6: In the left graph the to last equation nodes have been differentiated. As a result of the differentiation the dashed edge appears. The new variable node is already included in the matching as shown in the left graph. Then there is a new MSS set with respect to $\widehat{var}(M_c^{max})$ in the right graph denoted with bold nodes. Note that MSS set in the left graph is a subset to the MSS set in the right graph.

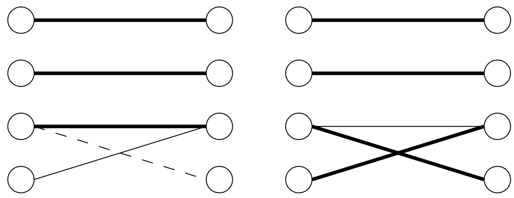

Figure 2.7: In the left graph the to last equation nodes have been differentiated. As a result of the differentiation the dashed edge appears. There is an augmenting path from the last equation node to the last variable node. In the right figure the new matching is defined switching assigned and unassigned edges.

any edge included in the matching let a bipartite graph be defined as $G_1 = (M_c^{max}, \widehat{var}(M_c^{max}))$. Consider two subsequent graphs G_1 and G_2, i.e. G_2 is the resultant bipartite graph in step 2 after step 2b) is applied to G_1. Suppose that step 2b) differentiate E, m number of times.

Lemma 2.2. Step 2b) preserves any matching, i.e. if there is an edge $\{e, x\}$ included in the matching in G_1, then there will be a corresponding edge in G_2 between $\{e^{(j)}, x^{(j)}\}$ where $j = 0 \vee m$.

Proof. Let $\hat{e}$ denote the last introduced equation before a complete matching is not found, i.e. $\hat{e}$ together with the previous matching defines the MSS set E. There are two cases:

1. The equation e is included in the last complete matching and is not differentiated. Then according to Algorithm 2 there can not be a directed path from $\hat{e}$ to e considering unmatched edges as directed edges from equation nodes to variable nodes. This implies that there is no directed path from $\hat{e}$ to x either. If any of the equations reachable from $\hat{e}$ included x then x would also be reachable from $\hat{e}$. This is not true and the conclusion is that no equations including x are differentiated. Hence the variable x is not involved in the differentiation. The edge $\{e, x\}$ and the nodes e and x are unchanged from G_1 to G_2.

2. The equation e is included in the last complete matching and is differentiated. Then according to Algorithm 2 there is a directed path from $\hat{e}$ to e considering unmatched edges as directed edges from equation nodes to variable nodes. Since the only incoming edge to e is the edge $\{e, x\}$ the only possible directed path to e goes through the variable node corresponding to x. Hence e and x are replaced with $e^{(1)}$ and $x^{(1)}$ in G_2 respectively. The edge $\{e^{(1)}, x^{(1)}\}$ is obviously included in G_2.

In both two cases the matching is preserved and therefore the lemma is proven. ■

Lemma 2.3. If there is a complete matching of the most differentiated equations in M_{diff} into the variable nodes in to $\widehat{var}(M_c^{max})$. Then all $MSS_{all}(M_{orig}) \subseteq MSS(M_{diff})$.

Proof. Let the equations and variables in the complete matching be denoted e_i and x_i respectively such that (e_i, x_i) is an assignment. It is clear that $\forall j \in \mathbb{Z}^+ : e_i^{(j)} \notin M_{diff}$ and $\forall j \in \mathbb{Z}^+ : x_i^{(j)} \notin var_{X_u \cup}(M_{diff})$.

Take an arbitrary set of equations E such that $E \subset M_\infty$ and $E \cap (M_\infty \backslash M_{diff}) \neq \varnothing$. Call this intersection E'. Let the equations in E' be

$$\begin{array}{l} e_1^{(\alpha_1)}, \cdots, e_1^{(\alpha_{n_1})} \\ \vdots \\ e_m^{(\alpha_1)}, \cdots, e_m^{(\alpha_{n_m})} \end{array} \tag{2.2}$$

Note that all $\alpha_i > 0$. According to the complete matching, it is clear that $x_i^{(\alpha)} \in var_{X_u}(e_i^{(\alpha)})$ for $\alpha > 0$. Further $x_i^{(\alpha)} \notin M_{diff}$, where $\alpha > 0$.

Now, the idea is to apply Lemma 1.11 on the variables set $X = \{x_i^{(\alpha_j)} | 1 \leq i \leq m, 1 \leq j \leq n_i\}$. The number of variables is $|X| = \sum_{i=1}^{m} n_i$. From the fact that $var_X(M_{diff}) = \emptyset$ and $x_i^{(\alpha)} \in var_X(e_i^{(\alpha)})$ it follows that $equ_E(X) = E'$. The number of equations in E' is $|E'| = \sum_{i=1}^{m} n_i = |X|$. Lemma 1.11 conclude that E can not be an MSS set. Hence, given any MSS set E, it follows that $E \subseteq M_{diff}$. ■

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}.

2.2 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 [3]. The remaining structural model is denoted M_{simp1}.

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 $1 + |X| = |equ_{M_{simp1}}(X)|$, then all equations in $equ_{M_{simp1}}(X)$ 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 $equ_{M_{simp1}}(X)$ has to include all equations in $equ_{M_{simp1}}(X)$. The idea is to find these sets. Then it is in these sets possible to eliminate internal variables, in the previous discussion denoted X. Each such 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 in common. When 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}.

Algorithm 3.
Input: M_{simp1}

1. Set $X = var_{X_u}(M_{simp1})$ and $M_{simp} = M_{simp1}$.
2. For all variables $x \in X$ do step 3.
3. If $|equ_{M_{diff}}(x)| = 2$ then set $X = X \backslash \{x\}$ and let the two equations $equ_{M_{diff}}(x)$ in M_{simp} be replaced with one new e_{new} where $var_{X_u \cup Y \cup F}(e_{new}) = var_{X \cup Y \cup F}(equ_{M_{simp}}(x))$. For all $e \in M_{simp} \backslash equ_{M_{simp}}(x)$ let $var_{X_u \cup Y \cup F}(e) = var_{X \cup Y \cup F}(e)$.
4. If some simplifications were made in step 3 go back to step 2.

Output: M_{simp}.

The complexity of Algorithm 3 is $O(|var_{X_u}(M_{simp})|^2)$. The next theorem ensures that no MSS set is lost in the simplification step.

Theorem 2.4. $MSS(M_{diff}) = MSS(M_{simp})$

The simplification step is divided into two parts. The proof of Theorem 2.4 is therefore divided into two Lemmas corresponding to the first and the second part of the simplification.

Proof. Theorem 2.4 follows directly from Lemma 2.5 and Lemma 2.6. ∎

The first simplification step relies on the following lemma.

Lemma 2.5. $MSS(M_{diff}) = MSS(M_{simp1})$

The proof can be found in [3]. The second part of the simplification step is correct according to Lemma 2.6.

Lemma 2.6. $MSS(M_{simp1}) = MSS(M_{simp})$

Proof. Since Algorithm 3 only changes the model M_{simp} in step 3 it is sufficient to prove that this operation on M_{simp} preserves the MSS sets included. Let the structural models before and after a simplification in step 3 be denoted M_1 and M_2 respectively.

The model M_{simp1} has according the first simplification step the property (1) in Lemma 2.7. Algorithm 3 finds for example that $x \in var_{X_u}(M_1)$ fulfill $|equ_{M_1}(x)| = 2$. Let $X = \{x\}$ in Lemma 2.7 then (2), and (3) in Lemma 2.7 are fulfilled.

Take an arbitrary MSS set $E \subseteq M_1$ i.e. property (4) in Lemma 2.7. There are two cases two consider:

1. It holds that $E \cap equ_{M_1}(x) = \varnothing$. Then E is not involved in the simplification and it is clear that $E \subseteq M_2$.

