A Structural Algorithm for Finding Testable Sub-models and Multiple Fault Isolability Analysis

Mattias Krysander[1], Jan Åslund[1], and Erik Frisk[1]

[1] *Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden*
{*matkr,jaasl,frisk*}*@isy.liu.se*

Abstract

Structural methods have previously been used to perform isolability analysis and finding testable sub-models, so called Minimal Structurally Overdetermined (MSO) sets, Analytical Redundancy Relations (ARR), or Possible Conflicts (PC). The number of MSO sets grows exponentially in the degree of redundancy making the task of computing MSO sets intractable for systems with high degree of redundancy. This paper describes an efficient graph-theoretical algorithm for computing a similar, but smaller, set of testable submodels called Test Equation Supports (TES). A key difference, compared to an MSO based approach, is that the influence of faults is taken into account and the resulting number of testable models as well as the computational complexity of finding them can be reduced significantly without reducing the possible diagnosis performance. It is shown that the TESs in a direct way characterize the complete multiple fault isolability property of a model and thus extends previous structural approaches from the single-fault case.

1 INTRODUCTION

Structural methods have previously been used in the field of model-based diagnosis to, for example, find testable subsets of equations in a model and to perform fault isolability analysis. This paper addresses the two mentioned applications of structural methods.

Many works, e.g. those cited in (Armengol *et al.*, 2009), have proposed different ways of finding testable subsets of equations such as (Krysander *et al.*, 2008; Gelso *et al.*, 2008) based on Minimal Structurally Overdetermined (MSO) sets, (Pulido and Alonso-González, 2004) based on Possible Conflicts, and (Travé-Massuyès *et al.*, 2006) based on Structural Analytical Redundancy Relations. All these concepts are related to MSO sets, which are minimal sets of equations containing redundancy.

A problem with these approaches is that the number of MSO sets grows exponentially in the degree of redundancy of the model. Thus, if a system has many sensors, the number of MSO sets will be large and it will not be possible to neither compute all MSO sets nor to design residuals for each of them.

However, in these cases it might not be necessary from a diagnosis point of view to construct and use all possible tests since there might exist a significantly smaller number of tests with sufficient capability of distinguishing between different faults. Instead of searching for all MSO sets, we propose to search for a smaller set of testable models, so called Test Equation Supports (TESs), where in addition to redundancy also the influence of faults is taken into account. By including fault information, the resulting number of testable models as well as the computational complexity of finding them can be reduced without reducing the possible diagnosis performance.

Regarding isolability analysis, this work extends the structural isolability analysis given in (Krysander and Frisk, 2008) for single faults to the case of multiple faults in a computationally efficient way by showing that the set of all TESs characterizes the multiple fault isolability. The results fit into the framework of multiple fault isolability analysis described in (Pucel *et al.*, 2009).

In conclusion, key contributions of this paper are to motivate and define the new concept of TESs, develop an efficient algorithm[1] for computing these, and to show that the TESs characterize multiple fault isolability.

2 TES MOTIVATION

This section first introduces and describes the intuition of the basic concepts of TES and the closely

[1] A Matlab implementation is available at `http://www.fs.isy.liu.se/Software/TestModelTool/`.

related concept Test Support (TS). A main objective of this section is then to show how TSs are incorporated in the important problems of multiple fault isolability analysis and test selection.

To introduce and motivate definitions and results, consider the following small state-space model with 3 states, 3 measurements, and 5 faults:

$$\begin{array}{ll} e_1: & \dot{x}_1 = -x_1 + u + f_1 \\ e_2: & \dot{x}_2 = x_1 - 2x_2 + x_3 + f_2 \\ e_3: & \dot{x}_3 = x_2 - 3x_3 \\ e_4: & y_1 = x_2 + f_3 \\ e_5: & y_2 = x_2 + f_4 \\ e_6: & y_3 = x_3 + f_5 \end{array} \tag{1}$$

where x_i represent the unknown variables, u and y_i the known variables, and f_i the faults to be monitored. For simplicity, this model is linear but the objective of the paper is to derive methods applicable to general non-linear model descriptions. In (1), faults are modeled as fault signals f_i that in the fault free case are equal to zero, i.e., $f_i = 0$. In a faulty case, e.g. if fault i is present, then nothing is assumed about the signal f_i. An alternative way of representing faults is by introducing components as follows. Let fault i correspond to component c_i such that $f_i = 0$ if and only if c_i is OK, otherwise c_i will be not OK. With this notation, equation e_1 including fault 1 can be written as

$$c_1 = \text{OK} \rightarrow \dot{x}_1 = -x_1 + u$$

Any of these two fault representations is equally applicable, but we will use the fault signal representation here.

2.1 Residuals and TSs

Diagnosis can be achieved by a set of thresholded residuals together with a fault isolation algorithm (Gertler, 1998; Blanke *et al.*, 2006). Fault isolability is then achieved by designing precompiled residual generators where different residuals are sensitive to different subsets of faults.

