MSO Sets and MTES for Dummies

Maxence Glotin
LAAS-CNRS, Université de Toulouse, INSA, France

Louise Travé-Massuyès
LAAS-CNRS, Université de Toulouse, CNRS, France

Elodie Chanthery
LAAS-CNRS, Université de Toulouse, INSA, France

Abstract

Structural analysis-based diagnosis allows for the extraction of a wealth of information and properties by studying a structural model that represents a physical system. This diagnosis approach is centered on structurally overdetermined sets, which enable the generation of residuals for fault detection and isolation. As the 'for Dummies' editorial collection, this article aims at taking on complex concepts and making them easy to understand. It aims to clarify and compare key concepts in structural analysis, focusing on Minimally Structurally Overdetermined (MSO) sets and Minimal Test Equation Supports (MTES). Additionally, we explain and illustrate the Dulmage-Mendelsohn decomposition, which helps identify structurally overdetermined parts of the system and plays a important role in the structural analysis process. Through detailed exploration and practical examples, we demonstrate the roles, applications, and interrelations of these sets, highlighting their respective strengths and limitations. The paper provides an overview of the algorithms used to identify and use these sets, including a theoretical and practical comparison of their computational efficiency and diagnostic capabilities.

2012 ACM Subject Classification Computing methodologies → Knowledge representation and reasoning

Keywords and phrases Structural analysis, MTES, MSO sets

Digital Object Identifier 10.4230/OASIcs.DX.2024.13

Funding This work is partially supported by the Artificial and Natural Intelligence Toulouse Institute (ANITI), French "Investing for the Future - PIA3" program under the Grant agreement ANR-19-PI3A-0004.

1 Introduction

In the field of model-based fault diagnosis, the ability to detect and isolate faults within complex systems is crucial for ensuring reliability and safety. By studying the model, it is possible to identify faults, which are indicative of abnormal or faulty behavior within the system.

Structural analysis consists in abstracting a model by keeping only the links between equations and variables. The main advantages of structural analysis are that the approach can be applied to large scale systems, linear or non linear, even under uncertainty. In the field, the concept of Minimally Structurally Overdetermined (MSO) sets [6] has been widely adopted due to its effectiveness in isolating faults by leveraging the redundancy in system models. However, while MSO sets have proven valuable, they come with limitations, particularly in terms of computational complexity when applied to large-scale systems.

To address these issues, more refined approaches have been introduced, such as Minimal Test Equation Supports (MTES), presented in [5], or Fault Driven Minimal Structurally Overdetermined (FMSO) sets introduced by [7]. These methods aim to enhance diagnosis precision while aiming at reducing computational demands, yet their practical application and comparative advantages remain areas of ongoing exploration.

35th International Conference on Principles of Diagnosis and Resilient Systems (DX 2024).
Editors: Ingo Pill, Avraham Natan, and Franz Wotawa; Article No. 13; pp. 13:1–13:15
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

As the 'for Dummies' editorial collection, this article aims at taking on complex concepts and making them easy to understand. The objective is to clarify and compare various key concepts used in structural analysis model-based fault diagnosis, specifically focusing on MSO sets, MTES, or FMSO sets. By providing a detailed exploration of these concepts, this paper seeks to clarify their roles, applications, and interrelations. Through carefully chosen examples, we illustrate how each of these concepts can be applied to different scenarios, highlighting their respective strengths and limitations. This comparative analysis is intended to enhance the understanding of these diagnosis tools.

This paper is structured as follows. Section 1 gives some theoretical foundations on structural analysis. Then in Section 3 MSO sets are defined and explained. The algorithm for their identification is also presented. Next, Section 4 discuss the motivations behind the introduction of MTES and describe the algorithm for their computation. A comparison between the performance of MSO and MTES is then conducted in Section 5. To illustrate the practical application of these concepts, a realistic example is provided in Section 6, including structural modeling, algorithmic complexity analysis, and an evaluation of diagnostic capability. Section 7 concludes the paper.

2 Background

One of the key concepts to understand the notion of MSO sets or even MTES is the Dulmage Mendelsohn decomposition of a bipartite graph [4].

Let the system description consist of a set of n_e equations involving a set of variables partitioned into a set Z of n_Z known (or measured) variables and a set X of n_X unknown (or unmeasured) variables. We refer to the vector of known variables as z and the vector of unknown variables as x. The system may be impacted by the presence of n_f faults that appear as parameters in the equations. The set of faults is denoted by F and we refer to the vector of faults as $\boldsymbol{f}$.

▶ **Definition 1** (System). *A system, denoted $\Sigma(z, x, \boldsymbol{f})$ or Σ for short, is any set of equations relating z, x and $\boldsymbol{f}$. The equations $e_i(z, x) \subseteq \Sigma(z, x, \boldsymbol{f})$, $i = 1, \dots, n_e$, are assumed to be differential or algebraic in z and x.*

Consider the illustrative example given by system (1), from [8], named Σ_1 that will be taken as a running example in this section. It is composed of six equations e_1 to e_6 relating the known variables $Z = \{z_1, z_2\}$, the unknown variables $X = \{x_1, x_2, x_3, x_4, x_5\}$ and the set of system faults $F = \{f_1, f_2\}$. Besides a, b, c are constant parameters.

$$\begin{aligned}
e_1 &: \quad \dot{x}_3 = e^{x_3} - a \\
e_2 &: \quad x_3^2 = -b\dot{x}_4 + f_1 \\
e_3 &: \quad z_1 = x_4 \\
e_4 &: \quad z_2 = x_1 + b^2 + x_4 \\
e_5 &: \quad \dot{x}_1 = e^{x_2} + x_5 \\
e_6 &: \quad \dot{x}_3 = x_4 + c + f_2
\end{aligned} \tag{1}$$

We can represent a system $\Sigma(z, x, \boldsymbol{f})$ by a biadjacency matrix, in which each row stands for an equation and each column an unknown variable. An X in position (i, j) means that equation i contains the unknown variable j. For example, Σ_1 can be represented by the biadjacency matrix of Table 1.

