An Efficient Algorithm for Finding Minimal Overconstrained Subsystems for Model-Based Diagnosis

Mattias Krysander, Jan Åslund, and Mattias Nyberg

***Abstract*—In model-based diagnosis, diagnostic system construction is based on a model of the technical system to be diagnosed. To handle large differential algebraic models and to achieve fault isolation, a common strategy is to pick out small overconstrained parts of the model and to test these separately against measured signals. In this paper, a new algorithm for computing all minimal overconstrained subsystems in a model is proposed. For complexity comparison, previous algorithms are recalled. It is shown that the time complexity under certain conditions is much better for the new algorithm. This is illustrated using a truck engine model.**

***Index Terms*—Fault diagnosis, model-based diagnosis, redundancy, structural analysis, structurally overdetermined.**

I. Introduction

In MODEL-BASED diagnosis, diagnostic system construction is based on a model of the technical system to be diagnosed. In order to achieve fault isolation, a common strategy is to pick out small parts of the model and to test these separately against measured signals.

To cope with large differential algebraic models, systematic structural approaches to find testable subsystems have been suggested in, for example, [1]–[6]. What all of these approaches have in common is that testable subsystems are found among the overconstrained subsystems. Furthermore, of all overconstrained subsystems, it is the minimal ones that are used to derive analytical redundancy relations. Several algorithms for computing all minimal overconstrained subsystems have been proposed in [1], [3]–[5]. However, all of these algorithms run into complexity problems when considering large industrial examples.

In this paper, we present a new algorithm for computing all minimal overconstrained subsystems in a structural model. For the new algorithm, the computational complexity is dependent on the order of structural redundancy, i.e., the difference between the number of equations and unknowns. For a fixed order of structural redundancy, the computational complexity is polynomial in the number of equations in contrast to previous algorithms where the complexity is at least exponential. In many applications, sensors are expensive, and thus, the structural redundancy is low even if the models are large. The algorithm is applied to a Scania truck engine model with 126 equations. There were 1419 minimal overconstrained subsystems, and all of these were found with the new algorithm in less than half a second on a PC with a 1-GHz processor.

To see how this paper is related to previous works, three different types of structural representations, used to describe differential algebraic systems, are recalled in Section II. We introduce a structural characterization, i.e., minimal structurally overdetermined (MSO) set of equations, which characterize overconstrained subsystems independent of structural representation. Several other proposed structural characterizations of overconstrained subsystems are then recapitulated in Section III. We show that all these are MSO sets of equations, and this means that the proposed algorithm can easily be used in any of these structural characterizations and representations to find overconstrained subsystems. In Section IV, a basic algorithm for finding all MSO sets will be presented. This algorithm illustrates the basic ideas, and then, in Section V, further improvements are described. Then, the computational complexity of the proposed algorithm is investigated in Section VI. For comparison, Section VII recalls previous algorithms for finding overconstrained subsystems and analyzes their computational complexity. In Section VIII, it is shown that the computation time for finding all MSO sets in a Scania truck engine model is significantly decreased by using the new algorithm as compared to a previous algorithm. Finally, the complete theory and proofs have been collected in an Appendix.

II. Structural Representations

The structure of a model is represented by a bipartite graph with variables and equations as node sets. There is an edge connecting an equation e and an unknown x if x is included in e. When considering differential algebraic systems, different alternatives for handling derivatives exist. In this section, three different structural representations of a differential algebraic system are recalled. These three variants will be exemplified by the following differential algebraic system:

$$\begin{aligned} \dot{x}_1 &= -x_1^2 + u \\ x_2 &= x_1^2 \\ y &= x_2 \end{aligned} \qquad (1)$$

where u and y are known, and x_1 and x_2 are unknown signals. We will later refer back to these and see how the proposed

Manuscript received October 26, 2005; revised June 18, 2006. This paper was recommended by Associate Editor D. Zhang.

The authors are with the Department of Electrical Engineering, Linköping University, 581 83 Linköping, Sweden (e-mail: matkr@isy.liu.se; jaasl@isy.liu.se; matny@isy.liu.se).

Digital Object Identifier 10.1109/TSMCA.2007.909555

algorithm can be used in any of these representations. The first structural representation of (1) is the following biadjacency matrix of the bipartite graph:

equation	unknown x_1	x_2
e_1	X	
e_2	X	X
e_3		X

(2)

In this representation, all unknowns, i.e., x_1 and x_2, are considered as signals. There is an "X" in position (i, j) in the biadjacency matrix if x_j or any of its time derivatives appear in equation e_i. This approach has been used in, for example, [4] and [7].

The second structural representation of (1) is

equation	unknown x_1	$\dot{x}_1$	x_2
e_1	X	X	
e_2	X		X
e_3			X

Unknowns and their time derivatives are, in contrast to previous representation, considered to be separate independent algebraic variables. New equations can be obtained by differentiation, for example

$$\dot{e}_2 : \dot{x}_2 = 2x_1\dot{x}_1$$
$$\dot{e}_3 : \dot{y} = \dot{x}_2.$$

Now, with these extra equations, the structural representation can be extended as

equation	unknown x_1	$\dot{x}_1$	x_2	$\dot{x}_2$
e_1	X	X		
e_2	X		X	
$\dot{e}_2$	X	X		X
e_3			X	
$\dot{e}_3$				X

(3)

This extended structure is used in [3] and [8].

In the third and final structural representation, unknowns and their time derivatives are, as in the second representation, considered to be separate independent algebraic variables. Thus, the equations are purely algebraic, and differential relations of the form

$$\dot{x}_i = \frac{d}{dt}x_i$$

are added. The structural representation of (1) is

equation	unknown x_1	$\dot{x}_1$	x_2
e_1	X	X	
e_2	X		X
e_3			X
d	X	X	

(4)

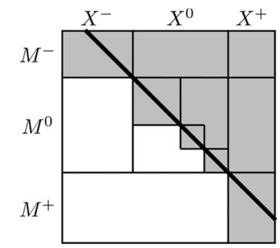

Fig. 1. Dulmage–Mendelsohn decomposition of a model M.

where d is the added differential equation. This representation is used for diagnosis in [1].