2. Otherwise it holds that $E \cap equ_{M_1}(x) \neq \varnothing$. This is condition (5) in Lemma 2.7. Since all 5 conditions in Lemma 2.7 are fulfilled the conclusion $equ_{M_1}(x) \subseteq E$ follows. This means that $equ_{M_1}(x)$ could be considered as one equation derived from $equ_{M_1}(x)$ by eliminating the variable x. Hence $E \subseteq M_2$. Moreover if M_1 has property (1) in Lemma 2.7, then M_2 has also this property because

$$\forall \bar{X} \neq \varnothing, \bar{X} \subseteq var_{X_u}(M_2) : |\bar{X}| = |\bar{X} \cup \{x\}| - 1 < \\ < |equ_{M_1}(\bar{X} \cup \{x\})| - 1 = |equ_{M_2}(\bar{X})|$$

Since E was an arbitrary MSS set then $MSS(M_1) = MSS(M_2)$. Algorithm 3 applies step 3 repeatedly, hence $MSS(M_{simp1}) = MSS(M_{simp})$. ■

Lemma 2.7. Given that

1. the system M has the property $\forall \bar{X} \neq \varnothing$, $\bar{X} \subseteq var_{X_u}(M) : |\bar{X}| < |equ_M(\bar{X})|$,

2. $X \neq \varnothing$,

3. $1 + |X| = |equ_M(X)|$,

4. E is an MSS set and $E \subseteq M$,

5. $E \cap equ_M(X) \neq \varnothing$,

then $equ_M(X) \subseteq E$.

Proof. Let $E' = E \cap equ_M(X)$ and $X_{E'} = var_X(E')$. Since $E \cap equ_M(X) \neq \emptyset$, then $\exists e \in E' \exists x \in X : x \in var_X(\{e\})$. It follows that $\emptyset \neq \{x\} \subseteq var_X(E') = X_{E'}$.

Suppose that $X \backslash X_{E'} = \emptyset$. Then $X_{E'} = X$ since $X_{E'} \subseteq X$. Apply Lemma 1.11 to $X \subseteq var_X(E)$ and $X \neq \emptyset$ then it follows that $|equ_E(X)| > |X|$. Then $|E'| > |X|$ since the definition of E' gives $|E'| = |equ_E(X)|$. Condition (3) imply an upper bound on $|E'|$,

$$|E'| = |E \cap equ_M(X)| \leq |equ_M(X)| = 1 + |X|. \tag{2.3}$$

From inequality (2.3) and $|E'| > |X|$ it follows that $equ_M(X) = E'$, hence $equ_M(X) \subseteq E$.

Suppose contrary that $X \backslash X_{E'} \neq \emptyset$. Now, condition (1) of the system M where $\bar{X} = X \backslash X_{E'}$ gives the inequality

$$|X \backslash X_{E'}| \leq |equ_M(X \backslash X_{E'})| - 1. \tag{2.4}$$

Consider the negation of the conclusion. Then $equ_M(X) \backslash E' \neq \emptyset$. E is an MSS set and $X_{E'} \neq \emptyset$. Then apply Lemma 1.11 where $\bar{X} = X_{E'}$ and it follows that

$$|X_{E'}| \leq |equ_E(X_{E'})| - 1. \tag{2.5}$$

Add inequality (2.4) and (2.5)

$$\begin{aligned} |X| = |X_{E'}| + |X \backslash X_{E'}| \leq \\ \leq |equ_E(X_{E'})| + |equ_M(X \backslash X_{E'})| - 2 \leq \\ \leq |equ_M(X)| - 2. \end{aligned} \tag{2.6}$$

The last inequality in (2.6) follows since $equ_E(X_{E'}) \cap equ_M(X \backslash X_{E'}) = \emptyset$. Condition (3) imply a contradiction $|X| + 2 \leq |equ_M(X)| = |X| + 1$. Hence, $equ_M(X) \subseteq E$. ■

Example 2.4 Consider again Example 2.3 and the output in Figure 2.4 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}	$\dot{f}_{yf}$	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	

(2.7)

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.

2.3 Finding MSS Sets

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_j be the current set of equations that is examined. The set of MSS sets found is denoted M_{alg4}. Then the following algorithm finds all MSS sets in M_{simp} if $M = M_{simp}$.

Algorithm 4.
Input: A structural model M.

1. Set $k = 1$ and $M_{alg4} = \varnothing$.
2. Choose equation e_k. Let $E_1 = \{e_k\}$ and $X_1 = \varnothing$.
3. Find all MSS sets that are subsets of M_k and include equation e_k.
   (a) Let $\tilde{X}_j = var_{X_u}(E_j) \backslash X_j$ be the unmatched variables.
   (b) If $\tilde{X}_j = \varnothing$, then E_j is an MSS set. Insert E_j into M_{alg4}.
   (c) Else take a remaining variable $\tilde{x}_j \in \tilde{X}_j$ and assign $\{\tilde{x}_j\}$ to e, i.e. let $X_{j+1} = X_j \cup \{\tilde{x}_j\}$. Let $\tilde{E}_j = equ_{M_k \backslash E_j}(\tilde{x}_j)$ be the remaining equations. For all equations e in $\tilde{E}_j$ let $E_{j+1} = E_j \cup \{e\}$, and goto (a).

4. If $k < n$ set $k = k + 1$ and goto number (2) .

Output: M_{alg4}

Theorem 2.8. $MSS_{alg4} = MSS(M)$.

Proof. To show the inclusion $MSS_{alg4} \subseteq MSS(M)$, we will take an arbitrary $E \in MSS_{alg4}$ and show that E is an MSS set. The set E is an MSS set if and only if $\forall e \in E$ there exist a perfect matching in $(E\backslash\{e\}, var_{X_u}(E))$ according to Theorem 1.6. The goal is to find a perfect matching in $(E\backslash\{e\}, var_{X_u}(E))$ for all e.

Number the equations in E as they were found in Algorithm 4, i.e. $E = \{e_1, \cdots, e_n\}$. Let $E_j = \{e_1, \cdots, e_j\}$ be the first j equations found. Note that when Algorithm 4 stores E in step 3b it holds that $j = n$. If $n = 1$, then $var_{X_u}(E_1) = \tilde{X}_1 = \varnothing$ in (3b) and $E_1 = \{e_1\}$ is an MSS set. Otherwise, if $n \geq 2$ take an arbitrary $e_\alpha \in E$. The next paragraph shows that there will be a perfect matching in $(E\backslash\{e_\alpha\}, var_X(E))$ for all $\alpha \in \{1, 2, \cdots, n\}$.

If $n \geq 2$ the algorithm finds a complete matching of $var_{X_u}(E)$ into E. This assignment is $\{e_2, x_1\}$, $\{e_3, x_2\}$, $\cdots$,$\{e_n, x_{n-1}\}$. If $\alpha = 1$ a perfect matching is the previous assignment. If $\alpha \neq 1$, then all variables but $x_{\alpha-1}$ have an assignment. The next paragraph shows that it is always possible to construct an augmenting path from e_α to e_1. This path defines a reassignment such that the new assignment is a perfect matching in $(E\backslash\{e_\alpha\}, var_{X_u}(E))$.

Algorithm 4 picks an equation e_i in step 3c only if x_{i-1} is included in e_i. In step 3c a search for x_{i-1} is performed only if $x_{i-1} \in var_{X_u}(E_{i-1})$ according to step 3a. The conclusion is that for any $i \in \{2, 3, \cdots, n\}$, it is possible to find an equation e_β such that $x_{i-1} \in var_{X_u}(e_\beta)$ and $\beta < i$. This is a sufficient condition to find an augmenting path from e_α to e_1.

Starting in equation e_α the assignment imply the first edge to x_{j-1}. From the previous paragraph there is an edge between e_{β_1} and x_{j-1} where $\beta_1 < j$. This can be repeated until $\beta_k = 1$. Since β_1 is finite and $\{\beta_i\}$ is a strictly decreasing list of natural numbers, it follows that k is finite. Reassign the equations and variables included in the augmenting path so $\{e_{\beta_1}, x_{\alpha-1}\}$, $\{e_{\beta_2}, x_{\beta_1-1}\}$, $\{e_{\beta_3}, x_{\beta_2-1}\}$, $\cdots$,$\{e_1, x_{\beta_k}\}$. This assignment is a perfect matching in $(E\backslash\{e_\alpha\}, var_{X_u}(E))$. Using Theorem 1.6 the conclusion is that $MSS_{alg4} \subseteq MSS(M)$.