Table 1 Biadjacency matrix of Σ_1.

	x_1	x_2	x_3	x_4	x_5
e_1			×		
e_2			×	×	
e_3				×	
e_4	×			×	
e_5	×	×			×
e_6			×	×	

Equivalently, a system $\Sigma(z, x, f)$ can be represented by a bipartite graph $G(E, X)$. E and X are the sets of nodes, an edge between node E_i and node X_j means that equation i contains the unknown variable j. The bipartite graph associated with the illustrative example Σ_1 is shown in Figure 1 [8].

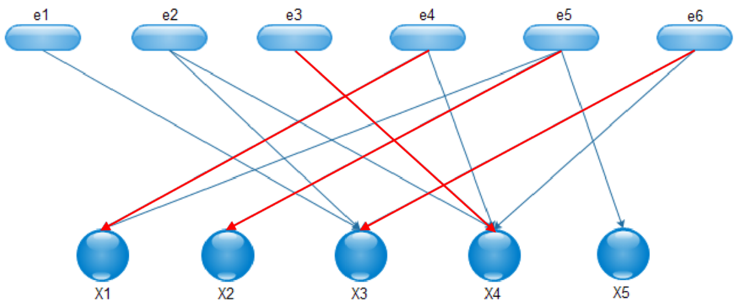

Figure 1 The bipartite graph of Σ_1 [8].

▶ **Definition 2** (Maximal Matching). *A maximal matching is a set of edges such that these edges do not share any common nodes, and no more edges can be added to this set without losing this property.*

In Table 1, as well as in Figure 1, a maximal matching is marked in red. It is the largest set of edges such that there are no common nodes (each equation and each variable appears at most once).

▶ **Definition 3** (Free equation). *For a given maximal matching, an equation is free if it is not part of this maximal matching.*

In the matching represented in Table 1, e_1 and e_2 are free.

▶ **Definition 4** (Free variable). *For a given maximal matching, a variable is free if it is not part of this maximal matching.*

In the matching represented in Table 1, x_5 is free.

▶ **Definition 5** (Alternating path). *Let M_{max} be a maximal matching. P, a path in the graph G, is an alternating path if it successively passes through edges that belong to M_{max} and edges that do not belong to M_{max}.*

The Dulmage-Mendelsohn decomposition allows separating a model Σ into 3 parts: an overdetermined part, denoted Σ^+, an exactly determined part, denoted Σ^0, and an underdetermined part, denoted Σ^-, as shown in Figure 2 .

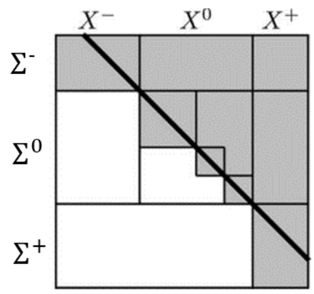

Figure 2 Dulmage-Mendelsohn decomposition of a model Σ.

The overdetermined part Σ^+ of Σ is the set of equations $e \in E$ such that, for any maximal matching, there exists an alternating path between at least one free equation and e [6]. Σ^+ can also be defined as the set of equations $e \in E$ such that there is at least one maximal matching in which e is free.

The part Σ^0 contains equations that are not in Σ^+ but are such that the set $\Sigma^0 \cup \Sigma^+$ is the maximal set of equations for which there exists a complete matching of the unknown variables in $\Sigma^0 \cup \Sigma^+$ into $\Sigma^0 \cup \Sigma^+$ [6]. We can also say that Σ^0 consists of the equations e that are matched in every maximal matching and do not contain any free variables.

The rest of the equations belongs to Σ^-. Σ^- only contains equations if the maximal matchings do not cover all the variables (if there exists at least one free variable in a maximal matching). Σ^- contains equations associated with the free variables.

Running example: Σ_1^+ contains all the equations that are free in at least one maximal matching. In the matching represented in Table 1, e_1 and e_2 are free so these two equations are in Σ_1^+. But $(e_1x_3, e_2x_4, e_4x_1, e_5x_2)$ is also a maximal matching, and in this matching, e_3 and e_6 are free. So e_3 and e_6 are also in Σ_1^+. So, we have $\Sigma_1^+ = \{e_1, e_2, e_3, e_6\}$.

$\Sigma_1^0 = \{e_4\}$ because the set $\{e_1, e_2, e_3, e_4, e_6\}$ is the biggest set such that there exists a complete matching of the unknown variables in this set into this set. If we add e_5 in that set, it would not be a complete matching because either x_2 or x_5 would not be part of the matching. Another way to see this is that e_4 is necessary to every maximal matching, as it is the only equation that contains x_1 (e_5 also contains x_1, but e_5 is used to cover x_2 or x_5). Thus, it is impossible to obtain a maximal matching without e_4, and e_4 does not contain any free variable, so e_4 is part of Σ_1^0.

