

Realizability Constrained Selection of Residual Generators for Fault Diagnosis with an Automotive Engine Application

Carl Svärd, Mattias Nyberg and Erik Frisk

Linköping University Post Print



N.B.: When citing this work, cite the original article.

©2013 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

Carl Svärd, Mattias Nyberg and Erik Frisk, Realizability Constrained Selection of Residual Generators for Fault Diagnosis with an Automotive Engine Application, 2013, IEEE Transactions on Systems, Man and Cybernetics: Systems, (43), 6, 1354-1369.

<http://dx.doi.org/10.1109/TSMC.2013.2258906>

Postprint available at: Linköping University Electronic Press

<http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-77191>

Realizability Constrained Selection of Residual Generators for Fault Diagnosis with an Automotive Engine Application

Carl Svård, Mattias Nyberg, and Erik Frisk

Abstract—This paper considers the problem of selecting a set of residual generators for inclusion in a model-based diagnosis system, while fulfilling fault isolability requirements and minimizing the number of residual generators. Two novel algorithms for solving the selection problem are proposed. The first algorithm provides an exact solution fulfilling both requirements and is suitable for small problems. The second algorithm, which constitutes the main contribution, is suitable for large problems and provides an approximate solution by means of a greedy heuristic and by relaxing the minimal cardinality requirement. The foundation for the algorithms is a novel formulation of the selection problem which enables an efficient reduction of the search-space by taking into account realizability properties, with respect to the considered residual generation method. Both algorithms are general in the sense that they are aimed at supporting any computerized residual generation method. In a case study the greedy selection algorithm is successfully applied in an industrial sized automotive engine system.

Index Terms—model-based fault diagnosis, residual generation

I. INTRODUCTION

MODEL-based Fault Detection and Isolation (FDI) systems typically contain the three sub-systems: residual generation, residual evaluation, and fault isolation, see e.g., [1]. In this work, as in for example [2], [3], [4], [5], design of the residual generation sub-system is considered to be a two-step approach. In the first step, a large set of candidate residual generators are found. In general, it may be possible to find a huge number of candidate residual generators for large models and due to implementation aspects, such as complexity and computational load, it is infeasible, or even impossible, to use all these in the final FDI-system. In addition, it is often possible to meet stated requirements with a, possibly small, subset of all residual generators. Therefore, in the second step, the set of candidate residual generators most suitable to be included in the FDI-system are selected. The topic of this paper is the selection problem in the second step.

The selection problem is formulated by considering two different requirements on the final set of residual generators. First, it is required that the set of residual generators fulfill an isolability requirement stating which faults that should be isolated from each other. Motivated by the implementation

aspects mentioned above, a set of residual generators of low cardinality is preferred over a set of high cardinality, given that the two sets have equal isolability properties. Therefore, secondly, it is required that the set of residual generators is of minimal cardinality.

Two novel algorithms for solving the selection problem are proposed in this paper. The first algorithm provides an exact solution fulfilling both the isolability and the minimal cardinality requirements and is suitable for small problems. The second algorithm, which is the main contribution, relaxes the minimal cardinality requirement and provides an approximate solution by means of a greedy heuristic. This algorithm is suitable for large real-world problems for which the first algorithm is intractable. Both algorithms are general in the sense that they are aimed at supporting any computerized residual generation method.

In general, all the candidate residual generators found in the first step of the design process are not realizable, i.e., it is not possible to implement residual generators for all found candidates. Typically, evaluation of realizability is a computationally demanding task and in those cases where the number of candidate residual generators found is large, it may not be feasible to first evaluate the realizability of all found candidate residual generators and then make the selection. To handle this, the proposed algorithms exploit a novel formulation of the selection problem which takes the realizability aspect into account. This, in addition, enables an efficient reduction of the search-space, which typically is quite large for practical problems. The problem is stated as an optimization problem where isolability and realizability properties are stated in terms of attributes of subsets of the model equations.