III. Use of MSO Sets for Test Construction

From the system of equations represented in (3), the equation

$$\dot{y}^2 - 4y(u - y)^2 = 0$$

can be derived by algebraic elimination of the unknown variables. This is called an analytical redundancy relation or a parity relation and can be used to check if u and y are consistent with the model (1). This means that (1) is a testable system. In this section, a structural characterization of testable subsystems is presented. The main objective in later sections will be to develop an algorithm that, given a large model, finds all subsystems with this structural property.

For a formal structural characterization of testable subsystems, we need first to introduce some important structural properties.

Definition 1 (SO): A set M of equations is structurally overdetermined (SO) if M has more equations than unknowns.

The biadjacency matrix in Fig. 1 shows a Dulmage–Mendelsohn canonical decomposition [9] of a bipartite graph with M and X as node sets. Here, we assume that M is a set of equations and X is a set of unknowns. The gray-shaded areas contain ones and zeros, while the white areas only contain zeros. The thick line represents a maximal matching in the graph defined by this biadjacency matrix. The model M is decomposed into three parts, where the one denoted by M^+ is the structurally overdetermined part with more equations than unknowns. The structurally overdetermined part M^+ of M is the equations $e \in M$ such that, for any maximal matching, there exists an alternating path between at least one free equation and e.

The set $M^0 \cup M^+$ is the maximal set of equations such that there exists a complete matching of the unknown variables in $M^0 \cup M^+$ into the set $M^0 \cup M^+$. The remaining set of equations is denoted as M^-.

Now, note that, in the generic case and for some given trajectories of the known signals, the model M^+ is consistent if and only if $M^0 \cup M^+$ is consistent. Therefore, if an equation in the M^0 part is violated, the consistency of $M^0 \cup M^+$ is not affected. In this sense, it is only the M^+ part that can be a testable subsystem. This motivates us to define the following structural characterization of testable subsystem.

Definition 2 (PSO): An SO set M is a proper structurally overdetermined (PSO) set if $M = M^+$.

A PSO set is generically a testable subsystem, but it may contain smaller PSO subsets that are also testable subsystems. The minimal PSO sets are of special interest since these have the attractive properties of giving good fault isolation. We therefore define the following structural characterization of these minimal sets.

Definition 3 (MSO): An SO set is a minimal structurally overdetermined (MSO) set if no proper subset is an SO set.

Note that an MSO set is also a PSO set. All three structural representations of (1) shown in (2)–(4) are examples of MSO models, and each has more equation than the number of unknowns. Generically, it holds that a minimal testable system has a corresponding MSO model for each of the three structural representations (2)–(4). This is exemplified by (1)–(4).

Since MSO sets represent testable subsystems in all of the three structural representations, comparisons to other structural characterizations of testable subsystems using different representations are possible. In [3] and [7], MSO sets are used to find testable subsystems. In [5], minimal evaluable chains are used, and these are MSO models with the additional requirement that they contain known variables. Other works that use equivalent structural characterizations of testable subsystems are [1], [4], [6], and [10].

IV. New Algorithm

In this section, we will present a new algorithm for finding all MSO sets. This algorithm is based on a top-down approach in the sense that we start with the entire model and then reduce the size of the model step by step until an MSO model remains. To illustrate the ideas, a basic version is presented here, and then, in the next section, improvements are discussed.

Before presenting the algorithm, we need the notion of structural redundancy. Given a bipartite graph, let $\mathrm{var}_X(M) \subseteq X$ be the subset of variables in X connected to at least one equation in M. Given a set of equations M, the structural redundancy $\bar{\varphi}M$ is defined by

$$\bar{\varphi}M = |M^+| - \left|\mathrm{var}_X(M^+)\right|.$$

The algorithm will be based on the following three lemmas.

Lemma 1: If M is a PSO set of equations and $e \in M$, then

$$\bar{\varphi}\left(M\backslash\{e\}\right) = \bar{\varphi}(M) - 1. \tag{5}$$

Lemma 2: The set of equations M is an MSO set if and only if M is a PSO set and $\bar{\varphi}M = 1$.

Lemma 3: If M is a set of equations, $E \subseteq M$ is a PSO set and $e \in M\backslash E$, then

$$E \subseteq \left(M\backslash\{e\}\right)^+.$$

The proofs of all lemmas and theorems can be found in the Appendix. Lemma 1 reveals how the structural redundancy decreases when one equation e is removed from M. It follows from this lemma that, if we start with any PSO set of equations, we can alternately remove equations and computing the structurally overdetermined part until the structural redundancy becomes one. We have then found an MSO set according to Lemma 2. Finally, Lemma 3 implies that an arbitrary MSO set can be obtained recursively in this way. By using this principle in combination with a complete search, the algorithm becomes as follows. The input set M is assumed to be a PSO set.

Algorithm 1: $\mathcal{M}_{MSO} := \texttt{FindMSO}(M)$

```
if φ̄M = 1 then
    M_MSO := {M};
else
    M_MSO := ∅;
    for each equation e in M do
        M' := (M\{e})^+;
        M_MSO := M_MSO ∪ FindMSO(M');
    end for
end if
return M_MSO
```

From the discussion earlier, it follows that the sets found in $\mathcal{M}_{MSO}$ are MSO sets and that all MSO sets are found.

To illustrate the steps in the algorithm, consider the following PSO model consisting of four equations and two unknown variables:

equation	unknown x_1	unknown x_2
e_1	X	
e_2	X	X
e_3		X
e_4		X

(6)

The structural redundancy of this set of equations is two. When entering the algorithm, e_1 is removed, and the Dulmage–Mendelsohn decomposition of the new set $M\backslash\{e_1\} = \{e_2, e_3, e_4\}$ is $(M\backslash\{e_1\})^+ = \{e_3, e_4\}$, $(M\backslash\{e_1\})^0 = \{e_2\}$, and $(M\backslash\{e_1\})^- = \varnothing$. The set M' in the algorithm becomes $(M\backslash\{e_1\})^+ = \{e_3, e_4\}$. In this case, $\bar{\varphi}M' = 1$, and the equation set is saved as an MSO in $\mathcal{M}_{MSO}$. Then, e_2 is removed, and $M' = (M\backslash\{e_2\})^+ = \{e_3, e_4\}$. This means that the same MSO set is found once again. Next, e_3 is removed, and the MSO set $\{e_1, e_2, e_4\}$ is found. Finally, e_4 is removed, and the MSO set $\{e_1, e_2, e_3\}$ is found.