The remaining equation, e_5, is in Σ_1^-. That is because e_5 is the only equation that contains both x_2 and x_5, so for every maximal matching, one of these two variables is free.

Note that, as shown above with Σ_1, if a set of equations Σ has more equations than unknown variables, it does not necessarily mean that Σ^0 and Σ^- are empty.

▶ **Definition 6** (PSO set). *A set Σ of equations is a PSO set (proper structurally overdetermined set) if $\Sigma = \Sigma^+$. In other words, Σ is a PSO set if $\Sigma^0 = \Sigma^- = \emptyset$.*

System represented Σ_1 as a whole is not a PSO set because Σ^0 and Σ^- are not empty. But the subset of equations $\{e_1, e_2, e_3\}$ is a PSO set: the three equations are free in at least one maximal matching, so $\Sigma = \Sigma^+$ (see Table 2).

Table 2 The subset of equations $\{e_1, e_2, e_3\}$.

	x_3	x_4
e_1	×	
e_2	×	×
e_3		×

▶ **Definition 7** (Structural redundancy)**.** *For a given set of equations Σ, the structural redundancy $\varphi(\Sigma)$ is defined by $\varphi(\Sigma) = |\Sigma^+| - |X^+|$, where $|\Sigma^+|$ is the number of equations in Σ^+, and $|X^+|$ is the number of unknown variables involved in Σ^+.*

PSO sets have structural redundancy, which is key to diagnosis since it allows residual generation. We recall hereafter some important definitions [8].

▶ **Definition 8** (ARR for $\Sigma(z, x, f)$)**.** *Let $\Sigma(z, x, f)$ be a system. Then, a relation of the form $arr(z, \dot{z}, \ddot{z}, ...) = r$ is an Analytical Redundancy Relation (ARR) for $\Sigma(z, x, f)$ if for each z consistent with $\Sigma(z, x, f)$ the relation evaluation is $r = 0$, i.e., the ARR is satisfied.*

▶ **Definition 9** (Residual Generator for $\Sigma(z, x, \mathtt{f})$)**.** *A system taking a subset of the variables z as input, and generating a scalar signal r as output, is a residual generator for the model $\Sigma(z, x, f)$ if, for all z consistent with $\Sigma(z, x, f)$, it holds that $\lim_{t\to\infty} r(t) = 0$.*

ARRs are residual generators and are used to verify the consistency of observations with the system model. The ARRs hold true if the observed behavior of the system meets the model constraints. They can be derived from the system model by eliminating unknown variables [7]. PSO sets can be used to generate ARRs as they identify the redundant parts of the system's model.

3 MSO sets

It is often more practical to focus on smaller, manageable subsets of the PSO sets that still retain the necessary redundancy for effective fault diagnosis.

▶ **Definition 10** (MSO set)**.** *A PSO set Σ is an MSO set (minimal structurally overdetermined) if no proper subset is a PSO set. MSO sets are PSO sets such that $\varphi(\Sigma) = 1$.*

MSO sets are interesting because they are just overdetermined and exactly identify a set of equations from which a residual generator can be obtained. [6].

In our running example, the subset $\{e_1, e_2, e_3\}$ is an MSO set because it is a PSO set, and there is no subset which is also a PSO set. Furthermore, $\varphi(\Sigma) = 3 - 2 = 1$.

A deep first search algorithm[1] to find MSO sets is presented in [6]. The general idea is to start with a PSO set Σ, to remove an equation e, and to compute the overdetermined part of the remaining set $(\Sigma \setminus \{e\})^+$. When an equation e is removed from a PSO set, the structural redundancy decreases by one. So the previous process is done until the structural redundancy becomes one. If $\varphi((\Sigma \setminus \{e\})^+) = 1$, then an MSO set is found.

Running example: To illustrate that, let us apply the first steps of the algorithm to the following example Σ_2 (cf. Table 3).

[1] All the algorithms presented in this article are provided in the "Fault Diagnosis Toolbox" available here: `https://faultdiagnosistoolbox.github.io/`

Table 3 Illustrative example Σ_2.

	x_1	x_2	x_3
e_1	×		
e_2		×	×
e_3	×	×	
e_4			×
e_5	×		×

The set $\Sigma_2 = \{e_1, e_2, e_3, e_4, e_5\}$ is a PSO set of structural redundancy 2. First, e_1 is removed. By applying the Dulmage-Mendelsohn decomposition on the remaining set, we have $(\Sigma_2 \setminus \{e_1\})^+ = \{e_2, e_3, e_4, e_5\}$. $\varphi((\Sigma_2 \setminus \{e_1\})^+) = 1$, which means that $\{e_2, e_3, e_4, e_5\}$ is an MSO set. Then, e_2 is removed from Σ_2. By applying the Dulmage-Mendelsohn decomposition on the remaining set, we have $(\Sigma_2 \setminus \{e_2\})^+ = \{e_1, e_4, e_5\}$. $\varphi((\Sigma_2 \setminus \{e_2\})^+) = 1$, which means that $\{e_1, e_4, e_5\}$ is also an MSO set. And so on, until covering the entire tree.