Examples of residuals derived from model (1) are

$$r_1 = y_1 - y_2 = f_3 - f_4 \tag{2}$$

$$r_2 = \dot{y}_3 + 3y_3 - y_2 = \dot{f}_5 + 3f_5 - f_4 \tag{3}$$

where both the computational form and the internal form, describing the fault influence, are given. Residual r_1 is influenced by the faults $\{f_3, f_4\}$ and r_2 by the faults $\{f_4, f_5\}$. Thus, these two residuals show that the faults

$$\{f_3, f_4\} \cup \{f_4, f_5\} = \{f_3, f_4, f_5\}$$

are detectable. Isolation can also be achieved with these residuals, for example if both these residuals deviates from 0, the diagnoses are the hitting sets of the conflicts

$$\{\{f_3, f_4\}, \{f_4, f_5\}\}$$

i.e. the minimal diagnoses are the single fault $\{f_4\}$ and the double fault $\{f_3, f_5\}$ (de Kleer and Williams, 1987; Reiter, 1987).

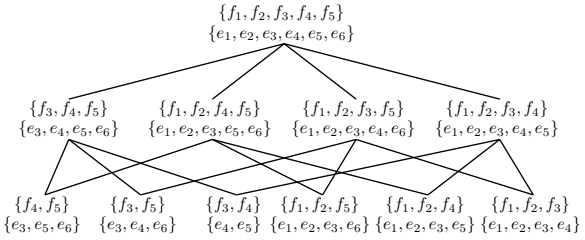

Figure 1: All TSs and TESs for model (1).

This example shows that the fault sets describing the fault influences on the residuals determine both the detectability and isolability capability of the residuals. These sets of faults will be called *test supports* (TSs) which later in Section 3 will formally be defined. For a linear dynamic system like (1), all TSs can be computed using simple rank-conditions on model matrices without the need to design any residuals (Krysander, 2006).

The TSs of (1) are given in Figure 1. There are in total 11 TSs and these sets represent all possible fault sensitivities that can be achieved with any residual derived from the model. For example, the TS of (2) is equal to the third set in the third line. The TSs are organized according to the subset relation. This means that the minimal sets, i.e. the *minimal TSs*, are found in the bottom. The equation sets in the figure indicate which part of the model to use in order to derive a residual with the corresponding TS. For example, $\{f_3, f_4\}$ corresponds to $\{e_4, e_5\}$ which is the set of equations needed for deriving residual (2). Such sets of equations will be referred to as TESs. Since there is a one-to-one correspondence between TESs and TSs we will only focus on TSs in this section.

2.2 TSs for Multiple Fault Diagnosability

In the previous section, TS was introduced and it was mentioned that the set of all TSs can be computed in the linear case. Now, we will show that the set of all TSs characterize detectability and multiple fault isolability of a model.

The definition of isolability used here is a straightforward generalization of the definition given in (Frisk *et al.*, 2009) to the multiple fault case. To define isolability, let the set of observations consistent with the model in mode F_i be denoted by $O(F_i)$. This set is called observation set in (Frisk *et al.*, 2009) and the signature of F_i in (Pucel *et al.*, 2009). Then, a mode F_i is isolable from a mode F_j if

$$O(F_i) \not\subseteq O(F_j) \tag{4}$$

i.e. there exists an observation consistent with F_i but not with F_j. A mode F_i is detectable if the fault is isolable from the fault free mode $F_j = \emptyset$, i.e.

$$O(F_i) \not\subseteq O(\emptyset)$$

It is straightforward to compute, in an exhaustive way, the detectability and isolability for a

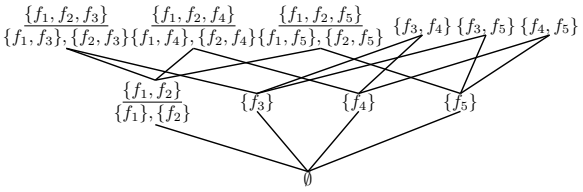

Figure 2: Multiple fault isolability properties of the model (1) represented by a lattice on fault modes.

linear dynamic model by evaluating condition (4) for each pair of modes. Then, for each pair of modes, condition (4) is equivalent to a simple rank-condition (Krysander and Frisk, 2008) in the model matrices. Such an approach is simple in principle but computationally often not feasible since the number of pairs grows exponentially with the number of faults.

The complete detectability and multiple fault isolability of (1) is represented in Figure 2 as a partial order on sets of modes (Pucel *et al.*, 2009). In each node there is one or several fault sets representing system modes. For any two modes F_i and F_j in the same node it holds that $O(F_i) = O(F_j)$, i.e., these faults are not isolable from each other. If F_i is isolable from F_j, i.e. $O(F_i) \not\subseteq O(F_j)$, then $F_i \not\leq F_j$. Finally if both F_i is isolable from F_j and F_j is isolable from F_i, then F_i and F_j are not related. The ideal detectability and isolability is represented by a complete subset lattice of the fault modes.