Since the same MSO set $\{e_3, e_4\}$ is found twice, we can suspect that the algorithm is not optimal in terms of efficiency. The next section will therefore present improvements in order to increase the efficiency.

V. Improvements

A straightforward improvement is of course to prohibit that any of the MSO sets are found more than once. Another and more sophisticated improvement is that sets of equations can be lumped together in order to reduce the size and the complexity of the structure. The proposed reduction preserves structural redundancy, and it is therefore possible to use the reduced structure to find all MSO sets in the original structure.

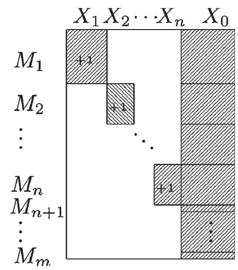

Fig. 2. Structural decomposition of a PSO set.

A. Structural Reduction

The reduction is based on a new unique decomposition of the overdetermined part of a bipartite graph. An illustration of the decomposition is shown in Fig. 2 as a biadjacency matrix. If M is the set of all equations and X is the set of all unknowns, the decomposition can be defined as follows. Let R be a relation on the set M of equations defined by $(e', e) \in R$ if

$$e' \notin (M \backslash \{e\})^+ . \tag{7}$$

Now, we show that R is an equivalence relation. It follows directly from the definition that R is reflexive. If $(e', e) \in R$, then it follows from (7) and Lemma 3, with E replaced by $(M \backslash \{e\})^+$, that $(M \backslash \{e\})^+ \subseteq (M \backslash \{e'\})^+$. Lemmas 1 and 3 imply that both sets have the same structural redundancy and that $(M \backslash \{e\})^+ = (M \backslash \{e'\})^+$. Hence, $(e, e') \in R$, and R is therefore symmetric. Furthermore, if $(e_1, e_2) \in R$ and $(e_2, e_3) \in R$, then it holds that $(M \backslash \{e_1\})^+ = (M \backslash \{e_2\})^+ = (M \backslash \{e_3\})^+$, which implies that R is transitive. The relation R is therefore an equivalence relation.

The set M can then be partitioned into m disjoint equivalence classes M_i. For each equation set M_i, the set X_i is defined as the unknowns included only in M_i and

$$X_0 = X \backslash \left(\bigcup_{i \neq 0} X_i \right).$$

It follows from Lemma 1 (see Corollary 1 in the Appendix) that

$$|M_i| = |X_i| + 1$$

for all $1 \leq i \leq m$, i.e., there is one more equation than unknowns, X_0 excluded, in each block. Furthermore, for $n+1 \leq i \leq m$ in the figure, M_i has cardinality 1 and $X_i = \varnothing$.

By using this partition, all PSO sets can be represented as follows.

Theorem 1: If $E \subseteq M$ is a PSO set, then E is a union of equivalence classes defined by (7), i.e.,

$$E = \bigcup_{i \in I} M_i$$

where $I \subseteq \{1, 2, \ldots, m\}$.

A new bipartite graph can be formed with equivalence classes $\{M_i\}$ and the unknowns X_0 as node sets. The unknowns connected to M_i are $\mathrm{var}_{X_0}(M_i)$. For example, the reduction of (6) is

equivalence class M_i	unknown x_2
$\{e_1, e_2\}$	X
$\{e_3\}$	X
$\{e_4\}$	X

and the decomposition is given by $M_1 = \{e_1, e_2\}$, $M_2 = \{e_3\}$, $M_3 = \{e_4\}$, $X_0 = \{x_2\}$, $X_1 = \{x_1\}$, and $X_2 = X_3 = \varnothing$. Note that it is only equivalence classes of cardinality greater than one that give a reduction. An interpretation of this reduction is that the two first equations are used to eliminate the unknown x_1. In the lumped structure, each equivalence class is considered as one equation, and the definitions of the PSO set, MSO set, and structural redundancy are thereby extended to lumped structures. In the example earlier, we have $\bar{\varphi}\{\{e_1, e_2\}, \{e_3\}, \{e_4\}\} = 2$. The structural redundancy for the lumped and the original structure are always the same.

The reduction is justified by the following theorem, which shows that there is a one-to-one correspondence between the PSO sets in the original and in the lumped structure, and that, the reduced structure can be used to find all PSO sets in the original structure.

Theorem 2: The set $\{M_i\}_{i \in I}$ is a PSO set in the lumped structure if and only if $\cup_{i \in I} M_i$ is a PSO set in the original structure.

B. Improved Algorithm

A drawback with Algorithm 1, presented in Section IV, is that some of the MSO sets are found more than once. There are two reasons why this happens, and these can be illustrated using the following example:

equation	unknown x_1	x_2
e_1	X	
e_2	X	X
e_3		X
e_4		X
e_5		X

(8)

First, the same PSO set $\{e_3, e_4, e_5\}$ is obtained if either e_1 or e_2 is removed. Second, the same MSO set is obtained if the order of equation removal is permuted. For example, the MSO set $\{e_4, e_5\}$ is obtained if, first, e_1 or e_2 and, then, e_3 is removed but also if the order of removal is reversed.

To illustrate how these two problems are handled in the improved algorithm, we use the example (8).

To avoid the first problem, the lumping described in the previous section is used. Initially, we start with the set $M = \{e_1, e_2, e_3, e_4, e_5\}$, and e_1 and e_2 are lumped together, and the resulting set is $\mathcal{M}' = \{\{e_1, e_2\}, \{e_3\}, \{e_4\}, \{e_5\}\}$. Similar to the basic algorithm, we remove one equivalence class at a time from $\mathcal{M}'$ and make a subroutine call which returns all MSO sets in the input set.

To avoid the problem with permuted removal order, an additional input set $\mathcal{R}'$ is used, which contains the equivalence classes that are allowed to be removed in the recursive calls.