In Section II, a motivating industrial application example is presented. Section III presents preliminaries regarding realizability and fault isolability, given a residual generation method. The residual generator selection problem is formalized in Section IV. The first selection algorithm is presented and discussed in Section V. The second, greedy, algorithm is presented and justified in Section VI. Section VII briefly describes the residual generation method [5] which is used in the application example. In Section VIII the greedy selection algorithm is used to solve the industrial application problem described in Section II. The paper is concluded in Section X.

II. MOTIVATING APPLICATION EXAMPLE

As a motivating industrial application example, consider the problem of selecting a set of suitable residual generators

C. Svård and M. Nyberg are affiliated with Scania CV AB, SE-15187 Södertälje, Sweden (e-mail: {carl.svard, mattias.nyberg}@scania.com)

E. Frisk is affiliated with the Department of Electrical Engineering, Linköping University, SE-58183 Linköping, Sweden (e-mail: frisk@isy.liu.se)

Manuscript received Month Date, Year; revised Month Date, Year.

for detecting and isolating faults in an automotive engine system. The studied engine is a 13-L six-cylinder Scania truck diesel engine equipped with Exhaust Gas Recirculation (EGR), Variable Geometry Turbine (VGT), and intake throttle.

There are in total 12 faults that should be detected and isolated from each other in this system. An overview of the system with the considered faults, is shown in Figure 1. More details regarding the system, and the faults, are given in Section VIII.

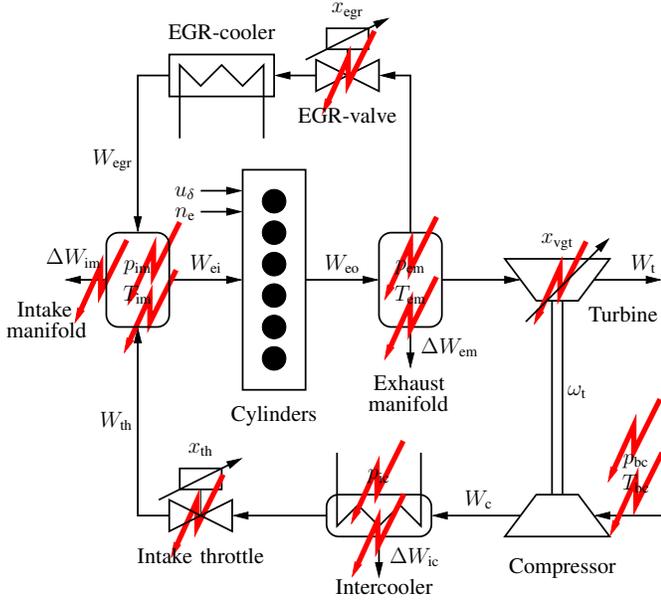


Fig. 1. Overview of the automotive engine System. Considered faults marked with red arrows.

For the model of this system, and for the specific residual generation method developed in [5], which is briefly described in Section VII, it is possible to find in total 14,242 candidate residual generators. Indeed, as argued in Section I, it is not possible to include all these residual generators in the FDI-system.

In order to isolate a certain fault from another, it is necessary to find a residual generator sensitive to the fault but not to the other. Intuitively, a set of approximately 12 residual generators would be sufficient in order to isolate the 12 considered faults from each other since single faults are considered and each residual isolates faults *from* one fault at a time. Thus, a set of approximately 12 residual generators, capable of isolating the 12 faults, should be selected from the set of 14,242 candidate residual generators which means that the search-space is quite large.

The fault sensitivity matrix for a small subset of the candidate residual generators found, with respect to the 12 considered faults, are shown in Figure 2. A square in position (i,j) in Figure 2 indicates that the residual generator corresponding to column j is sensitive, i.e., responds, to the fault corresponding to row i. According to the figure, most of the residual generators are sensitive to most of the faults and it is therefore not straightforward to perform the selection. In addition, as said in Section I, the sought set of residual generators should be realizable and preferably of minimal cardinality. Due to

the large number of candidate residual generators, it is not possible to perform a complete search to find the set of residual generators. This makes the selection problem non-trivial. In Section VIII this selection problem will be reconsidered and solved.