The second part of the proof shows that $MSS(M) \subseteq MSS_{alg4}$. Take arbitrary $E \in MSS(M)$. Let $e_1 \in E$ be the first equation that Algorithm 4 picks in step 2. If $E = \{e_1\}$ then $var_{X_u}(e_1) = \varnothing$ and the algorithm finds the MSS set immediately in step 3b. If $\{e_1\} \subset E$ then according to Theorem 1.6 there is a perfect matching in $(E \backslash \{e_1\}, var_X(E))$. Take any perfect matching in $(E \backslash \{e_1\}, var_X(E))$. This perfect matching is (x_1, e_2), $(x_2, e_3) \cdots (x_{|E|-1}, e_{|E|})$. We will now show that the algorithm will find this perfect matching. The enumerations of the variables are defined step by step as they are found in Algorithm 4.

Since $\{e_1\}$ is not an MSS set then $var_X(e_1) \neq \varnothing$. The algorithm picks a $x \in var_X(e_1)$ in step 3c. This x is defined as x_1 by the algorithm. The given perfect matching assigns x_1 to e_2. This is only possible if $e_2 \in equ_{M_k}(x_1)$. Then $\tilde{E}_1 = equ_{M_k}(x_1)$ in step 3c. Finally step 3c will assign x_1 once at a time to all $e \in equ_{M_k}(x_1)$. Particularly the algorithm will assign x_1 to e_2.

Now, suppose that the algorithm has assigned $\{x_1, e_2\}$, $\{x_2, e_3\}$, $\cdots$,$\{x_i, e_{i+1}\}$ for any $1 \leq i \leq |E| - 2$. This means that step 3c is just done and the algorithm will start in step 3a again.

The current value of the variables are

$$\begin{aligned} E_{i+1} &= \{e_1, e_2, \cdots, e_{i+1}\} \\ X_{i+1} &= \{x_1, x_2, \cdots, x_i\}. \end{aligned}$$

In step 3a $\tilde{X}_{i+1} = var_{X_u}(E_{i+1}) \backslash X_{i+1}$. From the assumption it follows that E_{i+1} is not structurally singular, because $i \leq |E| - 2$, hence $var_{X_u}(E_{i+1}) \backslash X_{i+1} \neq \varnothing$. This implies that $\tilde{X}_i \neq \varnothing$. Hence it must be at least one variable in $\tilde{X}_i$. The variable that the algorithm picks is denoted x_{i+1}.

The variable x_{i+1} is assigned e_{i+2} according to the given matching. Then $e_{i+2} \in equ_{M_k}(x_{i+1})$ and especially $e_{i+2} \in equ_{M_k \backslash E_{i+1}}(x_{i+1})$. Hence $e_{i+2} \in \tilde{E}_{i+1}$ in step 3c. Since step 3c assign x_{i+1} to all $e \in \tilde{E}_{i+1}$ one at a time, the algorithm will particularly assign x_{i+1} to e_{i+2}.

Now, $E_{|E|} = \{e_1, \cdots, e_n\}$, $X_{|E|} = \{x_1, \cdots, x_{|E|-1}\}$, and Algorithm 4 starts at step 3a. Since E is an MSS set it follows that $|E| > |var_{X_u}(E)|$. The

From the definition of $\{x_1, \cdots, x_{|E|-1}\}$, it follows that $var_{X_u}(E) = \{x_1, \cdots, x_{|E|-1}\}$. Then $\tilde{X}_{|E|} = var_{X_u}(E) \backslash \{x_1, \cdots, x_{|E|-1}\} = \varnothing$. This is detected in step 3b and the algorithm conclude that $E_{|E|} = E$ is an MSS set. Hence $MSS(M) \subseteq MSS_{alg4}$.

■

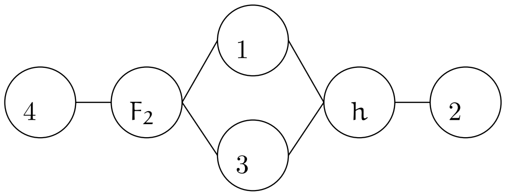

Figure 2.8: The structural model (2.8) viewed as a graph.

The next example shows how the search for MSS sets is performed in the Example 2.4.

Example 2.5 To simplify the notation, let the equations be renamed as

renamed equation	equation	unknown	fault
1	$\{e_1, e_2, \dot{e}_4, \dot{e}_5\}$	F_2, h	$f_u, f_{yf}, \dot{f}_{yf}$
2	e_3	h	f_{yh}
3	e_4	F_2, h	
4	e_5	F_2	f_{yf}

(2.8)

Algorithm 4 is explained using the structural model (2.8) shown as a graph in Figure 2.8. The algorithm order the equations according to their numbers and the variables for example as F_2 then h. First all MSS sets that include equation 1 are found. Set the current equation as $E_1 = \{1\}$, i.e. equation 1 will be used. Equation 1 includes the unknown variables F_2 and h, i.e. $\tilde{X}_1 = \{F_2, h\}$. The algorithm starts to find an equation that is able to eliminate the first unknown variable, i.e. in this case variable F_2. This can be done either with equation 3 or 4. Algorithm 4 starts to include equation 3, i.e. $E_2 = \{1, 3\}$ and $\tilde{X}_2 = \{h\}$. The only equation that is able to eliminate the unknown variable h is equation 2, because equation 3 is already used to eliminate F_2. Then $E_3 = \{1, 3, 2\}$ and $X_3 = \varnothing$, hence an MSS set is found. The stack of current equations E_j is throughout the entire algorithm:

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

The bold columns represent the MSS sets found. The MSS sets found are $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, and $\{2, 3, 4\}$ or in the original notations

$\{e_1, e_2, e_3, e_4, \dot{e}_4, \dot{e}_5\}$, $\{e_1, e_2, e_3, \dot{e}_4, e, \dot{e}_5\}$, $\{e_1, e_2, e_4, \dot{e}_4, e_5, \dot{e}_5\}$, and $\{e_3, e_4, e_5\}$.

Example 2.6 Another 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 stack of current equations, i.e. E_j 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.

2.4 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.

Example 2.7 Now follows the continuation of Example 2.5. The matrix in Figure 2.9 is the incidence matrix of the MSS sets produced of (2.8). If any equation in the MSS set i include fault j, the element (i, j) of the incidence matrix is equal to X. The derivatives of the faults are omitted in the incidence matrices. The third MSS set in

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

Figure 2.9: The incidence matrix of the MSS sets in Example 2.5.

Figure 2.10 (a) 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 are large it is not easy to see how good the diagnosis can be. 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 the detection capability of the diagnosis system. If fault j can explain the not satisfied consistency relations, then element (i, j) of fault matrix is equal to X. To get an upper limit of the diagnosability it is assumed that if a fault i is present then all consistency relations including fault i are not satisfied. No analytical consistency relations are known at this stage and therefore it is impossible to determine the true diagnosability. Instead upper limits of the diagnosability are calculated and shown in the fault matrices in the continuation of the report.

Example 2.8 The fault matrix of the incidence matrix in Example 2.7 is shown in Figure 2.10. Remember that the fault matrix shows the upper limit of the diagnosability. Consider the first row in the fault matrix. Suppose that the fault f_u is present. Then, the first three consistency relations are not satisfied in the best case according to the discussion above. 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 the third consistency relation is not 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 [9]. This algorithm returns a *maximal matching* [6] which is in block upper-triangular form. The diagonal blocks corresponds to strong Hall components of the adjacency

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

Figure 2.10: The fault matrix of the incidence matrix in Figure 2.9.

graph of the fault matrix. The interpretation is that the diagnostic system considered handles the faults in a diagonal block as equivalent faults. In the small example in Figure 2.10, the same matrix is returned after Dulmage-Mendelsohn permutations, which usually is not the case. The diagonal blocks are the 1×1 diagonal elements, i.e. all faults act different on the diagnostic system.

2.5 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 in the new model obtained. Apply the new model to Algorithm 1 step 1. An MSS set that is able to isolate faults 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 2.9 In Figure 2.10, the fault matrix shows that f_u and f_{yh} can not be isolated. The problem is that there is no consistency relation 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 step 1 in Algorithm 1.

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
$\{e_1, e_2, e_3, e_4, \dot{e}_4, e_5, \dot{e}_5, e_6\}$	X	X	

Figure 2.11: The incidence matrix for the decoupled model.