To give some examples of how Figure 2 should be interpreted, note first that all faults in (1) are detectable since no fault mode is equal to the fault free mode $\emptyset$. Furthermore, the second level shows that all single faults are isolable from each other except that $\{f_1\}$ is not isolable from $\{f_2\}$ and vice versa.

When there are more than one mode in a node the maximal set is underlined. The maximal set in each node characterizes all sets in the node as follows. A mode F_i belongs to the node with maximal set F_j if and only if $F_i \subseteq F_j$ and there exists no $F_k < F_j$ such that $F_i \subseteq F_k$, i.e., F_i is included in the least node where F_i is a subset of the maximal set. Therefore, we will refer to a node by its maximal element. To give an example, $\{f_1\}$ is a subset of $\{f_1, f_2\}$ but not of $\emptyset$. Hence $\{f_1\}$ belongs to the node $\{f_1, f_2\}$.

From the characterization of nodes with their maximal elements, it follows that the multiple fault detectability and isolability properties of the model is fully represented by the lattice and the corresponding maximal elements for each node. These maximal elements are given by the TSs and this can be realized by comparing Figure 1 and Figure 2. First, if the lattice in Figure 2 is flipped horizontally the structure of the graphs becomes equal. Then, by comparing the maximal sets in each node in Figure 2 with the corresponding fault sets in Figure 1, it is clear that these are set complements. Hence, by computing the TSs, the full multiple fault isolability is also obtained.

2.3 TSs for Test Selection

In Section 2.1, TS was introduced and it was discussed how the set of all TSs, here denoted by $\mathcal{T}$, could be used for test design. Now, we will exemplify that this set of TSs is fundamental also when selecting which tests to design, i.e. the test selection problem.

Diagnosis system design can be performed with a two step approach, where in the first step a subset of TSs $T \subseteq \mathcal{T}$ is selected. For each $t_i \in T$, a residual generator is derived in the second step. If there are difficulties in deriving a residual generator for some TS or the resulting diagnosis performance for the residual is unsatisfactory, it is often possible to make another TS selection.

To describe and formalize TS selection, we need first to define detectability and isolability of a set of TSs and then also formalize an isolability specification.

Detectability and isolability of a set T of TSs will be defined as the combined isolability of the individual TSs. A mode F_i is isolable from a mode F_j with test support t_i if a corresponding residual is sensitive to F_i but decouples F_j, i.e., $t_i \cap F_i \neq \emptyset$ and $t_i \cap F_j = \emptyset$. If a mode F_i is isolable from the fault free mode $\emptyset$ with t_i, we say that F_i is detectable with t_i. The fault F_i is isolable from F_j with TSs T, if there exists a TS $t_i \in T$ with that property and the same holds for detectability.

A detectability and isolability specification can be formulated as $\mathcal{I} = \{I_k | 1 \leq k \leq n\}$, where each isolability property I_k specifies that some F_i should be isolable from F_j, $i \neq j$. Given a set of TSs $\mathcal{T}$, let the subset of test supports that provide the isolability property I_k be denoted by $\mathcal{T}_k \subseteq \mathcal{T}$.

The task of selecting tests can then be formulated as, given the set of all TSs $\mathcal{T}$ and an isolability specification $\mathcal{I} = \{I_k | 1 \leq k \leq n\}$, select a subset of test supports $T \subseteq \mathcal{T}$ such that T is a minimal hitting set of the sets $\mathcal{T}_k$ for all $k \in \{1, 2, \ldots, n\}$. If the specified isolability cannot be achieved with $\mathcal{T}$, the best possible isolability is obtained by the minimal hitting sets of all the non-empty sets $\mathcal{T}_k$. An example of test selection will be given in Section 5.1.

To conclude this discussion, the TSs of a model can play an important role for selecting tests to achieve a desired isolability specification.

2.4 Algorithm Objective

To sum up the discussions, TSs are of fundamental importance both for test selection and isolability analysis. All TSs can in a brute force way be computed for linear systems, but since we aim also for analyzing non-linear models a structural approach is chosen. The problem studied in the rest of the paper is how to efficiently compute all TSs and the corresponding TESs using a structural description of the model and the fault locations.

3 THEORETICAL FOUNDATIONS

This section first recapitulates some basic theory in Section 3.1 and then introduces new definitions and basic theoretical concepts used in the proposed approach in Section 3.2.

3.1 Background Theory

As noted in Section 2, general non-linear model descriptions will be analyzed based on the *model structure*. The model structure is a coarse model description which only describes, for each model equation, which variables that are included. For example, the model (1) corresponds to the model structure

	x_1	x_2	x_3	
(1)	x			$\leftarrow f_1$
(2)	x	x	x	$\leftarrow f_2$
(3)		x	x	
(4)		x		$\leftarrow f_3$
(5)		x		$\leftarrow f_4$
(6)			x	$\leftarrow f_5$

For a detailed introduction into the use of model structure for diagnosis, see e.g. (Blanke *et al.*, 2006; Krysander and Frisk, 2008; Krysander *et al.*, 2008).