In the example, we start initially with the set $\mathcal{R}' = \mathcal{M}'$, meaning that all equivalence classes are allowed to be removed. In the first step, the equivalence class $\{e_1, e_2\}$ is removed, and the subroutine is called with the input sets

$$\mathcal{M}' \backslash \{\{e_1, e_2\}\} \text{ and } \mathcal{R}' = \{\{e_3\}, \{e_4\}, \{e_5\}\}.$$

To prevent that the order of removal is permuted, we remove the equivalence class $\{e_1, e_2\}$ permanently from $\mathcal{R}'$. In the following step, the equivalence class $\{e_3\}$ is removed, and the inputs are

$$\mathcal{M}' \backslash \{\{e_3\}\} \text{ and } \mathcal{R}' = \{\{e_4\}, \{e_5\}\}.$$

Following the same principles, the final calls are made with the input sets

$$\mathcal{M}' \backslash \{\{e_4\}\} \text{ and } \mathcal{R}' = \{\{e_5\}\}$$

$$\mathcal{M}' \backslash \{\{e_5\}\} \text{ and } \mathcal{R}' = \varnothing.$$

To apply these ideas in all steps in the recursive algorithm, the lumping strategy has to be extended to subsets of previously lumped structures. Equivalence classes are then lumped together into new sets of equations by taking the union of the sets in the equivalence class. We illustrate this with a new example. Assume that we start with six equations and that e_2 and e_3 are lumped together and the following structure is obtained:

equation	unknown x_1	unknown x_2
$\{e_1\}$	X	
$\{e_2, e_3\}$	X	
$\{e_4\}$	X	X
$\{e_5\}$		X
$\{e_6\}$		X

(9)

In the first recursive call, $\{e_1\}$ is removed, and the graph corresponding to the remaining part has the same structure as in (6). Now

$$[\{e_2, e_3\}] = [\{e_4\}] = \{\{e_2, e_3\}, \{e_4\}\}$$

where $[E]$ denotes the equivalence class containing E. The sets $\{e_2, e_3\}$ and $\{e_4\}$ are therefore lumped together into the set $\{e_2, e_3, e_4\}$.

Given a model $\mathcal{M}$ and corresponding set $\mathcal{R}$, the lumped structure $\mathcal{M}'$ is constructed as described earlier, and the problem is then on how to form the new set $\mathcal{R}'$ of equivalence classes that are allowed to be removed in the new structure $\mathcal{M}'$. The following principle will be used. An equivalence class in $\mathcal{M}'$ is allowed to be removed, i.e., belongs to $\mathcal{R}'$, if and only if it is a union of classes that are all allowed to be removed in $\mathcal{M}$, i.e., belongs to $\mathcal{R}$. It will be shown that, in this way, all MSO sets are found once and only once.

It is sufficient to only lump equivalence classes with a nonempty intersection with $\mathcal{R}$, and this is used in the algorithm. To do this partial lumping, we will use the notation $\texttt{Lump}([E], \mathcal{M}')$ in the algorithm to denote that only the equivalence class $[E]$ in $\mathcal{M}'$ is lumped and that the other equations remain unchanged. The improved algorithm can now formally be written as follows.

Algorithm 2: $\mathcal{M}_{MSO} = \texttt{MSO}(M)$

$\mathcal{M} := \{\{e\} | e \in M^+\}$;
$\mathcal{M}_{MSO} := \texttt{FindMSO}(\mathcal{M}, \mathcal{M})$;
return $\mathcal{M}_{MSO}$;
Subroutine: $\mathcal{M}_{MSO} := \texttt{FindMSO}(\mathcal{M}, \mathcal{R})$
if $\bar{\varphi}\mathcal{M} = 1$ **then**
  $\mathcal{M}_{MSO} := \{\cup_{E \in \mathcal{M}} E\}$
else
  $\mathcal{R}' := \varnothing; \mathcal{M}' := \mathcal{M}$;
  % `Lump the structure` $\mathcal{M}'$ `and create` $\mathcal{R}'$
  while $\mathcal{R} \neq \varnothing$ **do**
    Select an $E \in \mathcal{R}$;
    $\mathcal{M}' := \texttt{Lump}([E], \mathcal{M}')$;
    if $[E] \subseteq \mathcal{R}$ **then**
      $\mathcal{R}' := \mathcal{R}' \cup \{\cup_{E' \in [E]} E'\}$;
    end if
    $\mathcal{R} := \mathcal{R} \backslash [E]$;
  end while
  $\mathcal{M}_{MSO} := \varnothing$;
  % `Make the recursive calls`
  while $\mathcal{R}' \neq \varnothing$ **do**
    Select an $E \in \mathcal{R}'$;
    $\mathcal{R}' := \mathcal{R}' \backslash \{E\}$;
    $\mathcal{M}_{MSO} := \mathcal{M}_{MSO} \cup \texttt{FindMSO}(\mathcal{M}' \backslash \{E\}, \mathcal{R}')$;
  end while
end if
return $\mathcal{M}_{MSO}$

The algorithm is justified by the following result.

Theorem 3: If Algorithm 2 is applied to a set M, then each MSO set contained in M is found once and only once.

VI. Computational Complexity

The objective of this section is to investigate the computational complexity of Algorithm 2. In general, the number of MSO sets may grow exponentially in the number of equations. This gives a lower bound for the computational complexity in the general case. However, in many applications, the order of structural redundancy is low, and it will be shown that, in this case, better computational complexity can be achieved. The redundancy is often low due to the fact that the structural redundancy depends on the number of available sensors, which are often expensive. One example of this is given in Section VIII. In this section, the computational complexity of the algorithm will be analyzed in the case where the structural redundancy is low.

The worst case is when all unknown variables are included in each equation. Algorithm 2 traverses the PSO sets exactly once in the subset lattice. The following lemma gives an upper bound for the number of PSO sets.

Lemma 4: Given a model with n equations and with structural redundancy $\bar{\varphi}$, there are at most

$$\sum_{k=n-\bar{\varphi}+1}^{n} \binom{n}{k} \tag{10}$$

PSO subsets.

Proof: In the worst case, the PSO sets are all subsets of equations with cardinality strictly greater than the number of unknowns in the original model, i.e., greater than $n - \bar{\varphi}$. The number of subsets with k equations is, in this case

$$\binom{n}{k}$$

which gives the result in the lemma. ■

The next theorem gives the computational complexity in the case of low structural redundancy.

Theorem 4: For a fixed order of structural redundancy $\bar{\varphi}$, Algorithm 2 has order of $n^{\bar{\varphi}+1.5}$ time complexity, where n is the number of equations.

