An Efficient Algorithm for Finding Over-constrained Sub-systems for Construction of Diagnostic Tests

Mattias Krysander, Jan Åslund, Mattias Nyberg
Linköpings universitet, Dept. of Electrical Engineering
SE - 581 83 Linköping, Sweden
{matkr, jaasl, matny}@isy.liu.se

Abstract

In this paper, a new algorithm for computing all minimal over-constrained sub-systems in a structural model is proposed. To handle large differential algebraic models in diagnosis, systematic structural approaches to find testable sub-systems have been suggested. It is shown how the algorithm can be incorporated and improve some of them. Previous algorithms are recalled and it is shown that the new algorithm is 14000 times faster when applied to a Scania truck engine model.

1 Introduction

In model based diagnosis, the diagnostic system construction is based on a model of the technical system to be diagnosed. To cope with large differential algebraic models, systematic structural approaches to find testable sub-systems have been suggested in e.g. [Cassar and Staroswiecki, 1997], [Blanke *et al.*, 2003], [Pulido and Alonso, 2002] [Travé-Massuyès *et al.*, 2001], and [Krysander and Nyberg, 2002]. All these approaches have in common that testable sub-systems are found among the over-constrained sub-systems. Furthermore, of all over-constrained sub-systems, it is the minimal ones that are used to derive analytical redundancy relations. Several algorithms for computing all minimal over-constrained sub-systems have been proposed in [Pulido and Alonso, 2002], [Krysander and Nyberg, 2002], and [Blanke *et al.*, 2003]. However all these algorithms run into complexity problems when considering large industrial examples.

In this paper we present a new algorithm for computing all minimal over-constrained sub-systems 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 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. All minimal over-constrained sub-systems were computed with the new algorithm more than 14000 times faster than with previous algorithms.

Three different types of structural representations used to describe differential algebraic systems are recalled in Section 2. We introduce a notion, i.e. *minimal structurally overdetermined* (MSO) set of equations, which characterize over-constrained sub-systems independent of structural representation. Several other proposed structural characterizations of over-constrained sub-systems are then recapitulated in Section 3. All these are MSO sets of equations and this means that the proposed algorithm can easily be used in any of these approaches to find over-constrained sub-systems. For comparison, previous algorithms for finding over-constrained sub-systems are recalled in Section 4. In Section 5 a basic algorithm for finding all MSO sets will be presented. This algorithm illustrates the basic ideas and then in Section 6 further improvements are described. Finally in Section 7, 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 compared to the previous ones and the complexity of the different algorithms are discussed.

2 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

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

where u and y are known, and x_1 and x_2 are unknown signals.

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 [Frisk *et al.*, 2003].

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

$$\begin{aligned} \dot{e}_2 &: \dot{x}_2 = 2\,x_1\,\dot{x}_1 \\ \dot{e}_3 &: \dot{y} = \dot{x}_2 \end{aligned}$$

The extended structure is then

equation	unknown			
	x_1	x_1'	x_2	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 modelling principle is used in [Krysander and Nyberg, 2002].

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	x_1'	x_2
e_1	X	X	
e_2	X		X
e_3			X
d	X	X	

(4)

where d is the added differential equation. This representation is used for diagnosis in [Blanke *et al.*, 2003].

3 Use of MSO Sets for Test Construction

Several structural approaches to find testable sub-systems have previously been suggested. In this section a structural characterization of testable sub-systems is presented.

First some important structural properties will be defined. The biadjacency matrix in Figure 1 shows a Dulmage-Mendelsohn canonical decomposition [Dulmage and Mendelsohn, 1958] 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 grey-shaded areas contain ones and zeros, while the white areas only contain zeros. The thick line

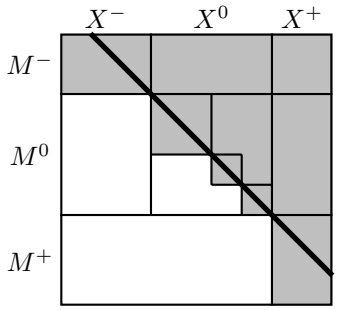

Figure 1: A Dulmage-Mendelsohn decomposition.

represents a maximal matching in the graph defined by this biadjacency matrix. The model M is decomposed into three parts where the one denoted 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.

In consistency based diagnosis, redundancy in the model is used and this motivates the following definitions.

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

Definition 2 (Proper structurally overdetermined) *A structurally overdetermined set M is a* proper structurally overdetermined *(PSO) set if $M = M^+$.*

Definition 3 (Minimal structurally overdetermined) *A structurally overdetermined set is a* minimal structurally overdetermined *(MSO) set if no proper subset is a structurally overdetermined 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 have one more equation than the number of unknowns. From the equation system (3) the equation

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

can be derived by algebraic elimination. 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). Hence the testable system (1) has a corresponding MSO model for each structural representation.