The model 1.1 is

equation	unknown						fault		known		
	F_1	F_2	h	$\dot{h}$	f_{yf}	$\dot{f}_{yf}$	f_u	f_{yh}	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					

(2.9)

The result is a new set of one MSS set $\{e_1, e_2, e_3, e_4, \dot{e}_4, e_5, \dot{e}_5, e_6\}$ in which f_{yf} is decoupled. The incidence matrix is showed in Figure 2.11 and the corresponding fault matrix is the identity matrix. This gives a possibility to detect and isolate all faults.

2.6 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 used for fault isolation. Small MSS sets are more robust and are usually sensitive to fewer faults which means that they have higher isolation capability. Therefore the goal must be to find a set of 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 becomes isolable from fault j. This test uses the fault matrix of the previous selected MSS sets. If the considered MSS set remove one or more entries in fault matrix, the MSS set is selected. In this way the final output from Algorithm 1 will be the most robust set of MSS sets with highest possible diagnosis capability.

Example 2.10 The last part in the structural analysis is to choose MSS sets. The MSS sets in Example 2.9 are ordered in increasing size as

MSS	f_u	f_{yh}	f_{yf}
$\{e_3, e_4, e_5\}$		X	X
$\{e_1, e_2, e_4, \dot{e}_4, e_5, \dot{e}_5\}$	X		X
$\{e_1, e_2, e_3, e_4, \dot{e}_4, \dot{e}_5\}$	X	X	X
$\{e_1, e_2, e_3, \dot{e}_4, e_5, \dot{e}_5\}$	X	X	X
$\{e_1, e_2, e_3, e_4, \dot{e}_4, e_5, \dot{e}_5, e_6\}$	X	X	

The MSS sets finally chosen are

MSS	f_u	f_{yh}	f_{yf}
$\{e_3, e_4, e_5\}$		X	X
$\{e_1, e_2, e_4, \dot{e}_4, e_5, \dot{e}_5\}$	X		X
$\{e_1, e_2, e_3, e_4, \dot{e}_4, e_5, \dot{e}_5, e_6\}$	X	X	

Chapter 3

Industrial Example: A Paper Plant

This industrial example from ABB is a stock preparation and broke treatment system of a paper plant. The system is used for mixing and purifying recycled paper for production of new paper. An overview of the system is shown in Figure 3.1.

3.1 System Description

After the preparation step the purified paper mixture is transferred to the *screen*. In the screen it is important that the mixture has a correct concentration of paper fibers and not exceed a critical pressure. The system starts with recycled paper and water. The recycled paper has a high concentration of paper fibers. The two fluids are mixed in the *pulper* tank to a correct concentration. Looking in the right part of Figure 3.1, the *cyclone* purifies the paper mixture. This is done by spinning the fluid in the cyclone. The result is that large particles are collected in the bottom of the cyclone and the clean paper mixture is collected in the top. A drawback with this method is that the purified mixture obtains a high pressure. To limit the outflow pressure from this part of the process, there is a pipe going back to a tank. When this opens the pressure in the outflow mixture decreases. The return of the

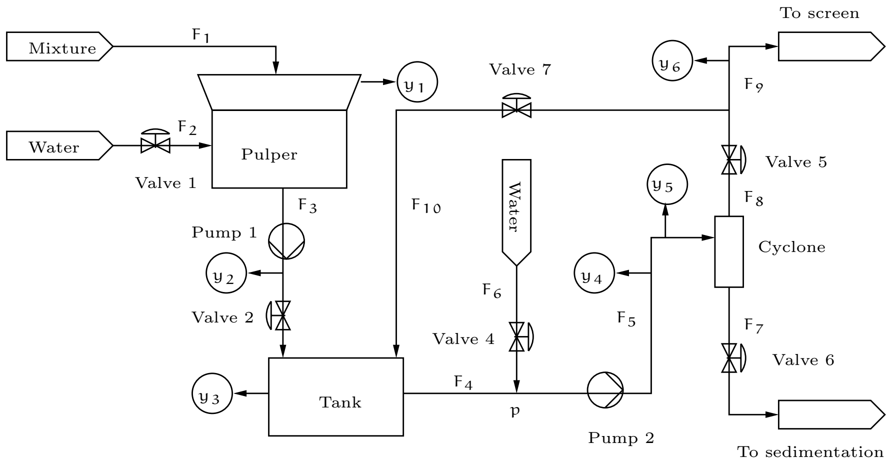

Figure 3.1: A stock preparation and broke treatment system of a paper plant

fluid increases the concentration in the tank. Therefore the mixture is diluted with water before entering the cyclone. For a more detailed description, see [1].

3.2 Model Description

Most parts of the system are nonlinear. It is only the tank and the pulper that are considered to be dynamic. The system has 4 states: the volumes x_1 and x_3 and concentrations x_2 and x_4 in the pulper and in the tank respectively. 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 into and out from this system are known, i.e F_1, F_2, F_6, F_7, and F_9 are known. Moreover the concentrations of the fluids flowing into the system are constant and known, i.e. c_1, c_2, and c_6 are known. There are 6 valves with control signals u_i, where $i \in \{1, 2, 4, 5, 6, 7\}$ and two pumps that have actuator signals z_{p1} and z_{p2}.

There are 21 faults that are analyzed. All sensors can have a constant offset fault f_{yi}, $i \in \{1, 2, 4, 5, 6, 7\}$. All valves can have a constant offset in the actuator signal f_{ui}, $i \in \{1, 2, 4, 5, 6, 7\}$. Clogging can occur in the pipes near the valves f_{cvi}, $i \in \{1, 2, 4, 5, 6, 7\}$ and also directly after the tank f_{cv8}. Finally, the pumps can have a constant offset on the actuator signal f_{p1} and f_{p2}.

The system is described by 29 equations. The model equations are

$$
\begin{array}{l|l}
e_1 & \dot{x}_1 - e_1(F_1 + F_2 - F_3) = 0 \\
e_2 & \dot{x}_2 - \frac{e_1(F_1(c_1+f_{y2}-y_2)+F_2(c_2+f_{y2}-y_2))}{y_1-f_{y1}} = 0 \\
e_3 & \dot{x}_3 - e_2(F_3 + F_{10} - F_4) = 0 \\
e_4 & \dot{x}_4 + \frac{e_2(F_{10}(f_{y4}+x_4-y_4)+F_3(f_{y2}+x_4-y_2))}{y_3-f_{y3}} = 0 \\
e_5 & g_1 - a_1 F_1^2 = 0 \\
e_6 & g_3 - F_2^2(a_2 + a_3 + b_1(f_{cv1} + z_{u1})) = 0 \\
e_7 & k_1(y_1 - f_{y1}) + d_{11} - 1 + f_{p1}^2 z_{p1} - F_3^2(bcv3 + a_4 + a_5 \\
 & +a_6 + a_7 + b_2 f_{cv2} + b_2 z_{u2}) = 0 \\
e_8 & p + k_2(y_3 - f_{y3}) - b_8 f_{cv8} F_4^2 - a_8 F_4^2 - p = 0 \\
e_9 & y_5 + d_{12} - 1 + f_{p2}^2 z_{p2} - a_{11} F_5^2 - a_{10} F_5^2 - f_{y5} - p = 0 \\
e_{10} & p + g_{27} - b_4 F_6^2(f_{cv4} + z_{u4}) - a_9 F_6^2 - p = 0 \\
e_{11} & y_5 - b_6 F_7^2(f_{cv6} + z_{u6}) - f_{y5} - g_{21} - p = 0 \\
e_{12} & f_{y6} + y_5 - b_5 F_8^2(f_{cv5} + z_{u5}) - y_6 - f_{y5} = 0 \\
e_{13} & y_6 - a_{13} F_9^2 - f_{y6} - g_{23} - p = 0 \\
e_{14} & y_6 - b_7 F_{10}^2(f_{cv7} + z_{u7}) - a_{14} F_{10}^2 - a_{12} F_{10}^2 - f_{y6} - p = 0 \\
e_{15} & f_{y4} + \frac{c_6 F_6 + F_4 x_4}{F_4 + F_6} - y_4 = 0 \\
e_{16} & -1 + \frac{d_{21} F_3^2}{(-1+f_{p1})^2} + z_{p1}^2 = 0 \\
e_{17} & -1 + \frac{d_{22} F_5^2}{(-1+f_{p2})^2} + z_{p2}^2 = 0 \\
e_{18} & -1 + f_{u1} + u_1^2 z_{u1} = 0 \\
e_{19} & -1 + f_{u2} + u_2^2 z_{u2} = 0 \\
e_{20} & -1 + f_{u4} + u_4^2 z_{u4} = 0 \\
e_{21} & -1 + f_{u5} + u_5^2 z_{u5} = 0 \\
e_{22} & -1 + f_{u6} + u_6^2 z_{u6} = 0 \\
e_{23} & -1 + f_{u7} + u_7^2 z_{u7} = 0 \\
e_{24} & F_4 + F_6 - F_5 = 0 \\
e_{25} & F_5 - F_8 - F_7 = 0 \\
e_{26} & F_8 - F_{10} - F_9 = 0 \\
e_{27} & f_{y1} + x_1 - y_1 = 0 \\
e_{28} & f_{y2} + x_2 - y_2 = 0 \\
e_{29} & f_{y3} + x_3 - y_3 = 0
\end{array}
\tag{3.1}
$$