Proof: An upper bound for the number of PSO sets is given by Lemma 4. For $k \geq n - \bar{\varphi} + 1$, the terms in (10) can be estimated as

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \leq \frac{n!}{k!} \leq n^{\bar{\varphi}-1} \tag{11}$$

and the number of terms is fixed. Hence, the sum is less than $\bar{\varphi}\, n^{\bar{\varphi}-1}$. In the worst case, the number of times the structurally overdetermined part has to be calculated is given by the sum of (10). To compute, the overdetermined part has order of $n^{2.5}$ time complexity [11]. Hence, Algorithm 2 has order of $n^{\bar{\varphi}+1.5}$ time complexity. ■

VII. Previous Algorithms

Different algorithms for finding all MSO sets have been presented in previous literature. These will now be recalled, and the complexity of the previous algorithms will be analyzed under the same condition as in the theorem earlier, i.e., for a fixed order of redundancy. The worst case, for all algorithms discussed in this paper, is when all unknown variables are included in each equation.

One approach for finding all MSO sets was presented in [12] and further developed in [5]. Independently, the same algorithm was presented in [3]. The basic principle is to choose one equation as the redundant equation and then find all possible ways to compute structurally all unknowns in the redundant equations. The redundant equation is first chosen to be the first equation and then the second and so on until the last equation is the redundant equation. When all possible ways to compute all unknowns in the first equation are found, all MSO sets, including the first equation, have been found. This means that the first equation will not be used further in the search for more MSO models.

In [4], a method based on elimination rules is presented. The unknowns are eliminated in a specified order. Each unknown is eliminated in all possible ways. For each way, the equations used form an MSO set.

In all the algorithms discussed earlier, a bottom–up approach is used, and all subsets of MSO sets are traversed at least once in the worst case. For this case, the proper subsets of MSO sets are exactly those sets that are not PSO sets. The number of PSO sets grows polynomially in n according to the discussion earlier [see Lemma 4 and the estimate in (11)]. Furthermore, the number of all subsets is 2^n. Hence, for a fixed order of structural redundancy, the number of subsets of MSO sets grows exponentially, and the computational complexity of these algorithms is exponential.

Another approach for finding all MSO sets is presented in [1]. All maximal matchings are first enumerated. Then, for each maximal matching and for each free equation for this matching, an MSO set is given by the equations reached by an alternating path from the free equation (for further details, see [13]). It follows from the discussion earlier that all maximal matchings have to be found. In the worst case, the number of maximal matchings is equal to the number of ordered subsets of equations with size $n - \bar{\varphi}$, i.e., there are $n!/\bar{\varphi}!$ number of maximal matchings. Hence, for a fixed order of structural redundancy, the computational complexity of this algorithm is factorial in the number of equations.

In conclusion, in the case of low structural redundancy, Algorithm 2 has better computational complexity than the others. However, it should be pointed out that, in the case of few unknowns, the roles are reversed. For a fixed number of unknowns, the new algorithm has exponential time complexity, and all previous algorithms has polynomial time complexity. However, this situation is, as pointed out before, not common in industrial applications.

VIII. Application to a Large Industrial Example

To demonstrate the efficiency of Algorithm 2, we will apply it here to a real industrial process. The process is a Scania truck diesel engine, and a sketch is shown in Fig. 3. This engine has two actuators, namely, the fuel injection δ and the exhaust gas recirculation (EGR) valve. It has eight sensors, namely, ambient pressure p_{amb}, ambient temperature T_{amb}, air flow W_{cmp}, inlet pressure p_{im}, inlet temperature T_{im}, exhaust pressure p_{em}, engine speed n_{eng}, and turbine speed n_{trb} (further details of the application are presented in [14]).

A simulation model of the engine was provided in Simulink. This model has four states and four outputs. These four outputs are W_{cmp}, p_{im}, p_{em}, and n_{trb}. The rest of the sensors are in the Simulink model implemented as inputs. To analyze the model, it was transferred to a flat list of equations. The number of equations is 126, and the structural redundancy is four. The fact that the structural redundancy is four is a consequence of that the number of outputs is four.

For comparison, three algorithms were tested on the set of 126 equations. The first is the old MSO algorithm presented in [3], where an alternative partial reduction is used. Without any reduction, the old MSO algorithm is practically intractable for this example. The second is the new basic algorithm presented in Section IV with the structural reduction in Section V-A applied initially, reducing the number of equations to 28. The third is the new improved algorithm presented in Section V.

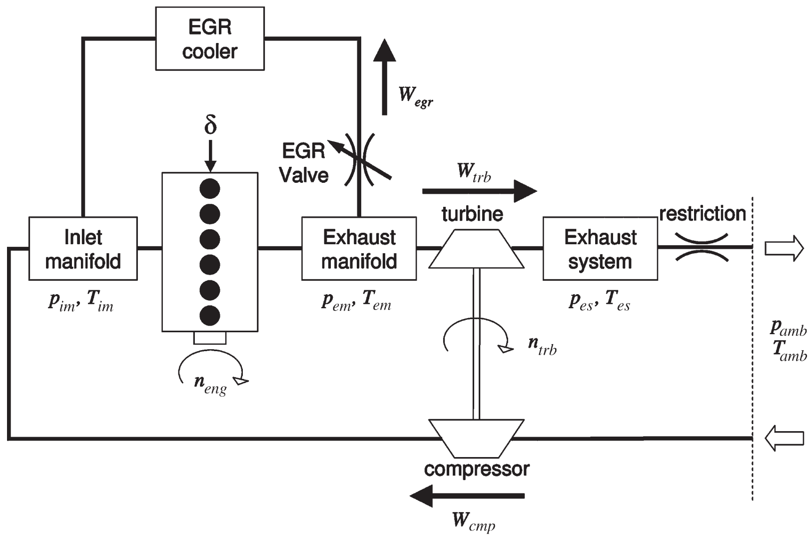

Fig. 3. Example of a Scania truck engine.

TABLE I
COMPARISON OF THREE MSO ALGORITHMS

Algorithm	Execution time
The old MSO algorithm	5900 s
The new basic algorithm	18 s
The new improved algorithm	0.42 s

All algorithms were implemented in MATLAB and executed on a PC with a 1-GHz processor. The execution times were measured in seconds and are presented in Table I.