Using this algorithm, all the MSO sets are found. However, this method is not optimal in terms of computation time because some MSO sets can be found multiple times. For example, here, when removing e_2, $(\Sigma_2 \setminus \{e_2\})^+) = (e_1, e_4, e_5)$ and when removing e_3, $(\Sigma_2 \setminus \{e_3\})^+) = (e_1, e_4, e_5)$. Therefore, the same MSO set is found twice. That is why the concept of equivalence class is introduced.

▶ **Definition 11** (Equivalence class). *Let Σ be a set of equations and (e_i, e_j), two equations in Σ. e_i and e_j are in the same equivalence class $[e] = \{e_i, e_j\}$ if $e_i \notin (\Sigma \setminus \{e_j\})^+$.*

This enables grouping equations into the same class such that, if they are removed from Σ, the same overdetermined part is obtained.

For the system Σ_2, the equivalences classes are $\{e_1\}$, $\{e_2, e_3\}$, $\{e_4\}$, $\{e_5\}$ because $e_2 \notin (\Sigma_2 \setminus \{e_3\})^+$ (the inverse is also true because it is an equivalence relation).

Thus, in the algorithm, it is no longer one single equation that is removed at each step, but equivalence classes, which prevent the same overdetermined part from being computed twice. At each step, equations that belong to the same equivalence class are lumped, one equivalence class is removed, and the overdetermined part of the remaining equations is computed. Note that for Σ_2, this optimization does not make a huge difference in terms of computation time, but it makes a big difference for huge models, with many equations and MSO sets.

There is another issue handled by the algorithm. The same MSO set can be obtained if the order of equations removal is permuted. Let us consider the following system Σ_3 (cf. Table 4).

Table 4 Illustrative example Σ_3.

	x_1	x_2	x_3
e_1		×	
e_2	×	×	×
e_3	×		
e_4		×	×
e_5	×		
e_6			×

At the beginning, e_1 is removed. The overdetermined part of the remaining equations $(\Sigma_3 \setminus \{e_1\})^+$ is $\{e_2, e_3, e_4, e_5, e_6\}$. Then, e_2 is removed, and the overdetermined part of the remaining equations $(\Sigma_3 \setminus \{e_1, e_2\})^+$ is $\{e_3, e_5\}$, which is an MSO set. But if e_2 is removed first, and then e_1, the same MSO set is obtained. To handle this problem, the algorithm uses a new input set R which contains the equivalence classes that are allowed to be removed.

At each node traversed by the algorithm, a Dulmage-Mendelsohn decomposition is performed. This is the most time-consuming part of the computation. This algorithm traverses a search tree, where each node represents a potentially overdetermined set of equations. Each level of the tree corresponds to the removal of an equation from the system, and each branch represents a subset of equations being tested. When an equation is removed from Σ (i.e., when moving down a level in the search tree), the structural redundancy $\varphi(\Sigma)$ decreases by 1 [6].

Consider the worst-case scenario, where each equation belongs to a different equivalence class (i.e., the equations are removed from Σ one by one). For a system with n_e equations and a redundancy degree of φ, the maximum number of nodes the algorithm might traverse is given by the sum of possible equation subset for each level k of the tree. A bound on the number of nodes traversed by the algorithm to find MSO sets is then given by Equation (2).

$$\sum_{k=0}^{\varphi-1} \binom{n_e}{k} \tag{2}$$

MSO sets are used to construct ARRs, and thus residual generators, but it is important to note that MSO sets do not take faults into account. If we are interested in system performance and we want to construct residual generators for diagnosis purposes, only MSO sets that contain equations possibly disrupted by faults are important.

▶ **Definition 12** (Detectable fault). *A fault f is structurally detectable in a set of equations Σ if f is in the overdetermined part Σ^+.*

▶ **Definition 13** (Fault support). *The fault support $F_{\Sigma'}$ of a set of equations $\Sigma' \subseteq \Sigma$ is defined as the set of faults that are involved in the equations of Σ' [8].*

A special class of MSO sets can be defined: FMSO sets (Fault-Driven Minimal Structurally Overdetermined) that are MSO sets with a non empty fault support [8].

▶ **Definition 14** (FMSO set). *A subset of equations $\Sigma^{'} \subseteq \Sigma(z, x, f)$ is an FMSO set of $\Sigma(z, x, f)$ if (1) $F_{\Sigma'} \neq \emptyset$ and $\varphi(\Sigma^{'}) = 1$ that means $|\Sigma^{'}| = |X_{\Sigma'}| + 1$, (2) no proper subset of $\Sigma^{'}$ is overdeterminated.*

Therefore, if there exists a FMSO set in Σ which contains the fault f, then f is detectable in Σ. Those special MSO sets are candidates as residual generators as they contain just the necessary redundancy and their fault support is not empty.

4 MTES

4.1 Motivation

MSO sets can be used to obtain residual generators, but are they all useful ? Let us take the previous system Σ_3 and add faults of the set $F = \{f_1, f_2, f_3\}$ (cf. Table 5).

Table 5 System Σ_3 with faults.

	x_1	x_2	x_3	
e_1		$\times$		f_1
e_2	$\times$	$\times$	$\times$	f_2
e_3	$\times$			
e_4		$\times$	$\times$	
e_5	$\times$			
e_6			$\times$	f_3

For this example, there are 8 MSO sets given in Table 6.