First, it is assumed that faults are modeled using signals, but as discussed in Section 2, it is equally possible to introduce a component oriented view. Without loss of generality, it is assumed that each fault enters in only one equation. The equation a fault f affects is denoted by e_f. If, in the original model, a fault signal f appears in more than one equation, introduction of an auxiliary equation $x_f = f$ and a simple substitution makes the system fulfill the assumption.

A key tool for analyzing structural models is the Dulmage-Mendelsohn decomposition (Dulmage and Mendelsohn, 1958). By a clever reordering of variables and equations, a unique block diagonal structure can be obtained as is shown in Figure 3.

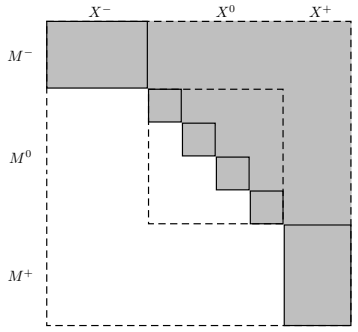

Figure 3: A Dulmage-Mendelsohn decomposition of a structural model.

The main property of the decomposition is that it separates the model into three main parts, the *overdetermined* part M^+ with more equations than variables, the *exactly* determined part M^0, and the *underdetermined* part M^-. The overdetermined part is interesting with respect to diagnosis since that part includes *redundancy* and therefore can be monitored, i.e. tests can be designed with the set of equations in M^+. Hence if a fault is to be detected by a diagnosis system, then there must exist a residual sensitive to that fault. Formally, from (Blanke *et al.*, 2006; Krysander and Frisk, 2008):

Definition 1. *A fault f is structurally detectable in a model M if*

$$e_f \in M^+$$

Note that a structural analysis only gives best case results when applied to a non-linear system. For example, a structurally detectable fault does not have to be detectable in practise, since it might not be possible to use the set M^+ in Definition 1 to compute a residual.

If a diagnosis system should have the capability to distinguish a single fault f_i from a single fault f_j then one of the tests must be sensitive to f_i but not to f_j. In general, if a fault mode is represented by a set F_i of faults, then F_i is isolable from F_j if there exists a residual sensitive to some fault $f \in F_i$ but insensitive to all faults in F_j. Following the ideas in (Krysander and Frisk, 2008), isolability is defined as

Definition 2. *A mode F_i is structurally isolable from mode F_j in a model M if*

$$\exists f \in F_i : e_f \in (M \setminus eq(F_j))^+$$

where $eq(F_j) = \cup_{f \in F_j} e_f$.

It is clear from the above definition that removal of an equation from the model, e.g. decoupling of a fault, and determining the overdetermined part is a crucial operation for determining isolability. In particular it is interesting to observe, when decoupling one fault, which other faults that are automatically decoupled since this implies that these faults are not isolable from each other. As in (Krysander *et al.*, 2008), define a relation on the set equations such that e_1 is related to e_2 if $e_1 \notin (M \setminus \{e_2\})^+$. It can be proven that this is an equivalence relation and the set of equations equivalent to e is denoted by $[e]$.

3.2 Basic Definitions

Now, the theoretical foundations for the proposed approach can be established. A key concept from Section 2 was test support, i.e. a set of faults for which there exists a residual with the corresponding fault sensitivity. To proceed, it is suitable to introduce, as in (Krysander *et al.*, 2008), sets of equations that all can be monitored.

Definition 3 (PSO and MSO)**.** *A set of equations M is proper structurally overdetermined (PSO) if $M = M^+$ and minimally structurally overdetermined (MSO) if no proper subset of M is overdetermined.*

Now, let $F(M)$ denote the set of faults that influence any of the equations in M. Then, since a PSO-set exactly characterizes a set of equations that can be used to form a test, a formal definition is then given by:

Definition 4 (Test Support)**.** *Given a model $\mathcal{M}$ and a set of faults $\mathcal{F}$, a subset of faults $\zeta \subseteq \mathcal{F}$ is a test support if there exists a PSO set $M \subseteq \mathcal{M}$ such that $F(M) = \zeta$.*

Of special interest are *minimal* test supports which is naturally defined as:

Definition 5 (Minimal Test Support)**.** *Given a model, a test support is a minimal test support (MTS) if no proper subset is a test support.*

The above two definitions state which *set of faults* that can affect a test. However, it is also of importance to characterize sets of equations that can be used to form a test. In particular for a given test support ζ we are interested in the maximal set of equations $M = M^+$ such that $F(M) = \zeta$. Without explicitly referring to the corresponding TS this concept is summarized in the following definition:

Definition 6 (Test Equation Support)**.** *An equation set M is a Test Equation Support (TES) if*

1. *$F(M) \neq \emptyset$,*
2. *M is a PSO set, and*
3. *for any $M' \supseteq M$ where M' is a PSO set it holds that $F(M') \supseteq F(M)$.*

Also here it is interesting to consider *minimal* such sets of equations.