Since MSO sets can be used in any structural representation, comparisons to other structural characterizations of testable models using different representations are possible. In [Krysander and Nyberg, 2002] and [Frisk *et al.*, 2003] MSO sets are used to find testable sub-systems. In [Pulido and Alonso, 2002] *minimal evaluable chains* are used, which are MSO models with the additional requirement that they contain known variables. In [Blanke *et al.*, 2003] and [Travé-Massuyès *et al.*, 2001] *redundancy relations* are used, which also correspond to MSO models.

In conclusion, MSO models are used in all structural approaches discussed above. The example in (1)- (4) shows that MSO models can be interpreted in a similar way independent

of structural representation. By noting these similarities the algorithm that will be proposed in Section 6 can be applied to any structural representation considered and be used in the previously discussed approaches.

4 Previous Algorithms

Two main ideas for finding all MSO sets have previously been presented. These will be recalled and compared. These will be used to compare the computational complexity of the new algorithm proposed in this paper.

The first approach for finding all MSO sets is presented independently in [Krysander and Nyberg, 2002] and [Pulido and Alonso, 2002]. 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.

The second approach for finding all MSO sets is presented in [Blanke *et al.*, 2003]. All maximal matchings are enumerated. 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.

The computation time of both approaches grows fast with the number of equations. Therefore they cannot be used in practice for large industrial examples. Each set of equations considered in the second approach is analyzed as least as many times as in the first approach. Hence the computational complexity of the second approach is not better than for the first one, and a comparison can be found in [Rattfält, 2004]. Therefore only the first approach will be considered when comparing the computational complexity with the new algorithm developed in this paper.

5 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 proper structurally overdetermined set of equations M, the structural redundancy $\bar{\varphi}\, M$ is defined by

$$\bar{\varphi}\, M = |M| - |\mathrm{var}_X(M)|$$

The algorithm will be based on the following three lemmas.

Lemma 1 *If M is a proper structurally overdetermined set of equations and $e \in M$, then $\bar{\varphi}\,(M \backslash \{e\})^+ = \bar{\varphi}\, M - 1$.*

Lemma 2 *The set of equations M is an MSO set if and only if M is a proper structurally overdetermined set and $\bar{\varphi}\, M = 1$.*

Lemma 3 *If M is a set of equations, $E \subseteq M$ is an PSO set, and $e \in M \setminus E$, then*

$$E \subseteq (M \setminus \{e\})^+$$

The proofs of all lemmas and theorems in this paper can be found in [Krysander *et al.*, 2005]. The first lemma reveals how the structural redundancy decreases when one equation is removed. It follows from this lemma that if we start with any proper structurally overdetermined set of equations we can alternately remove equations and computing the overdetermined part until the structural redundancy becomes 1. We have then found an MSO-set, according to Lemma 2. Finally, Lemma 3 implies that an arbitrary MSO set can be obtained recursively 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 $\bar{\varphi}\, M = 1$ ***then***

$\mathcal{M}_{MSO} := \{M\}$;

else

$\mathcal{M}_{MSO} := \varnothing$;

for each *equation* e *in* M ***do***

$M' := (M \setminus \{e\})^+$;
$\mathcal{M}_{MSO} := \mathcal{M}_{MSO} \cup \texttt{FindMSO}(M')$;

end for

end if
return $\mathcal{M}_{MSO}$

From the discussion above, 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 proper structurally overdetermined model consisting of four equations and two unknown variables:

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

(5)

The structural redundancy of this set of equations is 2. When entering the algorithm, e_1 is removed and the set M' 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.

6 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.

6.1 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 Figure 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

$$e' \notin (M \setminus \{e\})^+ \tag{6}$$

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 (6) and Lemma 3, with E replaced by $(M \setminus \{e\})^+$, that $(M \setminus \{e\})^+ \subseteq (M \setminus \{e'\})^+$. Lemma 1 and Lemma 3 imply that both sets have the same structural redundancy and that $(M \setminus \{e\})^+ = (M \setminus \{e'\})^+$. 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 $(M \setminus \{e_1\})^+ = (M \setminus \{e_2\})^+ = (M \setminus \{e_3\})^+$, which implies that R is transitive. The relation R is therefore an equivalence relation and the equivalence class containing e is denoted by $[e]$.

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

$$X_0 = X \setminus (\bigcup_{i \neq 0} X_i)$$

It follows from Lemma 1, by considering the complementary sets, that

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

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

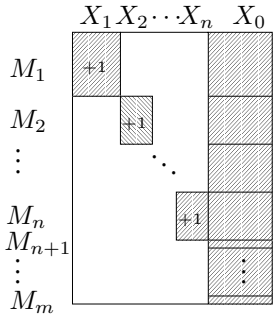

Figure 2: A structural decomposition of a PSO set.