Equations $e_1, \dots, e_4$ describe the dynamics; $e_5, \dots, e_{14}$ 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; $e_{18}, \dots, e_{23}$ are valve equations; e_{24}, e_{25}, e_{26} are flow equations, and finally e_{27}, e_{28}, e_{29} are sensor equations for sensor 1, 2, and 3. Furthermore there are 21 equations one for each fault expressed as $\dot{f}_i = 0$. The structural model for equation system (3.1) can be viewed in the matrix in Figure 3.2.

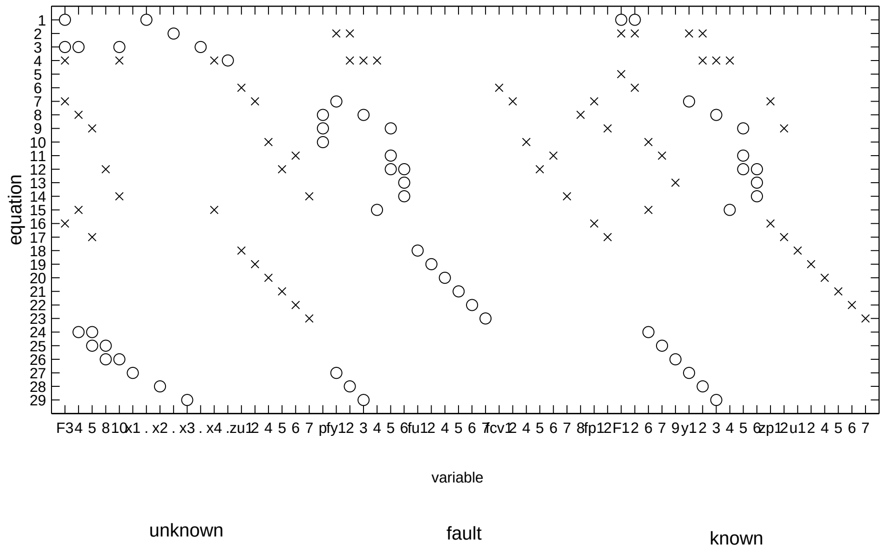

Figure 3.2: Structural model for the set of equations (3.1). The circles denote that the corresponding variable is linearly dependent and the crosses denote nonlinearly dependent variables. Because of space considerations the variables are abbreviated. For example is the first 7 variables are F_3, F_4, F_5, F_8, F_{10}, x_1, and $\dot{x}_1$.

The circles denote that the corresponding variable is linearly included. The variables are divided as follows

type of variable	variable
X_u	$F_3, F_4, F_5, F_8, F_{10}, x_1, \dot{x}_1, x_2, \dot{x}_2, x_3, \dot{x}_3, x_4, \dot{x}_4, z_{u1}, z_{u2}, z_{u4}, z_{u5}, z_{u6}, z_{u7}, p$
F	$f_{y1}, f_{y2}, f_{y3}, f_{y4}, f_{y5}, f_{y6}, f_{u1}, f_{u2}, f_{u4}, f_{u5}, f_{u6}, f_{u7}, f_{cv1}, f_{cv2}, f_{cv4}, f_{cv5}, f_{cv6}, f_{cv7}, f_{cv8}, f_{p1}, f_{p2}$
Y	$F_1, F_2, F_6, F_7, F_9, y_1, y_2, y_3, y_4, y_5, y_6, z_{p1}, z_{p2}, u_1, u_2, u_4, u_5, u_6, u_7$

3.3 Differentiating the Model

The highest order of derivatives that can be used for all known variables is 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 structural model in Figure 3.2. The result is that all equations except equation 1, 2, 3, and 4 are differentiated. This results in additionally 25 differentiated equations showed in Figure 3.3. Note how the knowledge concerning linear dependence influences the structural model in Figure 3.3 by comparing it with the original structural model in 3.2. For example, x_3 is linearly contained in e_{29}, hence $var_{X_u}(\dot{e}_{29}) = \{\dot{x}_3\}$ and z_{u1} is nonlinearly contained in e_6 and then it follows that $var_{X_u}(\dot{e}_6) = \{z_{u1}, \dot{z}_{u1}\}$.

3.4 Simplifying the Model

In the first step of simplification applied to the matrix in Figure 3.3, the equations e_{27}, e_{28}, and e_{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 $\dot{F}_3$, $\dot{F}_{10}$, x_1, x_2, x_3, x_4, $\dot{x}_4$, $\dot{z}_{u1}$, $\dot{z}_{u2}$, $\dot{z}_{u4}$, $\dot{z}_{u5}$, $\dot{z}_{u6}$, and $\dot{z}_{u7}$ can be eliminated. The equations that form groups are $\{\dot{e}_7, \dot{e}_{16}, \dot{e}_{19}\}$, $\{\dot{e}_{14}, \dot{e}_{23}, \dot{e}_{26}\}$, $\{e_1, \dot{e}_{27}\}$, $\{e_2, \dot{e}_{28}\}$, $\{e_3, \dot{e}_{29}\}$, $\{e_4, e_{15}, \dot{e}_{15}\}$, $\{\dot{e}_6, \dot{e}_{18}\}$, $\{\dot{e}_{10}, \dot{e}_{20}\}$, $\{\dot{e}_{12}, \dot{e}_{21}\}$, and $\{\dot{e}_{11}, \dot{e}_{22}\}$. The simplified structural model is showed in Figure 3.4. Note the simplification of the model by comparing Figure 3.3 and Figure 3.4. The simplification reduces the model from 54 equations to 38 equations and reduces the unknown variables from 32 to 16. To compare the reduction computational complexity of finding the MSS set the number of times the algorithm

To give an example of the reduction of the computational complexity using this simplification step, the algorithm has found all MSS sets in the structural model in Figure 3.3 without using the simplification step. The number of times that the algorithm asks for a row or a column in the structural model is computed. The result is that the simplification step required 88 calls and Algorithm 4 335,107 calls. When the simplification step was not first applied, the Algorithm 4 used 1,872,753 calls. This result indicates that simplification is cheap and considerable decreases the computational complexity of Algorithm 4. The next step is to find all MSS sets in the simplified model.