Definition 7 (Minimal Test Equation Support)**.** *A TES M is a minimal TES (MTES) if there exists no subset of M that is a TES.*

Here it is clear that there is a one to one correspondence between a TES M and TS ζ given by the relation

$$\zeta = F(M)$$

To give examples of Definition 4-7, recall that all TSs, MTSs, TESs, and MTESs for (1) are given in Figure 1.

4 ALGORITHM

This section describes an algorithm for finding all TESs and MTESs in a model. The algorithm is based on the algorithm developed in (Krysander *et al.*, 2008) for finding all MSO sets. It is a recursive algorithm where in each step one equation is removed from the model and then the overdetermined part of the remaining part is computed. This means for the example in Figure 1 that the nodes, i.e. the TESs, are traversed through spanning-tree of the graph with a depth-first search. The first Lemma shows that as long as we only remove an equation affected by a fault, we remain in the class of sets that are TESs. Due to space constraints, some of the proofs are omitted.

Lemma 1. *Assume that M is a TES and that $f \in F(M)$. If $F((M\backslash\{e_f\})^+) \neq \emptyset$ then $(M\backslash\{e_f\})^+$ is a TES.*

The next Lemma gives a necessary and sufficient criteria to determine if a set is an MTES. This criteria will be used as the stop criteria in the algorithm.

Lemma 2. *Let M be a TES. Then, M is an MTES if and only if there exists an $e \in M$ such that $e_f \in [e]$ for all faults $f \in M$.*

Now the algorithm for finding all MTESs is presented. As pointed out in the beginning of this section, the basic idea in the algorithm is to remove one equation from the model and then compute the overdetermined part of the remaining part of the model. By doing this recursively all MTESs will be found. However as was shown in (Krysander *et al.*, 2008), the equivalence classes, introduced in Section 3.1, can be lumped together in order to reduce the computational complexity of the algorithm. This improvement of the algorithm is used in the algorithm presented here. The steps in the algorithm will be described in the proof of Theorem 1 below.

```
function 𝒮 = MTES(M)
  𝒮 = ∅;
  ℳ = {{e}|e ∈ M⁺};
  if F(ℳ) ≠ ∅
    𝒮 = FindMTES(ℳ, ℳ);
  end
```

The main procedure above calls the recursive procedure `MTES` described below.

```
function 𝒮 = FindMTES(ℳ,ℛ)
  Select an E ∈ ℳ such that F(E) ≠ ∅;
  if e_f ∈ ∪_{E'∈[E]} E' for all f ∈ F(ℳ)
    % ℳ is an MTES
    𝒮 = {∪_{E∈ℳ} E};
  else
    ℛ' = ∅; ℳ' = ℳ;
    % Lump the structure ℳ' and create ℛ'
    while F(ℛ) ≠ ∅
      Select an E ∈ ℛ such that F(E) ≠ ∅;
      ℳ' = (ℳ' \ [E]) ∪ {∪_{E'∈[E]} E'};
      if [E] ⊆ ℛ
        ℛ' = ℛ' ∪ {∪_{E'∈[E]} E'};
      end
      ℛ = ℛ \ [E];
    end
    𝒮 = ∅;
    % Make the recursive calls
    while ℛ' ≠ ∅ do
      Select an E ∈ ℛ';
      ℛ' = ℛ' \ {E};
      𝒮 = 𝒮 ∪ FindMTES(ℳ' \ {E}, ℛ' ∪ ℛ);
    end
end
```

Note that by replacing the first 6 lines in the function `FindMTES` with:

```
function 𝒮 = FindTES(ℳ,ℛ)
  𝒮 = {∪_{E∈ℳ} E};
  % Check if ℳ is not an MTES
  Select an E ∈ ℳ such that F(E) ≠ ∅;
  if exists an f ∈ F(ℳ) : e_f ∉ [E]
```

we obtain an algorithm that outputs all TESs instead.

Theorem 1. *If the function `MTES` is applied to an equation set M, then each MTES is found once and only once.*

Proof. The set $\mathcal{M}$ is a family of equation sets. Initially the sets in $\mathcal{M}$ consist of single equations taken from the overdetermined part M^+ of the model M; see line 3 in MTES.

The argument $\mathcal{R}$ in FindMTES is the subset of sets in $\mathcal{M}$ that are allowed to be removed from $\mathcal{M}$. The reason for introducing this set is to prevent that the same set is traversed more than once in the search tree and in this way avoid that the same set is found more than once by the algorithm. The basic idea is that if a recursive call is made where a set E in $\mathcal{R}$ is removed from $\mathcal{M}$ then E is removed from $\mathcal{R}$ in all other branches. Hence, the search tree is split into two parts, one branch where all sets do not contain the set E and the other branches where all sets do contain E since it is not allowed to be removed from $\mathcal{M}$. It follows that the two parts have no sets in common and by following this rule in all recursive calls it is clear from this discussion that no set will be traversed more than once.