Theorem 4 *If $E \subseteq M$ is a PSO set, then E is a union of equivalence classes defined by (6), i.e.*

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

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

This theorem motivates why the equations in each equivalence class can be lumped together when computing all PSO sets. A new bipartite graph is formed with equivalence classes $\{M_i\}$ and the unknowns X_0 as node sets. The unknowns connected to M_i are $\text{var}_{X_0}(M_i)$. The reduction of (5) 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 PSO set, MSO set, and structural redundancy are thereby extended to lumped structures. In the example above 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 together with Theorem 4 shows that there is a one-to-one correspondence between the PSO sets in the original and in the lumped structure.

Theorem 5 *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.*

It follows from the discussion above that the reduced structure can be used to find all PSO sets in the original structure.

6.2 Improved Algorithm

A drawback with Algorithm 1, presented in Section 5, is that some of the MSO sets are found more than once. There are two main reasons why this happens and these can be illustrated using the following example:

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

(7)

First, the same PSO set $\{e_3, e_4, e_5\}$ is obtained, if either e_1 or e_2 is removed. This is avoided by using the lumping described in the previous section. Second, the same MSO set is obtained if the removed equations are permuted. For example, the MSO set $\{e_4, e_5\}$ is obtained if first e_1 and then e_3 is removed but also if the order of removal is reversed. In the

next algorithm such permutations are prevented and the set $\mathcal{R}$ is an additional input argument, in the recursive calls, that contains the equations that are allowed to be removed.

Lumping can be extended and applied to subsets of previously lumped structures. Sets of equations are then lumped together into new sets of equations by taking the union of the sets in the equivalence class. In each call of the subroutine, lumping of the equivalence classes is used. We illustrate this with an example. Assume that we start with 6 equations and that e_2 and e_3 are lumped together and the following structure is obtained:

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

(8)

In the first recursive call $\{e_1\}$ is removed and the graph corresponding to the remaining part has the same structure as (5). Now, lumping is performed and the sets $\{e_2, e_3\}$ and $\{e_4\}$ are lumped together into the set $\{e_2, e_3, e_4\}$.

A problem is how to form the new set of equations that are allowed to be removed, i.e. the set corresponding to $\mathcal{R}$, for the lumped structure and the following principle will be used. If all the original equivalence classes in a lumped set are included in $\mathcal{R}$, then the lumped set is included in the new set, denoted by $\mathcal{R}'$. Otherwise the lumped set is excluded from $\mathcal{R}'$. It can be shown that, in this way, all MSO sets are found once and only once. The algorithm can formally be defined as follows.

Algorithm 2 $\mathcal{M}_{MSO} = \texttt{MSO}(M)$
$\mathcal{M} := \{\{e\} | e \in M^+\}$;
$\mathcal{M}_{MSO} := \texttt{MSOsubsets}(\mathcal{M}, \mathcal{M})$;
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
  $\mathcal{R}' := \varnothing$; $\mathcal{M}' := \mathcal{M}$;
  while $\mathcal{R} \neq \varnothing$ ***do***
    Select an $E \in \mathcal{R}$;
    Lump $[E]$ *in* $\mathcal{M}'$;
    if $[E] \subseteq \mathcal{R}$ ***then***
      $\mathcal{R}' := \mathcal{R}' \cup \{\cup_{E' \in [E]} E'\}$;
    end if
    $\mathcal{R} := \mathcal{R} \setminus [E]$;
  end while
  $\mathcal{M}_{MSO} := \varnothing$;
  while $\mathcal{R}' \neq \varnothing$ ***do***
    Select an $E \in \mathcal{R}'$
    $\mathcal{R}' := \mathcal{R}' \setminus \{E\}$;
    $\mathcal{M}_{MSO} := \mathcal{M}_{MSO} \cup \texttt{FindMSO}(\mathcal{M}' \setminus \{E\}, \mathcal{R}')$;
  end while
end if
return $\mathcal{M}_{MSO}$

By *lump* $[E]$ *in* $\mathcal{M}'$, in the algorithm, we mean that only the equivalence class $[E]$ in $\mathcal{M}'$ is lumped and that the other equations remain unchanged. Note that only classes that intersect $\mathcal{R}$ are lumped in the subroutine, which is sufficient in order to avoid that the same MSO set is found more than once. To illustrate the algorithm, we use the set (7) and describe the first steps in the recursion. First, the subroutine is called with input sets $\mathcal{M} = \mathcal{R} = \{\{e_1\}, \{e_2\}, \{e_3\}, \{e_4\}, \{e_5\}\}$. In the first *while* loop, lumping is performed with the resulting sets $\mathcal{M}' = \mathcal{R}' = \{\{e_1, e_2\}, \{e_3\}, \{e_4\}, \{e_5\}\}$. Then the subroutine is called recursively with the following four pairs of input sets

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

6.3 Computational Complexity

The structural redundancy depends on the number of available sensors, which are often expensive, and therefore the structural redundancy is low in many applications. One example of this is given in the next section. 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 exponential. This follows from the fact that the number of subroutine calls are equal to the number of PSO sets, which grows polynomially, and that the computational complexity to obtain the set M^+ is polynomial. It should be pointed out that, in the case of few unknowns, the roles are reversed. For a fixed number of unknowns, the complexity of the new algorithm is exponential and the complexity of the old algorithm is polynomial in the number of equations. However, this situation is not common in diagnosis applications.