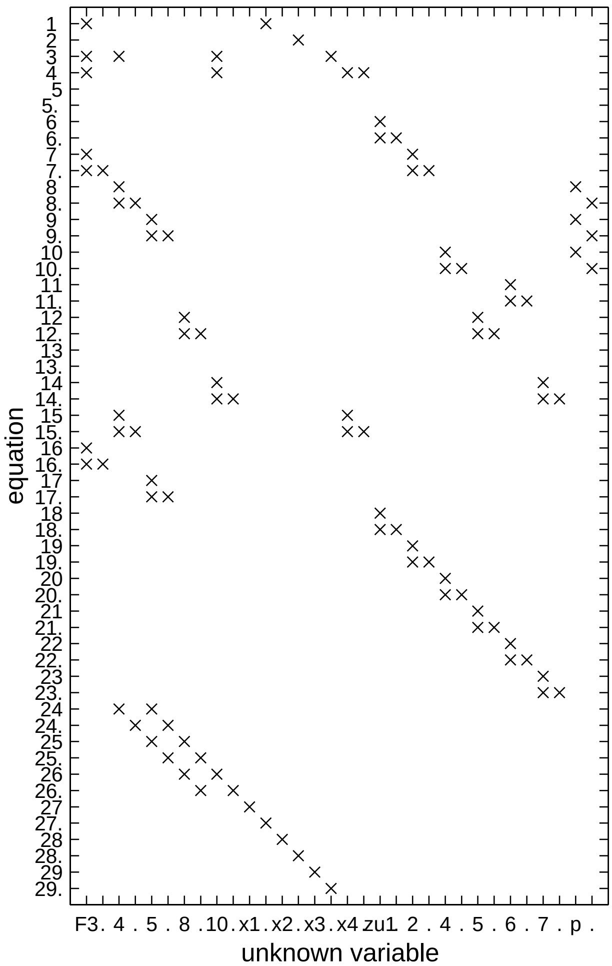

Figure 3.3: The resulting structural model when the differentiation step is applied to the structural model in Figure 3.2. The variables F and Y are not shown. Differentiated equations are denoted with a dot after the number.

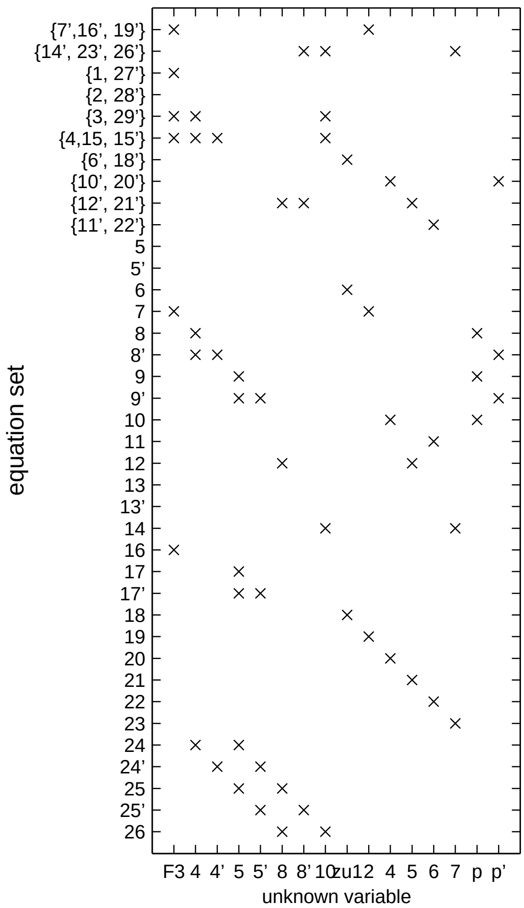

Figure 3.4: The simplified structural model.

3.5 Finding MSS Sets

Algorithm 4 is applied to the simplified model. The algorithm returns 35770 MSS sets that are contained in the simplified model. The five smallest MSS sets are $\{e_5\}$, $\{\dot{e}_5\}$, $\{e_{13}\}$, $\{\dot{e}_{13}\}$, and $\{e_2, \dot{e}_{28}\}$. The largest MSS sets consists of 23 equations. Now the diagnosability of the MSS sets found is analyzed.

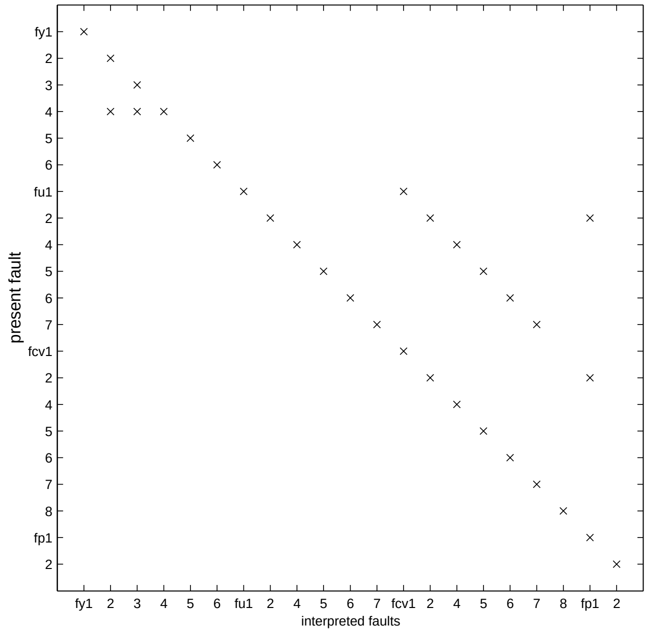

Figure 3.5: The fault matrix of the MSS sets corresponding to Figure 3.4

3.6 Analyzing Diagnosability

The diagnosability of the MSS sets found can be seen in the fault matrix in Figure 3.5. All faults are detectable with the MSS sets found in the previous step. The fault f_{ui} is not isolable from f_{cvi} where $i \in \{1, 2, 4, 5, 6, 7\}$, i.e. a constant offset in the actuator signal to valve i can always be explained as clogging in valve i. Moreover f_{y4} is not isolable from f_{y2} and f_{y3}. Finally f_{u2} and f_{cv2} are not isolable from f_{p1}.

3.7 Decoupling Faults

In the fault matrix shown in Figure 3.5 the columns that have non-diagonal entries are collected. These columns correspond to the faults that will be decoupled, i.e. f_{y2}, f_{y3}, f_{cv1}, f_{cv2}, f_{cv4}, f_{cv5}, f_{cv6}, f_{cv7}, and f_{p1}. First f_{y2} will be decoupled by applying Algorithm 1 step 1 to the original model, but this time the fault variable f_{y2} and its derivatives are considered to be unknown variables. The goal is to find an MSS set that decouple f_{y2} and is sensitive to fault f_{y4}. An MSS set with this property increases the diagnosability because it gives the possibility to isolate fault f_{y4} from f_{y2}. This implies that the cross in the row corresponding to f_{y4} and the column corresponding to f_{y2} in the fault matrix in Figure 3.5 is removed. The result of decoupling fault f_{y2} is that 26959 new MSS sets are found. The smallest MSS set of these new MSS sets that isolate fault f_{y4} from f_{y2} is $\{e_2, e_3, e_4, e_{15}, \dot{e}_{15}, e_{16}, e_{17}, \dot{e}_{17}, e_{24}, \dot{e}_{24}, \dot{e}_{28}, \dot{e}_{29}\}$.

Next also the faults f_{y3}, f_{cv1}, f_{cv2}, f_{cv4}, f_{cv5}, f_{cv6}, f_{cv7}, and f_{p1} are decoupled. The results are:

decoupled fault	sensitive to fault	smallest MSS set with desired property
f_{y2}	f_{y4}	$\{e_2, e_3, e_4, e_{15}, \dot{e}_{15}, e_{16}, e_{17}, \dot{e}_{17}, e_{24}, \dot{e}_{24}, \dot{e}_{28}, \dot{e}_{29}\}$
f_{y3}	f_{y4}	$\{e_4, e_8, e_9, e_{14}, e_{15}, \dot{e}_{15}, e_{16}, e_{17}, \dot{e}_{17}, e_{23}, e_{24}, \dot{e}_{24}\}$
f_{cv1}	f_{u1}	$\{e_6, \dot{e}_6, e_{18}, \dot{e}_{18}\}$
f_{cv2}	f_{u2}	$\{e_7, \dot{e}_7, e_{16}, \dot{e}_{16}, e_{19}, \dot{e}_{19}\}$
f_{cv4}	f_{u4}	$\{e_9, \dot{e}_9, e_{10}, \dot{e}_{10}, e_{17}, \dot{e}_{17}, e_{20}, \dot{e}_{20}\}$
f_{cv5}	f_{u5}	$\{e_{12}, \dot{e}_{12}, e_{17}, \dot{e}_{17}, e_{21}, \dot{e}_{21}, e_{25}, \dot{e}_{25}\}$
f_{cv6}	f_{u6}	$\{e_{11}, \dot{e}_{11}, e_{22}, \dot{e}_{22}\}$
f_{cv7}	f_{u7}	$\{e_{12}, \dot{e}_{12}, e_{14}, \dot{e}_{14}, e_{21}, \dot{e}_{21}, e_{23}, \dot{e}_{23}, e_{26}, \dot{e}_{26}\}$
f_{p1}	f_{u2}	$\{e_1, e_7, e_{16}, e_{19}, \dot{e}_{27}\}$

With those additional MSS sets all faults are detectable and isolable. The next step is to select a small subset of all found MSS sets that have full diagnosability.