There were 1419 MSO sets, and in the table, we can see that the new MSO algorithm is more than 14 000 times faster than the old algorithm.

IX. CONCLUSION

A new algorithm for finding all MSO sets of equations was developed. The proposed algorithm can be used for all structural representations presented in Section II. There are three main ideas that are used in the new algorithm. First, it is based on a top-down approach, as described in Section IV. Second, a structural reduction is used where subsets of equations are lumped together in order to reduce the size of the structural model. Third and last, it is prohibited that any MSO set is found more than once. For a fixed order of structural redundancy, the computational complexity of the new algorithm is polynomial in the number of equations, in contrast to previous algorithms where the complexity is at least exponential. The efficiency of the algorithm was demonstrated by applying the new and a previous algorithm to a model of a Scania truck engine.

APPENDIX
THEORY

The following concepts and their theoretical foundation are given in [15]. Let $\varphi : 2^{\mathcal{M}} \to \mathbb{Z}$ be defined by

$$\varphi M = |M| - |\mathrm{var}_{\mathcal{X}}(M)|. \tag{12}$$

This number φM will be called the surplus of M. Note that $\varphi\varnothing = 0$. The surplus function φ is a supermodular function on the family of equation subsets in $\mathcal{M}$ since

$$\varphi(M_1 \cup M_2) + \varphi(M_1 \cap M_2) \geq \varphi M_1 + \varphi M_2 \tag{13}$$

for all $M_1 \subseteq \mathcal{M}$ and $M_2 \subseteq \mathcal{M}$. A set M is said to be a minimal set of surplus φM if

$$\varphi E < \varphi M \tag{14}$$

for all $E \subset M$.

Let M be an arbitrary subset of $\mathcal{M}$. Each subset E of M defines a surplus φE, and we define $\bar{\varphi}$ by

$$\bar{\varphi} M = \max_{E \subseteq M} \varphi E. \tag{15}$$

This number will be called the structural redundancy of M and is equivalent with the definition introduced in Section IV. It holds that $\varphi M \geq 0$, and the surplus of M is clearly less or equal to the structural redundancy of M, i.e.,

$$\bar{\varphi} M \geq \varphi M. \tag{16}$$

Furthermore, the structural redundancy $\bar{\varphi}$ is a supermodular function [15], i.e., $\bar{\varphi}$ satisfies inequality (13). A set M is said to be a minimal set of structural redundancy $\bar{\varphi} M$ if

$$\bar{\varphi} E < \bar{\varphi} M \tag{17}$$

for all $E \subseteq M$.

Let M be any subset in $\mathcal{M}$. Among all subsets E of M with maximal surplus, i.e.,

$$\varphi E = \bar{\varphi} M \tag{18}$$

there exists a unique minimal subset [15]. This set will be denoted as M^+ and will be called the structurally overdetermined part of M. In [15], it is shown that the set M can be partitioned into $M^+ \cup (M \backslash M^+)$ such that

$$\bar{\varphi} M = \bar{\varphi} M^+ \quad (19)$$
$$\bar{\varphi}(M \backslash M^+) = 0. \quad (20)$$

This means that M^+ contains all structural redundancy of M, and this is also discussed in Section III. The Dulmage–Mendelsohn decomposition can be used to compute the structurally overdetermined part in an efficient way [9], [11]. Note that given any set M of equations such that

$$M^+ \neq \varnothing \quad (21)$$

it follows from (18) and Definition 2 that M^+ is a PSO set. This means that properties of M^+ sets carry over to PSO sets.

Lemma 5: The following three statements about a set M are equivalent.

1) The set M is a PSO set.
2) The set M is a minimal set of surplus $\varphi M > 0$.
3) The set M is a minimal set of structural redundancy $\bar{\varphi} M > 0$.

Proof: 1) $\Rightarrow$ 2). Since $M = M^+$ and $M \neq \varnothing$, it follows from (18) that M is a minimal set of surplus $\varphi M > 0$.

2) $\Rightarrow$ 3). Since M is a minimal set of surplus $\varphi M > 0$, i.e., it satisfies (14) for all $E \subset M$. Let M_1 be an arbitrary proper subset of M. It follows that

$$\bar{\varphi} M_1 = \max_{E \subseteq M_1} \varphi E < \max_{\bar{E} \subseteq M} \varphi \bar{E} = \bar{\varphi} M$$

according to (14). Since M_1 is an arbitrary proper subset of M, it follows that M is a minimal set of structural redundancy $\bar{\varphi} M = \varphi M > 0$.

3) $\Rightarrow$ 1). Since M is a minimal set of structural redundancy $\varphi M > 0$, it follows from (19) that $M = M^+$ and $M \neq \varnothing$, i.e., M is a PSO set. ■

Proof of Lemma 1: From the definition of the surplus function φ in (12), it follows that

$$\varphi \left(M \backslash \{e\}\right) \geq \varphi(M) - 1. \quad (22)$$

This, (16), and $\varphi M = \bar{\varphi} M$ give that

$$\bar{\varphi} \left(M \backslash \{e\}\right) \geq \bar{\varphi}(M) - 1. \quad (23)$$

Since M is a PSO set, Lemma 5 states that M is a minimal set of structural redundancy $\bar{\varphi} M$, i.e.,

$$\bar{\varphi}(M) > \bar{\varphi} \left(M \backslash \{e\}\right) \geq \bar{\varphi}(M) - 1 \quad (24)$$

which implies (5). This completes the proof. ■

From this theorem, (18) and (21), it follows that, for any PSO set with structural redundancy $\bar{\varphi}_1 > 1$, there exists a proper subset which is a PSO set with structural redundancy $\bar{\varphi}_1 - 1$.