The first step in FindMTES is to check if the stop criteria is fulfilled. The stop criteria is that all e_f, $f \in F(\mathcal{M})$, belong to the same equivalence class. It will be shown below that all sets in the search tree are TES, and it will then follow from Lemma 2 that all sets that fulfill the stop criteria are MTESs.

For all $E \in \mathcal{R}$ such that $F(E) \neq \emptyset$, $\mathcal{M}'$ is formed by first removing all sets in $[E]$ and adding a single set containing all equations in the sets $[E]$. This is where the equivalence classes are lumped together, as described in the paragraph before the algorithm. The set $\mathcal{R}'$ consists of the equivalence classes lumped in the previous step with the additional property that all equations are allowed to be removed, i.e., $[E] \subseteq \mathcal{R}$.

After $\mathcal{M}'$ and $\mathcal{R}'$ have been created, the recursive calls are made in the way outlined above. If $\mathcal{M}$ does not fulfill the stop criteria, then it follows that $F(\mathcal{M}' \setminus \{E\}) \neq \emptyset$, for all equivalence classes $E \in \mathcal{R}'$. Furthermore, the root node M^+ is a TES and all equivalence classes in $E \in \mathcal{R}'$ fulfills the condition $F(E) \neq \emptyset$. Hence, the conditions in Lemma 1 are fulfilled, and it follows that all sets $\mathcal{M}$ in the search tree are TESs. As pointed out above, this implies that all sets that fulfill the stop criteria are MTESs.

To summarize the discussion above, it has been shown that all sets that fulfill the stop criteria are MTESs, and now it remains to show that all MTES are found by the algorithm. In fact we shall prove the stronger statement that all TESs are traversed by the algorithm.

Consider an arbitrary TES M^*. It will now be described which branch in the search to follow to reach the set M^*. The set M^* can be written as the union of a subset of the equivalence classes of $\mathcal{M}$. Hence, for each $E \in \mathcal{M}$, either $F(E) \subseteq F(M^*)$ or $F(E) \subseteq F(\mathcal{M}) \setminus F(M^*)$. To reach the set M^*, select, in each node in the recursive tree, the first recursive call on line 22 with the property that $F(E) \subseteq F(\mathcal{M}) \setminus F(M^*)$. It is clear that if we follow this path, then $F(M^*) \subseteq F(\mathcal{M})$ and $F(\mathcal{M}) \setminus F(M^*) \subseteq F(\mathcal{R})$ hold and the set M^* is reached when $F(\mathcal{M}) = F(M^*)$. □

The algorithm developed in (Krysander *et al.*, 2008) was designed to find all MSO sets. The MSO sets can be characterized by redundancy which is defined as follows. The degree of redundancy $\varphi(M)$ for a model M is defined as the number of equations in M^+ minus the number of variables in M^+, i.e. with some abuse of notation

$$\varphi(M) = |eq(M^+)| - |var(M^+)|$$

The MSO sets are the smallest subsets with redundancy equal to one, i.e. all sets M such that $\varphi(M) = 1$ and no proper subset fulfill the same condition. This can be used as stop criteria in the MSO algorithm.

The next Lemma gives a similar characterization of an MTES, which can also be used as stop criteria in the algorithm. A consequence of this Lemma is that all MTESs are also MSO sets if $(M \setminus M_f)^+ = \emptyset$ where M_f is the set of equations in M that are affected by faults.

Lemma 3. *Given a model M_0, a TES $M \subseteq M_0$ is an MTES if and only if $\varphi(M) = \varphi((M_0 \setminus M_f)^+) + 1$, where M_f is the set of equations in M_0 that are affected by faults.*

The main advantage with the condition in Lemma 3, compared to the condition in Lemma 2, is that it can be checked without computing the Dulmage-Mendelsohn decomposition. It is sufficient to compute the number of equations and unknowns in the model to verify it.

5 EXAMPLES

In this section, the MTES-algorithm will be demonstrated by first applying it to a small three-tank system to illustrate basic properties, and then to a larger truck engine model. Advantages with the proposed algorithm will be shown in a comparison with the results obtained when applying the MSO-algorithm given in (Krysander *et al.*, 2008).