3.8 Selecting a Subset of MSS Sets

First the MSS sets are reordered in increasing size. The first 6 MSS sets in the reordered list are

MSS		
1	e_5	
2	$\dot{e}_5$	
3	e_{13}	
4	$\dot{e}_{13}$	
5	e_2	$\dot{e}_{28}$
6	e_6	e_{18}

(3.2)

Then the algorithm selects those MSS sets that increase the diagnosability starting from the smallest MSS sets in (3.2). According to Figure 3.2 neither the first $\{e_5\}$ nor the second MSS set $\{\dot{e}_5\}$ is sensitive to any fault and is therefore not selected. The third MSS set $\{e_{13}\}$ is sensitive only to f_{y6}, i.e. f_{y6} can be detected and isolated. The diagnosability is improved with this MSS set and therefore it is selected. The fourth MSS set is not sensitive for any fault and is therefore not selected. The 5th MSS set detects f_{y1} and f_{y2} and isolate f_{y1} and f_{y2} from all other faults. This MSS set is selected. When all MSS sets have been analyzed, 36 MSS sets are selected. These are

MSS												
1	e_{13}											
2	e_2	$\dot{e}_{28}$										
3	e_6	e_{18}										
4	e_{11}	e_{22}										
5	e_1	e_{16}	$\dot{e}_{27}$									
6	e_6	$\dot{e}_6$	$\dot{e}_{18}$									
7	e_{11}	$\dot{e}_{11}$	$\dot{e}_{22}$									
8	e_{11}	e_{22}	$\dot{e}_{22}$									
9	e_7	e_{16}	e_{19}									
10	e_8	e_9	e_{17}	e_{24}								
11	e_9	e_{10}	e_{17}	e_{20}								
12	e_{12}	e_{17}	e_{21}	e_{25}								
13	e_6	$\dot{e}_6$	e_{18}	$\dot{e}_{18}$								
14	e_{11}	$\dot{e}_{11}$	e_{22}	$\dot{e}_{22}$								
15	e_7	$\dot{e}_7$	e_{16}	$\dot{e}_{16}$	$\dot{e}_{19}$							
16	$\dot{e}_7$	e_{16}	$\dot{e}_{16}$	e_{19}	$\dot{e}_{19}$							
17	e_8	e_{10}	e_{17}	e_{20}	e_{24}							
18	e_{12}	e_{14}	e_{21}	e_{23}	e_{26}							
19	e_{14}	e_{17}	e_{23}	e_{25}	e_{26}							
20	e_1	e_7	e_{16}	e_{19}	$\dot{e}_{27}$							
21	$\dot{e}_8$	$\dot{e}_9$	e_{17}	$\dot{e}_{17}$	e_{24}	$\dot{e}_{24}$						
22	e_7	$\dot{e}_7$	e_{16}	$\dot{e}_{16}$	e_{19}	$\dot{e}_{19}$						
23	e_9	$\dot{e}_9$	e_{10}	$\dot{e}_{10}$	e_{17}	$\dot{e}_{17}$	$\dot{e}_{20}$					
24	e_{12}	$\dot{e}_{12}$	e_{17}	$\dot{e}_{17}$	$\dot{e}_{21}$	e_{25}	$\dot{e}_{25}$					
25	e_8	e_{10}	e_{12}	e_{20}	e_{21}	e_{24}	e_{25}					
26	$\dot{e}_{12}$	e_{17}	$\dot{e}_{17}$	e_{21}	$\dot{e}_{21}$	e_{25}	$\dot{e}_{25}$					
27	e_9	$\dot{e}_9$	e_{10}	$\dot{e}_{10}$	e_{17}	$\dot{e}_{17}$	e_{20}	$\dot{e}_{20}$				
28	e_{12}	$\dot{e}_{12}$	e_{17}	$\dot{e}_{17}$	e_{21}	$\dot{e}_{21}$	e_{25}	$\dot{e}_{25}$				
29	e_{12}	$\dot{e}_{12}$	e_{14}	$\dot{e}_{14}$	e_{21}	$\dot{e}_{21}$	$\dot{e}_{23}$	e_{26}	$\dot{e}_{26}$			
30	e_{14}	e_{17}	$\dot{e}_{17}$	e_{23}	$\dot{e}_{23}$	e_{25}	$\dot{e}_{25}$	e_{26}	$\dot{e}_{26}$			
31	e_3	e_4	e_{15}	$\dot{e}_{15}$	e_{16}	e_{17}	e_{24}	$\dot{e}_{24}$	$\dot{e}_{29}$			
32	e_{12}	$\dot{e}_{12}$	e_{14}	$\dot{e}_{14}$	e_{21}	$\dot{e}_{21}$	e_{23}	$\dot{e}_{23}$	e_{26}	$\dot{e}_{26}$		
33	e_1	e_3	e_4	e_{15}	$\dot{e}_{15}$	e_{17}	$\dot{e}_{17}$	e_{24}	$\dot{e}_{24}$	$\dot{e}_{27}$	$\dot{e}_{29}$	
34	e_3	e_4	e_8	$\dot{e}_8$	e_{10}	$\dot{e}_{10}$	e_{15}	$\dot{e}_{15}$	e_{16}	e_{20}	$\dot{e}_{20}$	$\dot{e}_{29}$
35	e_4	e_8	e_9	e_{14}	e_{15}	$\dot{e}_{15}$	e_{16}	e_{17}	$\dot{e}_{17}$	e_{23}	e_{24}	$\dot{e}_{24}$
36	e_2	e_3	e_4	e_{15}	$\dot{e}_{15}$	e_{16}	e_{17}	$\dot{e}_{17}$	e_{24}	$\dot{e}_{24}$	$\dot{e}_{28}$	$\dot{e}_{29}$

(3.3)

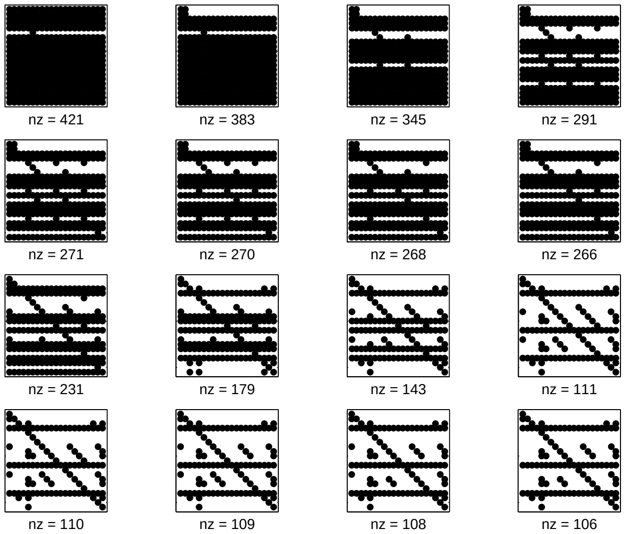

Figure 3.6: The sequence the first 16 fault matrices defined by adding one MSS at the time from (3.3). The numbers denote the number of crosses in each fault matrix.

The first three MSS sets can be recognized from (3.2). In Figure 3.6 the sequence of fault matrices defined by adding one MSS at a time from (3.3) is shown. Note that the number of crosses in each fault matrix can be interpreted as inversely proportional to the isolability. From the 36 MSS sets the incidence matrix in Figure 3.7 is obtained.

3.9 Generating Consistency Relations

In this report 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. The consistency relations corresponding to the 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 for elimination.