Table 6 MSO sets and their fault support for Σ_3 with faults.

	MSO sets	Faults
MSO_1	$\{e_3, e_5\}$	$\{\emptyset\}$
MSO_2	$\{e_2, e_4, e_5, e_6\}$	$\{f_2, f_3\}$
MSO_3	$\{e_2, e_3, e_4, e_6\}$	$\{f_2, f_3\}$
MSO_4	$\{e_1, e_4, e_6\}$	$\{f_1, f_3\}$
MSO_5	$\{e_1, e_2, e_5, e_6\}$	$\{f_1, f_2, f_3\}$
MSO_6	$\{e_1, e_2, e_4, e_5\}$	$\{f_1, f_2\}$
MSO_7	$\{e_1, e_2, e_3, e_6\}$	$\{f_1, f_2, f_3\}$
MSO_8	$\{e_1, e_2, e_3, e_4\}$	$\{f_1, f_2\}$

MSO_2 and MSO_3 as well as MSO_6 and MSO_8 have the same fault support $\{f_2, f_3\}$ and $\{f_1, f_2\}$, respectively. This means that the generated residuals would be sensitive to the same faults. Note also that if the equations of MSO_2 and MSO_3 are put together, the fault support of the resulting subset of equations remains $\{f_2, f_3\}$. In addition, some MSO sets have a non minimal fault support like MSO_5 whose fault support is $\{f_1, f_2, f_3\}$ that includes the fault support of MSO_3 and MSO_4. This indicates that it is not necessary to compute all the MSO sets to obtain a parsimonious set of residual generators guaranteeing maximal diagnosability.

The above remark tells us that we need to change the perspective and focus on the possible fault supports of MSO sets rather than the MSO sets themselves. Hence the following definitions from [5] focusing on fault supports.

▶ **Definition 15** (Test Support (TS)). *Given a model Σ and a set of faults $\mathcal{F}$, a subset of faults $\zeta \subseteq \mathcal{F}$ is a TS if there exists a PSO set $\Sigma^{'} \subseteq \Sigma$ such that $F_{\Sigma'} = \zeta$.*

▶ **Definition 16** (Minimal Test Support (MTS)). *Given a model Σ, a test support is an MTS if no proper subset is a test support.*

With the two above definition, it is now necessary to identify the subset of equations that correspond to TSs and MTSs.

▶ **Definition 17** (Test Equation Support (TES)). *A PSO set Σ is a TES if (1) $F_\Sigma \neq \emptyset$ and (2) for any PSO set $\Sigma^{'}$ such that $\Sigma^{'} \supsetneq \Sigma$, it holds that $F_{\Sigma'} \supsetneq F_\Sigma$.*

▶ **Definition 18** (Minimal Test Equation Support (MTES)). *A TES Σ is an MTES if there is no subset of Σ that is also a TES.*

The idea of MTES is to identify the PSOs with maximal number of equations that have a given test support [5]. In other words, for a given test support ζ, it is important to identify the maximal set of equations $\Sigma^{'} = \Sigma^{'+}$ such that $F_{\Sigma'} = \zeta$, thus avoiding that two MTES have the same TS.

It is also important to pay attention to the minimality property, i.e., to identify MTES instead of simply TES. Indeed minimality avoids obtaining redundant residual generators. This can be proved by showing that, considering the set of MTES for a given system, an ARR generated from a given MTES is never redundant with respect to the set of ARRs generated from the other MTESs. This statement holds true based on the following proposition, which is proved in [3], specifically due to the second condition.

▶ **Proposition 19.** *A given ARR_i is redundant with respect to a set of ARR_s, $i \in I$, $j \notin I$, if and only if there exists $I' \subseteq I$ such that:*

1. *$\forall z(observations)$, if all ARR_{i_s}, $i \in I'$, are satisfied by z, then ARR_j is satisfied by z.*
2. *The fault support of ARR_j contains the fault support of each ARR_i, $i \in I'$:*

$$\bigcup_{i \in I'} F_{ARR_i} \subseteq F_{ARR_j}.$$

In summary, the fault supports of the MTESs are sufficient to guarantee maximal diagnosability because they correspond to all the possible MTS, i.e. to all the fault supports of possible MSOs. Every MTES has a different MTS and no MTES is redundant.

Interestingly, the set of MTES is much smaller than the set of MSO sets, which results in computation time saving.

4.2 MTES based diagnosis process

The following steps, applied on system Σ_3 illustrate the process of diagnosis based on MTES.

Step 1: Specification

The first step involves defining the isolability requirements for the faults in the system. The pair of faults that need to be isolated are specified, for example:

- I_1: isolate f_1 from f_2.
- I_2: isolate f_1 from f_3.
- I_3: isolate f_2 from f_3.

Step 2: Compute the set of MTES

MTESs are computed for the system Σ_3:

- $MTES_1 = \{e_2, e_3, e_4, e_5, e_6\}$ with corresponding MTS $\{f_2, f_3\}$ and $\varphi(MTES_1) = 2$.
- $MTES_2 = \{e_1, e_3, e_4, e_5, e_6\}$ with corresponding MTS $\{f_1, f_3\}$ and $\varphi(MTES_1) = 2$.
- $MTES_3 = \{e_1, e_2, e_3, e_4, e_5\}$ with corresponding MTS $\{f_1, f_2\}$ and $\varphi(MTES_1) = 2$.