7 Application to a Large Industrial Example

To demonstrate the efficiency of the algorithm, described in the previous section, we will here apply it to a real industrial process. The process is a Scania truck diesel-engine and a sketch is shown in Figure 3. This engine has two actuators, namely the fuel injection δ and the 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 is presented in [Eriksson, 2004].

A simulation model of the engine was provided in Simulink. This model has 4 states and 4 outputs. These 4 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 4.

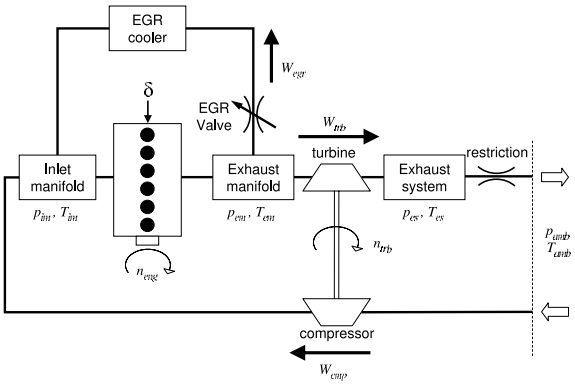

Figure 3: Example of a Scania truck engine.

The fact that the structural redundancy is 4 is a consequence of that the number of outputs is 4.

For comparison, three algorithms were tested on the set of 126 equations. The first is the old MSO algorithm presented in [Krysander and Nyberg, 2002], 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 5 with the structural reduction in Section 6.1 applied initially, reducing the number of equations to 28. The third is the new improved algorithm presented in Section 6.

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 1.

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

Table 1: A comparison of three MSO algorithms.

In the table we can see that the new MSO algorithm is more than 14000 times faster than the old algorithm!

8 Conclusions

A new approach to compute minimal structurally overdetermined sets of equations was developed. The proposed algorithm can be used in other structural approaches for finding testable sub-systems. 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 5. 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 exponential. The efficiency of the algorithm was demonstrated by applying the new and previous algorithms to a model of a Scania truck engine.

References

[Blanke *et al.*, 2003] M. Blanke, M. Kinnert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer-Verlag, 2003.

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

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

[Eriksson, 2004] Lars Eriksson. Structural algorithms for diagnostic system design using simulink models. Master's thesis, Linköpings universitet, 2004.

[Frisk *et al.*, 2003] Erik Frisk, Dilek Dştegör, Mattias Krysander, and Vincent Cocquempot. Improving fault isolability properties by structural analysis of faulty behavior models: application to the DAMADICS benchmark problem. In *Proceedings of IFAC Safeprocess 03*, Washington, USA, 2003.

[Krysander and Nyberg, 2002] Mattias Krysander and Mattias Nyberg. Structural analysis utilizing MSS sets with application to a paper plant. In *Proc. of the Thirteenth International Workshop on Principles of Diagnosis*, Semmering, Austria, May 2002.

[Krysander *et al.*, 2005] Mattias Krysander, Mattias Nyberg, and Jan Åslund. An efficient algorithm for finding over-constrained sub-systems for construction of diagnostic tests. Technical Report LiTH-ISY-R-2679, Dept. of Electrical Engineering Linköpings universitet, 2005. URL:http://www.vehicular.isy.liu.se/Publications/.

[Pulido and Alonso, 2002] B. Pulido and C. Alonso. Possible conflicts, ARRs, and conflicts. In M. Stumptner and F. Wotawa, editors, *Proceedings DX-2002*, pages 122–127, Semmering, Austria, 2002.

[Rattfält, 2004] Linda Rattfält. A comparative study of two structural methods for fault isolability analysis. Master's thesis, Linköpings Universitet, SE-581 83 Linköping, 2004. URL:http://www.vehicular.isy.liu.se/Publications/.

[Travé-Massuyès *et al.*, 2001] L. Travé-Massuyès, T. Escobet, and R. Milne. Model-based diagnosability and sensor placement. application to a frame 6 gas turbine subsystem. In *DX01 twelfth international workshop on principles of diagnosis*, pages 205–212, 2001.