All MSS sets with 7 or less equations were easily eliminated to a consistency relation. The consistency relations from the MSS set 23, 24, 25 and 26 were obtained from the elimination function, but

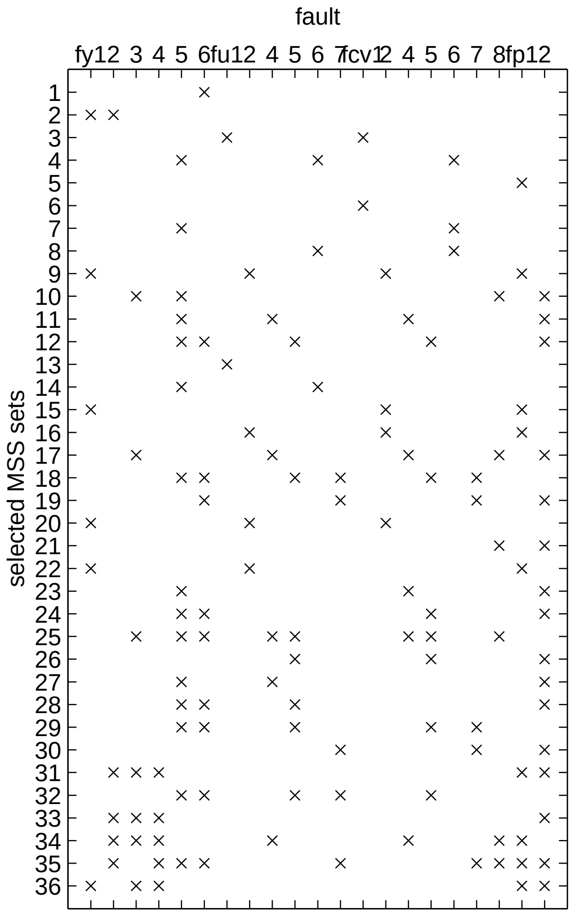

Figure 3.7: The incidence matrix of the selected MSS sets corresponding to Figure 3.4

were not useful because of bad numerical properties. However, small MSS sets make the largest contribution to the isolability. To see this, Figure 3.8 shows the percentage of full isolability when only the first n selected MSS sets in (3.3) are used. The number n is plotted on the x-axis. It is clear that the diagnosability reduces slightly, without using large MSS sets that are difficult to calculate. A few examples of

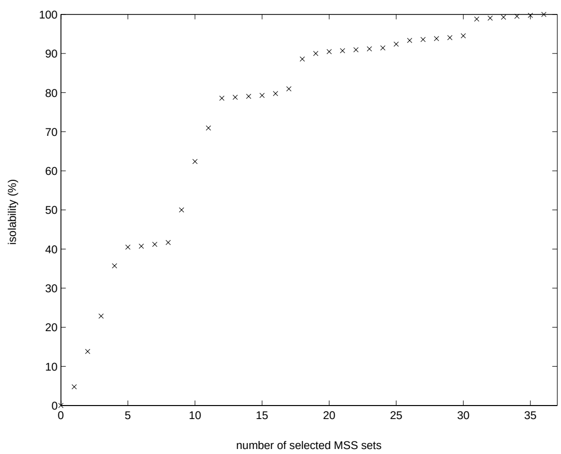

Figure 3.8: The label on the x-axis indicate how many of the first selected MSS sets in (3.3) that are used. The y-axis shows the percentage of full isolability with those MSS sets.

consistency relations derived by Mathematica are

MSS	consistency relations with faults
1.	$y_6(t) = 292890. + f_{y6}(t) + 3.13291 * 10^{-7} * F_9(t)^2$
2.	$F_1(t) * (-22173. - 389000. * f_{y2}(t) + 389000.y_2(t))$
	$+F_2(t) * (-389. - 389000. * f_{y2}(t) + 389000.y_2(t))$
	$+6.\,10^{11} * y_1(t) * (y_2)'(t) = 6. * 10^{11}\, f_{y1}(t) * (y_2)'(t)$
5.	$f_{p1}(t) \neq 1. \wedge 8.796\,10^{17} + 8.796\,10^{17}\, f_{p1}(t)^2 +$
	$1.7592\,10^{18}\, f_{p1}(t)\, z_{p1}(t)^2 +$
	$1.02649\,10^{15}\, F_1(t)\,(y_1)'(t) + 1.02649\,10^{15}\, F_2(t)\,(y_1)'(t)$
	$= 1.7592\,10^{18}\, f_{p1}(t) + 3.32755\,10^8\, F_1(t)^2 +$
	$6.6551\,10^8\, F_1(t)\, F_2(t) + 3.32755\,10^8\, F_2(t)^2 +$
	$8.796\,10^{17}\, z_{p1}(t)^2 +$ (3.4)
	$8.796\,10^{17}\, f_{p1}(t)^2\, z_{p1}(t)^2 +$
	$7.9164\,10^{20}\, (y_1)'(t)^2$
6.	$f_{u1}(t)^2 \left(-1.21562\,10^{34} - 9.35281\,10^7\, F_2(t)^2\right) (F_2)'(t) +$
	$f_{u1}(t)\,((-2.43123\,10^{34} - 1.87056\,10^8\, F_2(t)^2)\, u_1(t)\,(F_2)'(t) +$
	$F_2(t)\,(1.21562\,10^{34} +$
	$(-1.33226\,10^{24} - 7.19341\,10^{25}\, f_{cv1}(t))\, F_2(t)^2)\,(u_1)'(t)) +$
	$u_1(t)\,((-1.21562\,10^{34} - 9.35281\,10^7\, F_2(t)^2)\, u_1(t)\,(F_2)'(t) +$
	$F_2(t)\,(1.21562\,10^{34} +$
	$(-1.33226\,10^{24} - 7.19341\,10^{25}\, f_{cv1}(t))\, F_2(t)^2)\,(u_1)'(t)) = 0$

The computational form of these consistency relations are

MSS	computational form of some consistency relations
1.	$y_6(t) = 292890. + 3.13291\,10^{-7}\,F_9(t)^2$
2.	$F_1(t)\,(-22173. + 389000.\,y_2(t)) +$
	$F_2(t)\,(-389. + 389000.\,y_2(t)) + 6.\,10^{11}\,y_1(t)\,(y_2)'(t) = 0$
5.	$8.796\,10^{17} + 1.02649\,10^{15}\,F_1(t)\,(y_1)'(t)+$
	$1.02649\,10^{15}\,F_2(t)\,(y_1)'(t) = 3.32755\,10^8\,F_1(t)^2+$
	$6.6551\,10^8\,F_1(t)\,F_2(t) + 3.32755\,10^8\,F_2(t)^2+$
	$8.796\,10^{17}\,z_{p1}(t)^2 + 7.9164\,10^{20}\,(y_1)'(t)^2$
6.	$u_1(t)\,((-1.21562\,10^{34} - 9.35281\,10^7\,F_2(t)^2)\,u_1(t)\,(F_2)'(t)+$
	$F_2(t)\,\left(1.21562\,10^{34} - 1.33226\,10^{24}\,F_2(t)^2\right)\,(u_1)'(t)) = 0$

(3.5)

The consistency relations are not normalized and therefore some coefficients are large. Furthermore the variables are not scaled and therefore big differences in the order of magnitude of coefficients occurs. For some simulation results utilizing consistency relations in this industrial example see [1].

Chapter 4
Conclusion

This report 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 1.2 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 2.1, 2.4, and 2.8. 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] Jonas Biteus. Diagnosis of fluid systems utilizing gröbner bases and filtering of consistency relations. Technical Report LiTH-ISY-EX-3237, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, 2001.

[2] E. Carpanzano and C. Maffezzoni. Symbolic manipulation techniques for model simplification in object-oriented modelling of large scale continuous systems. *Mathematics and Computers in Simulations*, (48):133–150, 1998.

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

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

[5] R. P. Grimaldi. *Discrete and Combinatorial Mathematics an Applied Introduction.* Number ISBN 0-201-60044-7. Addison-Wesley publishing company, 1994.

[6] F. Harary. *Graph theory.* Addison-Wesley publishing company, 1969.

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

[8] J. De Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 1987.

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

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

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

[12] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 1987.

[13] L. Travé-Massuyès, T. Escobet, and R. Milne. Model-based diagnosibility and sensor placement. application to a frame 6 gas turbine subsystem. In D. T. Dupre' S. Mcilraith, editor, *DX01 twelfth international workshop on princpals of diagnosis*, pages 205–212, 2001.