Step 3: Find the subset of MTES that satisfies all the isolability specifications

In this step, we check if there exists a Minimal Hitting Set that satisfies the isolability conditions defined in Step 1. We examine whether each specification can be satisfied by finding a set of equations where the faults can be isolated according to the conditions.

- For I_1 (isolate f_1 from f_2), an MTES whose corresponding MTS includes f_1 and not f_2 is required. It could be $MTES_2$.

- For I_2 (isolate f_1 from f_3): an MTES whose corresponding MTS includes f_1 and not f_3 is required. It could be $MTES_3$.
- For I_3 (isolate f_2 from f_3): an MTES whose corresponding MTS includes f_2 and not f_3 is required. $MTES_3$ also works.

Step 4: Find an FMSO within each selected MTES

- $MTES_2$ includes only $FMSO_4$ that can be used as residual generator.
- $MTES_3$ includes $FMSO_6$ and $FMSO_8$, so any of those could be used as residual generator.

4.3 Algorithm

An algorithm for computing the set of MTES is presented in [5]. The concept behind this algorithm is similar to the one used to find MSO sets. In each step, an equation is removed from the model, and the overdetermined part of the remaining set is recalculated. The main difference is that instead of removing all equations successively (or equivalence classes in the improved version), only equations associated with faults are removed. By doing so, the fault support of the resulting set decreases at each iteration, aligning with our goal of finding minimal fault supports.

The algorithm begins with a TES (the largest PSO set, ensuring no larger PSO set has the same fault support), and remains within the class of TES by progressively removing the fault-carrying equations. Thus, as these equations are removed, it is ensured that the resulting set continues to satisfy the properties of a TES.

The stopping condition for this algorithm is as follows: a TES Σ is considered an MTES if all fault-carrying equations in Σ belong to the same equivalence class. Indeed, when this condition is met, if the process goes on by removing a fault-carrying equation and recalculating the overdetermined part, the resulting equation set will no longer contain any fault-carrying equations. This stopping condition is tested at each iteration, and when it is satisfied, the TES Σ is confirmed as an MTES.

Another stopping condition is given in [5]. A TES $\Sigma^{'} \subseteq \Sigma$ is an MTES if and only if $\varphi(\Sigma^{'}) = \varphi((\Sigma \setminus \Sigma_f)^+) + 1$, with Σ_f the set of equations in Σ affected by faults. $(\Sigma \setminus \Sigma_f)^+$ represents the equations that can be added without modifying the fault support, while remaining within the class of PSO sets. If we take any TES, it can be extended with the equations from $(\Sigma \setminus \Sigma_f)^+$. The consequence of this condition is that all MTES have the same structural redundancy (for Σ_3, the 3 MTES have structural redundancy equal to 2). Thus, either all MTES are MSO sets, i.e., their structural redundancy is 1, or none of them is exactly an MSO set because their structural redundancy is more than 1, in which case MTES obviously include several MSO sets. If all the equations of a system are affected by faults, then MTES are exactly MSO sets. The advantage of using this condition is that it avoids performing a Dulmage-Mendelsohn decomposition at each step.

If n_f is the number of fault-carrying equations, the number of nodes traversed by the search algorithm is bounded by Equation (3).

$$\sum_{k=0}^{\varphi-1} \binom{n_f}{k}. \tag{3}$$

Running example: We now apply the algorithm step by step on the illustrative example Σ_4 presented in Table 7.

Table 7 Illustrative example Σ_3.

	x_1	x_2	x_3	x_4	x_5	
e_1		×	×	×		f_1
e_2	×	×				
e_3		×				f_2
e_4			×			f_3
e_5				×	×	
e_6	×					
e_7		×				
e_8					×	
e_9			×			

First, we consider the entire set. It is a PSO set, so there is no need to compute the overdetermined part. The equivalence classes are: $\{e_1, e_5, e_8\}, \{e_2, e_6\}, \{e_3\}, \{e_4\}, \{e_7\}, \{e_9\}$. The stopping condition is not verified because all fault-carrying equations are not in the same class. Thus, $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$ is only a TES, but not a MTES. Then, an equivalence class containing a fault (f_1) is removed. The set $\{e_1, e_5, e_8\}$ is eliminated (equivalent to removing e_1 and then computing the overdetermined part). The remaining set is $\{e_2, e_3, e_4, e_6, e_7, e_9\}$. The equivalence classes are as follow: $\{e_2, e_6\}, \{e_3\}, \{e_7\}, \{e_4, e_9\}$. Since the stopping condition is still not met, we proceed by removing the equivalence class containing f_2. The resulting set is $\{e_2, e_4, e_6, e_7, e_9\}$. This time, the stopping condition is satisfied (only one fault remaining), $MTES_1$ is identified. The same process is applied to find all the MTES.

5 Summary

To sum up the previous sections, MSO sets are sets of equations in a model where the number of equations exceeds the number of unknowns by one. This overdetermination is minimal, meaning that if any equation was removed, the set would no longer be overdetermined.

MSO sets are particularly useful in fault detection as they provide the basis for producing residual generators. They retain the necessary and sufficient redundancy. However, as seen in the previous sections, relying on the generation of MSO sets to generate the residual generators that guarantee maximal diagnosability has some drawbacks. The first point is that some MSO sets may not be sensitive to faults, making those irrelevant for fault detection and diagnosis. Second, even if they can be generated systematically, finding all MSO sets can be computationally expensive for large systems. Last but not least, they do not provide any information about fault isolation.