III. PRELIMINARIES

The purpose of this section is to formally introduce the notions of realizability and isolability, given a residual generation method, and ultimately derive necessary and sufficient conditions for fault isolability in terms of properties of sets of equations.

A model M is represented by a tuple $M = (E, X, Z, F)$, containing an equation set E relating the unknown variables X , known variables Z , and fault variables F . Without loss of generality, the following is assumed regarding the model.

Assumption 1. Each fault $f \in F$ is contained in one, and only one, of the equations in the model M .

Note that if a fault $f \in F$ is contained in more than one equation, the fault f can be replaced with a new variable x_f in these equations, and the equation $x_f = f$ added to the equation set E . This added equation will then be the only equation where f occurs.

Given a model, a residual generator is formally defined as follows.

Definition 1 (Residual Generator). Let $M = (E, X, Z, F)$ be a model. A system R with input Z and output r is a residual generator for M , and r is a residual, if $f = 0$ implies $r = 0$ for all $f \in F$.

An important property of a residual generator is whether or not it responds to a certain fault.

Definition 2 (Fault Sensitivity). Let R be a residual generator for the model M . Then R is sensitive to fault $f \in F$ if $f \neq 0$ implies $r \neq 0$.

Note that in practice, residuals typically deviate from zero even in the case when all faults are zero due to for example unknown initial conditions, changes in operating conditions, and uncertainties such as modeling errors and noise. Therefore, residuals are often thresholded as a part of the residual evaluation mentioned in Section I, where the aim is to detect changes in the residual behavior caused by faults.

The notions of residual generators and fault sensitivity can be made more precise and formal, see for example [1], [6], [7], and references therein. This is however not necessary in the context of this work for which the above definitions are sufficient.

A. Realizability

The method used for design of residual generators plays a central role in this work. A residual generation method is formally defined as follows.

Definition 3 (Residual Generation Method \mathcal{M}). Let $M = (E, X, Z, F)$ be a model. A residual generation method, \mathcal{M} , is a procedure, denoted $\mathcal{M}(\cdot)$, taking as input a set of equations

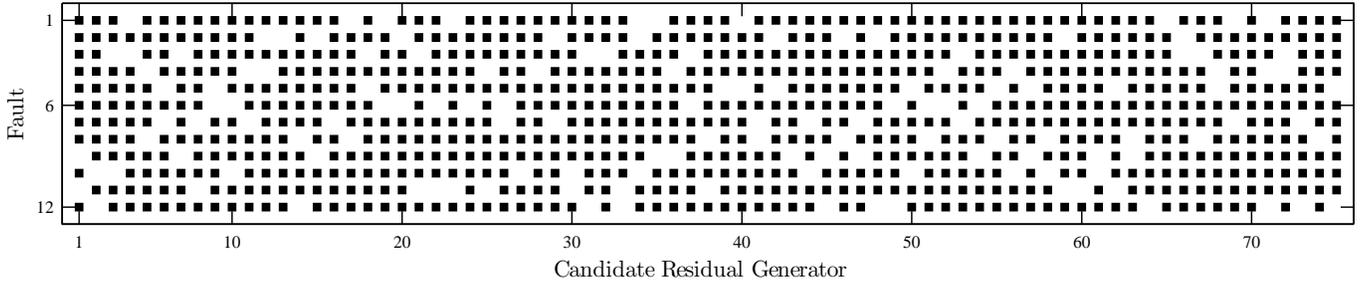


Fig. 2. Fault sensitivity for a small subset of the 14,242 candidate residual generators found for the automotive engine system. A square in position (i, j) denotes that the residual generator corresponding to column j is sensitive to the fault corresponding to row i .

$S \subseteq E$ and giving as output a residual generator R for M , or an empty set \emptyset .