5.1 Three-tank Example

Consider the simple three-tank system in Figure 4. A first principles model of the system is given by the equations

$$e_1 : q_p = \frac{1}{R_{pipe}}(p_s - p_0) \qquad e_2 : q_p = q_0$$

$$e_3 : p_f = p_0 - p_1 \qquad e_4 : q_0 = \frac{1}{R_{v0}} p_f$$

$$e_5 : q_1 = \frac{1}{R_{v1}}(p_1 - p_2) \qquad e_6 : q_2 = \frac{1}{R_{v2}}(p_2 - p_3)$$

$$e_7 : q_3 = \frac{1}{R_{v3}}(p_3 - 0) \qquad e_8 : \dot{p}_1 = \frac{1}{C_1}(q_0 - q_1)$$

$$e_9 : \dot{p}_2 = \frac{1}{C_2}(q_1 - q_2) \qquad e_{10} : \dot{p}_3 = \frac{1}{C_3}(q_2 - q_3)$$

where p_s is the pressure in the (large) fluid source, q_i the flow through valve v_i, q_p the flow through the pipe, and p_i the pressure in the tanks.

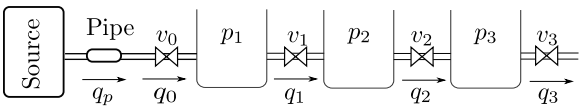

Figure 4: A simple three-tank system.

The measurement equations and a controller for valve v_0 are given by the equations

$$e_{11} : y_1 = q_0 \quad e_{12} : y_2 = p_1 \qquad e_{13} : y_3 = q_3$$
$$e_{14} : y_4 = p_3 \quad e_{15} : R_{v0} = f(y_2)$$

The set of known signals are the measurements y_i and the restriction R_{v0} which is the controlled variable according to equation e_{15}. Seven faults are considered: changes in tank capacities C_1, C_2, C_3, partial blocks in the valves v_1, v_2, v_3 and in the pipe, modeled as changes in the restriction resistances R_{v1}, R_{v2}, R_{v3}, and R_{pipe}.

For this small subsystem there exists 54 MSO sets, i.e. even by only considering the minimal submodels with redundancy there exists 54 different tests that can be designed. For the set of 7 faults, significantly less number of tests are needed to obtain full isolability. To illustrate, consider single fault isolability and use the hitting-set based test selection strategy presented in Section 2.3. As it turns out, even for this small example this is not a feasible approach since there are more than 10^6 minimal set of tests that achieves full isolability. A minimal solution involves 7 tests which is significantly smaller than the complete set of 54 tests. The reason for the high number of minimal solutions is that many of the 54 MSO sets have same, or almost the same, fault signatures.

Algorithm `MTES` from Section 4 delivers 7 MTESs with fault sensitivities according to

$$\{\{f_{v3}\}, \{f_{v2}, f_{c1}, f_{c2}, f_{c3}\}, \{f_{v1}, f_{c1}, f_{c2}, f_{c3}\}, \{f_{v1}, f_{v2}, f_{c2}, f_{c3}\}, \{f_{v1}, f_{v2}, f_{c1}, f_{c3}\}, \{f_{v1}, f_{v2}, f_{c1}, f_{c2}\}, \{f_{pipe}\}\}$$

The hitting-set based approach show that all 7 fault signatures are needed to achieve full single-fault isolability and thus that 7 tests are sufficient. This example shows that, even for a small toy example, there are significant advantages in analyzing the set of MTESs rather than the set of MSO sets. These advantages become even more significant when more realistic, larger, examples are considered.

5.2 Engine Example

The next example analyzes the gas-flow in a truck engine. The gas-flow is modeled in Simulink and the model equations have been automatically generated from the Simulink file. The structure of the resulting equations can be seen in Figure 5. This model has 532 equations, 528 unknowns, and 8 states. The model includes many equations and this is typical for models generated automatically from Simulink or component based modeling approaches. The number of equations can be reduced by analytical computations, but this might reduce the structural information contained in the model and is therefore not recommended.

There are 3 actuators controlling the variable geometry turbine (VGT), the exhaust gas recirculation (EGR), and the fuel injectors (FI). The air-mass flow through the compressor (W), the pressure in the intake manifold (PIM), the pressure in the exhaust manifold (PEM), and the turbine speed (NTRB) are measured and the degree of structural redundancy is equal to 4. Faults in the 3 actuators and the 4 sensors are considered. There

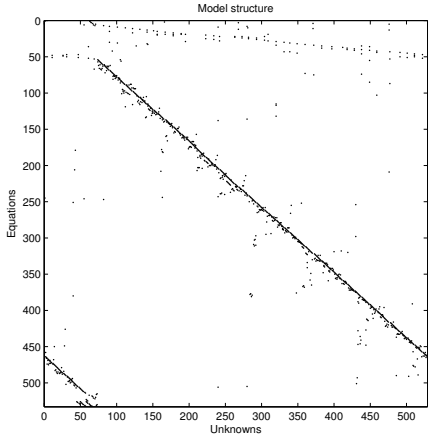

Figure 5: The structure of the gas-flow model where rows correspond to equations and columns to unknowns.

are 1436 MSO sets in this model but only 32 MTESs which all are MSO sets. This result shows two things.