FMSO sets are MSO sets but they only retain MSO sets with non-empty fault support, so they are all relevant for fault detection and isolation. They are less numerous than MSO sets, which reduces significantly computation time. However, as shown in Section 4.1, several FMSO sets may have the same fault support and some may be redundant.

By focusing on fault supports rather than specific sets of equations, TESs effectively identify exactly those fault supports that correspond to all the possible MSO sets. MTESs gather all the equations from which all the FMSO sets with the corresponding fault support can be generated. As a result, MTESs inherently address the problem of test selection. In addition, generating MTES is obviously less computationally demanding than generating MSO sets or FMSO sets. Note however that, once MTESs are obtained, one FMSO set must be extracted for each of them to achieve maximal diagnosability.

One should note that for some problem, in particular test selection for distributed systems, having all the MSO sets at hand is an advantage because subsystem interconnections express in terms of equations so equation subsets are as important as their fault supports [1, 7].

Table 8 gives a summary of the comparison of MSO sets, FMSO sets, and MTES.

Table 8 Summary comparison of MSO sets, FMSO sets, and MTES.

Aspect	MSO sets	FMSO sets	MTES
Diagnosis target	residual generation	fault driven residual generation	isolability driven residual generation
Scalability	too complex for large systems	too complex for large systems	good scalability

6 Application of the concepts on a real example: the ADCS

In this section, all the concepts discussed earlier are explored through a realistic example: the Attitude Determination and Control System (ADCS) of a Low Earth Orbit satellite [2].

6.1 Structural model of the ADCS

In this example, the dynamics of a satellite's motion are modeled using equations that describe the behavior of torque and angular velocity. The total torque acting on the satellite, T, is the sum of the torques generated by the magnetorquers, reaction wheels, and external disturbances. The rotational dynamics are influenced by the satellite's moment of inertia I and angular velocity ω.

The global system is divided into 2 main parts: the Attitude Determination System (ADS) and Attitude Control System (ACS). The ADS uses a combination of gyroscopes and vector sensors (such as sun and star sensors) to estimate the satellite's attitude state, including pitch, roll, and yaw angles, along with their corresponding rates. The resulting state vector for the satellite is $X = [\psi, \theta, \phi, \dot{\psi}, \dot{\theta}, \dot{\phi}]$. These estimates are then fused together to provide feedback to the ACS, which controls actuators like reaction wheels and magnetorquers to adjust the satellite's attitude.

The structural model of the satellite's ADCS is enhanced by incorporating fault information directly into the system's equations, with faults modeled as signals. The model considers faults affecting the rate and vector sensors of the ADS (respectively *frs* and *fvs*), as well as the reaction wheels of the ACS (*frw*), including various types such as hard, soft, and intermittent faults. Each fault can affect all three axes of the system (x, y, and z).

The ADCS's structural model includes 42 equations including 42 unknowns variables, 15 known variables, and 9 faults that appear as parameters in the equations. The majority of these equations are based on three behavioral equations that address the system dynamics along the x, y, and z axes. The biadjacency matrix of the system is given by Figure 3. The links between relations and unknown variables are represented in blue, those with faults in red and those with known variables in black.

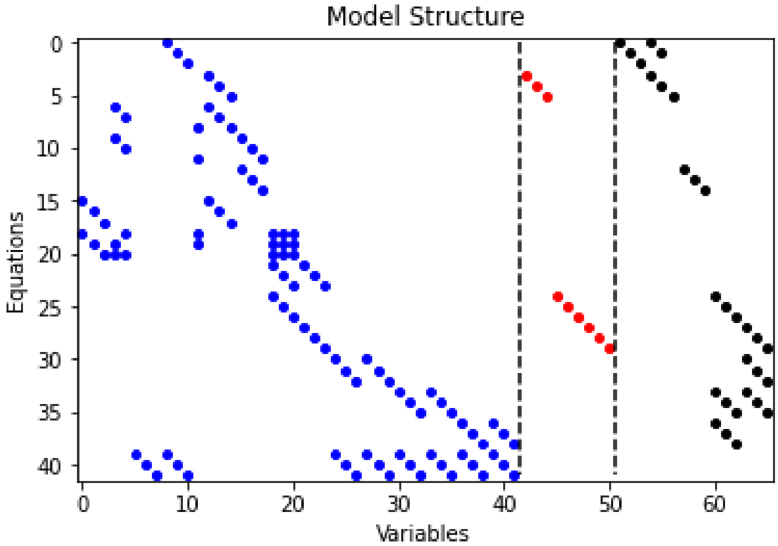

Figure 3 Biadjacency matrix of the ADCS system.

6.2 Results

Using the "Fault Diagnosis Toolbox"[2], 2448 MSO sets are found for this model, but only 9 MTES. There are also 2448 FMSO sets, as each MSO set has a non empty fault support.

In our example, we have $n_e = 42$, $n_f = 9$, and $\varphi = 9$. Applying Equation 3 results in a maximum of 511 nodes traversed during the search for MTES, whereas applying Equation 2 gives a maximum number of nodes exceeding 10^9. These values clearly illustrate the difference in complexity between the two algorithms, with the MSO algorithm requiring the traversal of far more nodes. However, these are only upper bounds, as in practice, the equations can be grouped into equivalence classes, significantly reducing the actual number of nodes traversed by the algorithm. Table 9 presents the computation time required to compute both the MSO sets and the MTES. It shows that the computation time for MSO sets is significantly higher than for MTES in large systems.