Given a residual generation method and an equation set, an important issue is whether the output from the method is non-empty, or not. That is, if a residual generator can be created with the method given the equation set as input. This property of an equation set, with respect to a method, is formalized below.

Definition 4 (Realizability with method \mathcal{M}). *Let S be an equation set and \mathcal{M} a residual generation method. Then S is realizable with \mathcal{M} if $\mathcal{M}(S) \neq \emptyset$.*

For an example, consider a model containing the following set of differential and algebraic equations

$$\begin{aligned} e_1 : \quad \dot{x}_1 &= -x_1 + u + f_1 \\ e_2 : \quad y_1 &= x_1 + f_2 \\ e_3 : \quad y_2 &= x_1 + f_3, \end{aligned} \quad (1)$$

where x_1 is an unknown variable, $\{u, y_1, y_2\}$ known variables, and $\{f_1, f_2, f_3\}$ fault variables. Let \mathcal{M}' be a residual generation method capable of handling linear, static, equation sets. It can then be concluded that the equation set $\{e_2, e_3\}$ is realizable with \mathcal{M}' , but not for instance the equation set $\{e_1, e_2\}$ since e_1 is a differential equation.

Let e_f denote the equation in an equation set containing fault f . From now on, the following is assumed regarding a residual generation method.

Assumption 2. *Let S be an equation set and \mathcal{M} a residual generation method. Further, let S be realizable with \mathcal{M} and $R = \mathcal{M}(S)$ the corresponding residual generator. Then, R is sensitive to fault f if and only if $e_f \in S$.*

The important implication of Assumption 2, in the context of this work, is that a residual generation method preserves structural fault information, in the sense that it does not discard, nor add, equations containing faults, during the realization process.

For an example, consider again the model (1) and assume that the equation set $\{e_1, e_2, e_3\}$ is realizable with a method \mathcal{M} . Assumption 2 guarantees that a residual generator obtained from $\mathcal{M}(\{e_1, e_2, e_3\})$ is sensitive to the faults f_1 , f_2 , and f_3 . Thus, the output from \mathcal{M} can neither be the residual generator $r = y_1 - y_2$ since this residual generator is not sensitive to f_1 , nor the trivial residual generator $r = 0$.

Assumption 2 enables usage of *structural analysis*, see, e.g., [1], in which only the information that a certain equation contains a certain fault, or variable, is taken into account. In a structural approach, analytical properties of the equations, uncertainties in the model, and measurement noise are not considered in the analysis. With this in mind, structural analysis has still proven to be a powerful tool that enables efficient analysis of large scale systems in early stages of the design process. For a detailed description of this approach, as well as a discussion of its pros and cons, see for instance [1].

B. Fault Isolability

In this section fault isolability is formally defined from two different perspectives. First, fault isolability is defined as a property of a given set of residual generators. Second, fault isolability is defined as a property of a model given a method for residual generation. The main motivation for introducing both definitions is to prove soundness and completeness of the selection algorithms in Sections V and VI. More specific, that the algorithms find a set of residual generators fulfilling the stated isolability requirement if, and only if, the corresponding faults are isolable in the model with the considered residual generation method.

Given a set of residual generators, fault isolability is defined as follows.

Definition 5 (Fault Isolability with residual generators \mathcal{R}). *Let $M = (E, X, Z, F)$ be a model and \mathcal{R} a given set of residual generators for M . A fault $f_i \in F$ is isolable from fault $f_j \in F$ with \mathcal{R} if there exists a residual generator $R \in \mathcal{R}$ that is sensitive to f_i but not to f_j .*

Note that Definition 5 is not dependent on the residual generation method. Next, fault isolability is defined as a property of a model, given a residual generation method.