First, the computational complexity of computing the MTESs is much lower than computing the MSO sets. The MSO-algorithm traverses 1774 nodes while the MTES algorithm only visits 61 nodes. The number of traversed nodes is equal to the number of performed Dulmage-Mendelsohn decompositions which is the most computationally demanding operation. The reduction is caused by considering only 7 faults. Generally, if n_f denotes the number of faults, φ the degree of structural redundancy, an upper bound for the number of nodes traversed in the MTES algorithm is given by

$$\sum_{k=0}^{\varphi-1} \binom{n_f}{k}$$

which for this example is equal to 64. For bigger models and especially models with higher degree of structural redundancy, the number of MSO sets can be intractable but with a limited number of faults all MTESs can be computed.

Second, in this example all MSO sets can be computed in about 1s so the computational complexity is not an issue here. However, in the three-tank example it was shown that the selection of MSO sets for a much smaller example is computationally demanding. This selection step can be significantly

simplified or even omitted, since it is sufficient to consider only the 32 MSO sets computed by the MTES algorithm instead of the complete set of 1436 MSO sets.

Concerning the complete multiple fault isolability analysis, this model has full detection, single, double, and triple fault structural isolability except for the following properties. Consider the set of components $C = \{VGT, EGR, FI, PEM\}$ and assume that multiple fault modes are represented by the set of faulty components. Then for each $c \in C$, the single fault $\{c\}$ is not structurally isolable from the triple fault $C \setminus \{c\}$. The only faults isolable from quadruple faults are defined as follows. For each $c \notin C$, the single fault $\{c\}$ is structurally isolable from the quadruple fault C. No fault modes are isolable from fault modes with cardinality strictly greater than 4.

6 CONCLUSIONS

An efficient graph-theoretical algorithm for computing all TESs or MTESs given structural information about the unknown signals and faults has been described. It is based on the MSO-algorithm described in (Krysander *et al.*, 2008), but by searching for MTESs instead of MSO sets the search tree can be pruned, reducing both the computational complexity and the number of resulting equation sets. Two main application of structural methods within the field of diagnosis have been finding and selecting testable models and to perform isolability analysis. We have shown that the less numerous TESs are sufficient for providing the answers to both these questions. The algorithm has been applied to two realistic examples and it has been shown that the number of MTESs are much smaller than the number of MSO sets for these examples, thus reducing the computational burden of finding the MTESs compared to finding the MSO sets. Furthermore, the test selection problem is significantly simplified due to the reduced number of possible tests. Finally, from the direct relation between the set of all TSs and multiple fault isolability, the proposed algorithm has also shown to be a powerful tool computing the complete multiple fault isolability of a system.

REFERENCES

(Armengol *et al.*, 2009) J. Armengol, A. Bregon, T. Escobet, E. Gelso, M. Krysander, M. Nyberg, X. Olive, B. Pulido, and L. Travé-Massuyès. Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09*, pages 1480–1485, Barcelona, Spain, 2009.

(Blanke *et al.*, 2006) M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, second edition, 2006.

(de Kleer and Williams, 1987) J. de Kleer and B.C. Williams. Diagnosing multiple faults. *Artificial Intelligence*, 32(1):97–130, 1987.

(Dulmage and Mendelsohn, 1958) A. L. Dulmage and N. S. Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

(Frisk *et al.*, 2009) E. Frisk, M. Krysander, and J. Åslund. Sensor Placement for Fault Isolation in Linear Differential-Algebraic Systems. *Automatica*, 45(2):364–371, 2009.

(Gelso *et al.*, 2008) E.R. Gelso, S.M. Castillo, and J. Armengol. An algorithm based on structural analysis for model-based fault diagnosis. In *Artificial Intelligence Research and Development. Frontiers in Artificial Intelligence and Applications*, volume 184, pages 138–147. IOS Press, 2008.

(Gertler, 1998) J. Gertler. *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, Inc., 1998.

(Krysander and Frisk, 2008) M. Krysander and E. Frisk. Sensor placement for fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(6):1398–1410, 2008.

(Krysander *et al.*, 2008) Mattias Krysander, Jan Åslund, and Mattias Nyberg. An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(1), 2008.

(Krysander, 2006) Mattias Krysander. *Design and Analysis of Diagnosis Systems Using Structural Methods*. PhD thesis, Linköpings universitet, June 2006.

(Pucel *et al.*, 2009) X. Pucel, W. Mayer, and M. Stumptner. Diagnosability analysis without fault models. In *20th International Workshop on Principles of Diagnosis (DX-09)*, pages 67–74, Stockholm, Sweden, 2009.

(Pulido and Alonso-González, 2004) B. Pulido and C. Alonso-González. Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, 34(5):2192–2206, Octubre 2004.

(Reiter, 1987) R. Reiter. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 32:57–95, 1987.

(Travé-Massuyès *et al.*, 2006) L. Travé-Massuyès, T. Escobet, and X. Olive. Diagnosability analysis based on component-supported analytical redundancy relations. *IEEE Transaction on Systems, Man, and Cybernetics – Part A*, 36(6):1146–1160, 2006.