Table 9 Computation time of both algorithms.

Algorithm	Calculation time
MSO algorithm	0.534 s
MTES algorithm	0.239 s

It is clear that the 2448 MSO sets for this model are not all useful for the generation of residuals. MSO sets with empty fault supports are not interesting, as they do not contribute to fault driven residual generation. This is why we focus on FMSO sets, which contain faults and are thus more relevant for diagnosis. However, in this case, all MSO sets contain faults, making the use of FMSO sets less critical. There must still be some MSO sets with identical fault supports or non-minimal ones. The MTES of this model, along with the associated fault supports, are given in Table 10.

[2] `https://faultdiagnosistoolbox.github.io/`

Table 10 MTES and their fault support for the ADCS example.

	MTES	Fault support
$MTES_1$	$\{e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{24}, e_{30}\}$	$\{fvs_z\}$
$MTES_2$	$\{e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{23}, e_{29}\}$	$\{fvs_y\}$
$MTES_3$	$\{e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{22}, e_{28}\}$	$\{fvs_x\}$
$MTES_4$	$\{e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{27}\}$	$\{frs_z\}$
$MTES_5$	$\{e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{26}\}$	$\{frs_y\}$
$MTES_6$	$\{e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}, e_{20}, e_{21}, e_{25}\}$	$\{frs_x\}$
$MTES_7$	$\{e_6, e_9, e_{12}, e_{15}\}$	$\{frw_z\}$
$MTES_8$	$\{e_5, e_8, e_{11}, e_{14}\}$	$\{frw_y\}$
$MTES_9$	$\{e_4, e_7, e_{10}, e_{13}\}$	$\{frw_x\}$

For this example, all these MTES are MSO sets. These are the MSO sets with the minimal fault supports, containing the largest number of equations. The other MSO sets are not necessary as they have fault supports that are unions of these, or they can be expanded with other equations without altering the fault support while still remaining MSO sets. These 9 MTES are necessary and sufficient to achieve maximal isolability as their fault support only includes one fault.

7 Conclusion

In this paper, a thorough examination and comparison of key concepts in model-based fault diagnosis is provided, specifically focusing on the structural analysis concepts named the Minimally Structurally Overdetermined (MSO) sets, Fault driven Minimally Structurally Overdetermined (FMSO) sets, and Minimal Test Equation Supports (MTES).

This article conducts a detailed analysis of the rules, applications and interrelationships of MSO sets, FMSO sets and MTESs, providing insights into their respective strengths and limitations. It then systematically examines the algorithms used to identify MSO and MTES sets, discussing their computational efficiency and impact on diagnosis capabilities. The final part of this work applies these concepts to a practical example, demonstrating their effectiveness in a real-world scenario. More specifically, the algorithms are applied to the Attitude Determination and Control System (ADCS) of a satellite in low Earth orbit, and the structural model, algorithmic complexity involved and the possibility of diagnosing faults are studied. This results highlight the practical usefulness of MTES in reducing the computational load while maintaining robust fault isolation capabilities.

References

1 Elodie Chanthery, Anna Sztyber, Louise Travé-Massuyès, and Carlos Gustavo Pérez-Zuñiga. Process decomposition and test selection for distributed fault diagnosis. In *Trends in Artificial Intelligence Theory and Applications. Artificial Intelligence Practices: 33rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2020, Kitakyushu, Japan, September 22-25, 2020, Proceedings 33*, pages 914–925. Springer, 2020. doi:10.1007/978-3-030-55789-8_78.

2 Elodie Chanthery, Louise Travé-Massuyès, and Saurabh Indra. Fault isolation on request based on decentralized residual generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 46(5):598–610, 2015. doi:10.1109/TSMC.2015.2479192.

3 Marie-Odile Cordier, Philippe Dague, François Lévy, Jacky Montmain, Marcel Staroswiecki, and Louise Travé-Massuyès. Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(5):2163–2177, 2004. doi:10.1109/TSMCB.2004.835010.

4 Andrew L Dulmage and Nathan S Mendelsohn. Coverings of bipartite graphs. *Canadian Journal of Mathematics*, 10:517–534, 1958.

5 Mattias Krysander, Jan Åslund, and Erik Frisk. A structural algorithm for finding testable sub-models and multiple fault isolability analysis. In *21st International Workshop on Principles of Diagnosis (DX-10), Portland, Oregon, USA*, pages 17–18, 2010.

6 Mattias Krysander, Jan Åslund, and Mattias Nyberg. An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(1):197–206, 2007. doi:10.1109/TSMCA.2007.909555.

7 Carlos Gustavo Pérez, Louise Travé-Massuyès, Elodie Chanthery, and Javier Sotomayor. Decentralized diagnosis in a spacecraft attitude determination and control system. *Journal of Physics: Conference Series*, 659(1):012054, 2015.

8 Carlos Pérez-Zuñiga, Elodie Chanthery, Louise Travé-Massuyes, Javier Sotomayor, and Christian Artigues. Decentralized diagnosis via structural analysis and integer programming. *IFAC-PapersOnLine*, 51(24):168–175, 2018.