Corollary 1: If M is a PSO set, then for all its equivalence classes M_i defined by (7), it holds that

$$|M_i| = |X_i| + 1. \quad (25)$$

Proof: Let M_i be an arbitrary equivalence class which, according to the decomposition, implies that for any $e \in M_i$, $(M \backslash \{e\})^+ = M \backslash M_i$. Then, we form

$$\varphi(M) - \varphi \left(M \backslash \{e\}\right)^+ = (|M| - |X|) - (|M \backslash M_i| - |X \backslash X_i|)$$

which can be simplified to

$$\varphi(M) - \varphi \left(M \backslash \{e\}\right)^+ = |M_i| - |X_i|.$$

Since M and $(M \backslash \{e\})^+$ are PSO sets, it follows that

$$\bar{\varphi}(M) - \bar{\varphi} \left(M \backslash \{e\}\right)^+ = |M_i| - |X_i|.$$

Then Lemma 1 and (19) imply (25), and this proves this corollary. ■

Proof of Lemma 2: Assume that M is an MSO set. The set M is therefore an SO set, and it follows that $0 < \varphi M \leq \bar{\varphi} M$. This and (19) imply that $M^+ \neq \varnothing$, i.e., M^+ is a PSO set and, therefore, also an SO set. Since M is an MSO set, it follows that $M = M^+$, i.e., M is a PSO set.

Assume that M has structural redundancy $\bar{\varphi} M > 1$. Then, it follows from Lemma 1 that

$$\bar{\varphi} \left(M \backslash \{e\}\right) = \bar{\varphi}(M) - 1 \geq 1. \quad (26)$$

This implies that $(M \backslash \{e\})^+ \neq \varnothing$ and that $(M \backslash \{e\})^+ \subset M$ is a PSO set which contradicts that M is an MSO set. Hence, $\bar{\varphi} M = 1$.

Assume that M is a PSO set and that $\bar{\varphi} M = 1$. This and (18) imply that $\varphi M = 1$. By using Lemma 5, it follows that all proper subsets $E \subset M$ have $\varphi E = 0$, i.e., E is not an SO set. Hence, M is an MSO set. ■

Lemma 6: Given two PSO sets M_1 and M_2, it follows that $M_1 \cup M_2$ is a PSO set and that

$$\bar{\varphi}(M_1 \cup M_2) \geq \max(\bar{\varphi} M_1, \bar{\varphi} M_2). \quad (27)$$

Equality is obtained if and only if $\bar{\varphi} M_1 \leq \bar{\varphi} M_2$ and $M_1 \subseteq M_2$ or $\bar{\varphi} M_2 \leq \bar{\varphi} M_1$ and $M_2 \subseteq M_1$.

Proof: See [15, Theorem 1.2.1]. ■

Lemma 7: If E and M are two equation sets such that $E \subseteq M$, then $E^+ \subseteq M^+$.

Proof: The fact that $E \subseteq M$ implies that $E^+ \cup M^+ \subseteq M$ and from (19) also that

$$\bar{\varphi}(M^+ \cup E^+) \leq \bar{\varphi} M = \bar{\varphi} M^+. \quad (28)$$

Lemma 6 implies that

$$\bar{\varphi} M^+ \leq \max(\bar{\varphi} M^+, \bar{\varphi} E^+) \leq \bar{\varphi}(M^+ \cup E^+). \quad (29)$$

The inequalities (28) and (29) give that

$$\bar{\varphi}(M^+ \cup E^+) = \bar{\varphi} M^+ \quad (30)$$

and $\bar{\varphi}E^+ \leq \bar{\varphi}M^+$. This and the equality in (30) imply that $E^+ \subseteq M^+$ according to Lemma 6. ■

Proof of Lemma 3: This theorem follows immediately from Lemma 7 by noting that E is a PSO set, i.e., $E = E^+$ and $E \subseteq (M\backslash\{e\})$. ■

Lemma 8: Let M be a PSO set and M_i an arbitrary equivalence class defined by (7). If E is a PSO set such that $E \subseteq M$ and $E \cap M_i \neq \varnothing$, then $M_i \subseteq E$.

Proof: Assume that there exists an $e \in M_i\backslash E \subseteq M\backslash E$. From Lemma 3, it follows that $E \subseteq (M\backslash\{e\})^+$. This and the definition of M_i imply that $E \subseteq M\backslash M_i$, which contradicts the assumption and the lemma follows. ■

Proof of Theorem 1: Follows immediately from Lemma 8. ■

Lemma 9: If M is a PSO set and $\{M_i\}_{i\in I}$ are its equivalence classes, then

$$\varphi(\cup_{i\in I'} M_i) = \varphi\left(\{M_i\}_{i\in I'}\right) \tag{31}$$

for all $I' \subseteq I$.

Proof: By using the notation of the structural decomposition described in Section V-A, the surplus of $\cup_{i\in I'} M_i$ can be expressed as

$$\varphi(\cup_{i\in I'} M_i) = |\cup_{i\in I'} M_i| - |\cup_{i\in I'} X_i| - |\mathrm{var}_{X_0}(\cup_{i\in I'} M_i)| \tag{32}$$

which can be rewritten as

$$\varphi(\cup_{i\in I'} M_i) = \sum_{i\in I'} (|M_i| - |X_i|) - |\mathrm{var}_{X_0}(\cup_{i\in I'} M_i)|. \tag{33}$$

Corollary 1 states that $|M_i| = |X_i| + 1$ for all $i \in I$ and, consequently, that

$$\varphi(\cup_{i\in I'} M_i) = |I'| - |\mathrm{var}_{X_0}(\cup_{i\in I'} M_i)| \tag{34}$$

which is equal to $\varphi(\{M_i\}_{i\in I'})$. ■

Proof of Theorem 2: Assume that $\cup_{i\in J} M_i$ is a PSO set. From Lemma 5, it follows that

$$\varphi(\cup_{i\in J'} M_i) < \varphi(\cup_{i\in J} M_i) \tag{35}$$

for all $J' \subset J$. From Lemma 9, it then follows that

$$\varphi\left(\{M_i\}_{i\in J'}\right) < \varphi\left(\{M_i\}_{i\in J}\right) \tag{36}$$

for all $J' \subset J$. Hence, $\{M_i\}_{i\in J}$ is a minimal set of surplus $\varphi(\{M_i\}_{i\in J})$, i.e., $\{M_i\}_{i\in J}$ is a PSO set according to Lemma 5.

Now, we will show the reverse implication. Assume that $\{M_i\}_{i\in J}$ is a PSO set. If $M' \subset \cup_{i\in J} M_i$, then

$$M' \supseteq (M')^+ = \cup_{i\in J'} M_i \tag{37}$$

for some $J' \subset J$ according to Theorem 1. Since $\{M_i\}_{i\in J}$ is a PSO set, it follows from Lemma 5 and Lemma 9 that

$$\begin{aligned}\varphi(\cup_{i\in J} M_i) &= \varphi\left(\{M_i\}_{i\in J}\right) > \varphi\left(\{M_i\}_{i\in J'}\right)\\ &= \varphi(\cup_{i\in J'} M_i).\end{aligned} \tag{38}$$

From (18) and (37), it follows that

$$\varphi(\cup_{i\in J'} M_i) = \varphi(M')^+ \geq \varphi M'. \tag{39}$$

The inequalities (38) and (39) imply that $\cup_{i\in J} M_i$ is a minimal set of surplus $\varphi(\cup_{i\in J} M_i)$, i.e., $\cup_{i\in J} M_i$ is a PSO set according to Lemma 5. ■

Proof of Theorem 3: First, it is shown that each MSO set is found at least once. Let $E \subseteq M$ be an arbitrary MSO set. A branch, of the recursive tree, that results in this MSO set can be obtained in the following way. In each recursive step, chose the first branch where an equivalence class not included in E is removed. It follows from Lemma 3 and Theorem 2 that, by following this branch, a sequence of decreasing PSO sets all containing E is obtained. Hence, the MSO set E is found this way.

Finally, it is shown that the same MSO set E cannot be found if we deviate from the branch described earlier, i.e., that the MSO set E is found only once. In each recursive step, in all branches that precede this branch, only equivalence classes contained in E have been removed. Therefore, these branches do not result in the set E. On the other hand all succeeding branches contain the first equivalence class $\hat{E}$ not contained in E, i.e., the class removed in the branch that gives the set E. This follows from the fact that $\hat{E}$ has been removed from $\mathcal{R}$ and is not allowed to be removed. Furthermore, in all lumped structures in these branches, $\mathcal{R}'$ is constructed such that $\hat{E}$ is an equivalence class not contained in $\mathcal{R}'$. Hence, the branch described earlier is the only branch that results in the MSO set E. This completes the proof. ■

References

[1] M. Blanke, M. Kinnert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. New York: Springer-Verlag, 2003.

[2] J. P. Cassar and M. Staroswiecki, "A structural approach for the design of failure detection and identification systems," in *Proc. IFAC Control Ind. Syst.*, Belfort, France, 1997.

[3] M. Krysander and M. Nyberg, "Structural analysis utilizing MSS sets with application to a paper plant," in *Proc. 13th Int. Workshop Principles Diagnosis*, Semmering, Austria, May 2002, pp. 51–57.

[4] S. Ploix, M. Desinde, and S. Touaf, "Automatic design of detection tests in complex dynamic systems," in *Proc. 16th IFAC World Congr.*, Prague, Czech Republic, 2005.

[5] B. Pulido and C. Gonzalez, "Possible conflicts: A compilation technique for consistency-based diagnosis," *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 5, pp. 2192–2206, Oct. 2004.

[6] L. Travé-Massuyès, T. Escobet, and R. Milne, "Model-based diagnosability and sensor placement. Application to a frame 6 gas turbine subsystem," in *Proc. 12th Int. Workshop Principles Diagnosis (DX)*, 2001, pp. 205–212.

[7] E. Frisk, D. Düştegör, M. Krysander, and V. Cocquempot, "Improving fault isolability properties by structural analysis of faulty behavior models: Application to the DAMADICS benchmark problem," in *Proc. IFAC Safeprocess*, Washington, DC, 2003.

[8] M. Krysander and J. Åslund, "Graph theoretical methods for finding analytical redundancy relations in overdetermined differential algebraic systems," in *Proc. IMACS World Congr.*, Paris, France, 2005.

[9] A. L. Dulmage and N. S. Mendelsohn, "Coverings of bipartite graphs," *Can. J. Math.*, vol. 10, pp. 517–534, 1958.

[10] T. Lorentzen, M. Blanke, and H. Niemann, "Structural analysis—A case study of the Römer satellite," in *Proc. IFAC Safeprocess*, Washington, DC, 2003.

[11] K. Murota, *Matrices and Matroids for System Analysis*. New York: Springer-Verlag, 2000.

[12] B. Pulido and C. Alonso, "Possible conflicts, ARRs, and conflicts," in *Proc. DX*, M. Stumptner and F. Wotawa, Eds, Semmering, Austria, 2002, pp. 122–127.

[13] D. Düştegör, E. Frisk, V. Coquempot, M. Krysander, and M. Staroswiecki, "Structural analysis of fault isolability in the DAMADICS benchmark," *Control Eng. Pract.*, vol. 14, no. 6, pp. 597–608, 2006.

[14] L. Eriksson, "Structural algorithms for diagnostic system design using Simulink models," M.S. thesis, Linköpings Univ., Linköping, Sweden, 2004.

[15] O. Ore, "Studies on directed graphs I," *Ann. Math.*, vol. 63, no. 3, pp. 383–406, May 1956.

Mattias Krysander was born in Linköping, Sweden, in 1977. He received the M.Sc. and Ph.D. degrees in electrical engineering from Linköping University, Linköping, in 2000 and 2006, respectively.

He is currently an Assistant Professor with the Department of Electrical Engineering, Linköping University. His current research interests include model-based fault diagnosis using graph-theoretical and structural methods.

Jan Åslund was born in Boden, Sweden, in 1971. He received the M.Sc. degree in mechanical engineering and the Ph.D. degree in applied mathematics from Linköping University, Linköping, Sweden, in 1996 and 2002, respectively.

He is currently an Assistant Professor with the Department of Electrical Engineering, Linköping University. His current research interests include model-based fault diagnosis and structural methods.

Mattias Nyberg was born in Sweden in 1970. He received the M.Sc. degree in computer science and engineering from Linköping University, Linköping, Sweden, in 1994, the M.Sc. degree in electrical engineering and applied physics from Case Western Reserve University, Cleveland, OH, in 1994, and the Ph.D. degree in electrical engineering from Linköping University, in 1999.

He has spent one year with Daimler Chrysler Research, Stuttgart, Germany, as a Postdoctoral Researcher. Since 1999, he has been an Assistant Professor with the Department of Electrical Engineering, Linköping University. Since 2002, he has also been working mainly at the truck manufacturer Scania, Södertälje, Sweden, where he is responsible for development and research in onboard diagnosis. His research interest is in model-based diagnosis and, particularly, methods for fault isolation and residual generation.