Definition 6 (Fault Isolability with method \mathcal{M}). *Let $M = (E, X, Z, F)$ be a model and \mathcal{M} a residual generation method. A fault $f_i \in F$ is isolable from fault $f_j \in F$ in M with \mathcal{M} if a residual generator R for M can be created with \mathcal{M} such that R is sensitive to f_i but not to f_j .*

Note that if $S \subseteq E$ and fault $f_i \in F$ is isolable from fault $f_j \in F$ with the residual generator $R = \mathcal{M}(S)$ then, by Definition 6, f_i is isolable from f_j in the model M with the

method \mathcal{M} . The converse is also true. For future reference, this trivial result is stated below.

Proposition 1. *Let $M = (E, X, Z, F)$ be a model and \mathcal{M} a residual generation method. Then, fault $f_i \in F$ is isolable from fault $f_j \in F$ in M with \mathcal{M} if and only if there exists $S \subseteq E$ such that f_i is isolable from f_j with $R = \mathcal{M}(S)$.*

By exploiting the notion of realizability and Assumption 2, necessary and sufficient conditions for fault isolability, given a model and a residual generation method, in terms of properties of subsets of the model equations can be established.

Proposition 2. *Let $M = (E, X, Z, F)$ be a model and \mathcal{M} a residual generation method. Then, for each $S \subseteq E$ it holds that fault $f_i \in F$ is isolable from fault $f_j \in F$ with $R = \mathcal{M}(S)$ if and only if S is realizable with \mathcal{M} , $e_{f_i} \in S$, and $e_{f_j} \notin S$.*

Proof: Assume first that f_i is isolable from f_j with $R = \mathcal{M}(S)$. By assumption, R is a residual generator and therefore $\mathcal{M}(S) \neq \emptyset$ and it follows that S is realizable with \mathcal{M} from Definition 4. Further, by Definition 5, R is sensitive to f_i but not to f_j . Assumption 2 then implies that $e_{f_i} \in S$ and $e_{f_j} \notin S$, and the first part of the proof is complete. For the converse, assume that S that is realizable with \mathcal{M} , i.e., $\mathcal{M}(S) \neq \emptyset$, $e_{f_i} \in S$, and $e_{f_j} \notin S$. Since $S \subseteq E$ and $\mathcal{M}(S) \neq \emptyset$, it follows from Definition 3, that $R = \mathcal{M}(S)$ is a residual generator for M . Assumption 2 then states that R is sensitive to f_i but not to f_j . Definition 5 completes the proof. ■

Consider again the model in (1) and the linear, static, residual generation method \mathcal{M}' with which the equation set $\{e_2, e_3\}$ is realizable. Due to this fact and since $e_{f_2} = e_2 \in \{e_2, e_3\}$, $e_{f_3} = e_3 \in \{e_2, e_3\}$, and $e_{f_1} = e_1 \notin \{e_2, e_3\}$, it can be deduced from Proposition 2 that faults f_2 and f_3 are both isolable from fault f_1 with the residual generator $R' = \mathcal{M}'(\{e_2, e_3\})$.

Note that even though additive faults were considered in this example above, the framework in this paper is general and independent on the fault model, i.e., also multiplicative faults are allowed.

IV. THE RESIDUAL GENERATOR SELECTION PROBLEM

In this section, the residual generator selection problem is formalized and stated as an optimization problem: fulfill an isolability requirement while minimizing the number of residual generators. This formulation exploits the notion of realizability introduced in the previous section and enables an efficient reduction of the search-space.

As input to the residual generator selection procedure the following are assumed to be given: a model $M = (E, X, Z, F)$, a method for residual generation \mathcal{M} , and an isolability requirement \mathcal{F} . The output from the selection procedure is a set of residual generators, \mathcal{R} . As said in Section I, two different requirements on \mathcal{R} are considered:

- 1) \mathcal{R} should fulfill the isolability requirement \mathcal{F} , and
- 2) \mathcal{R} should be of minimal cardinality.

A. The Isolability Requirement

The *isolability requirement*, \mathcal{F} , is defined as a set of ordered fault pairs $(f_i, f_j) \in F \times F$, where the interpretation of (f_i, f_j)

is that f_i should be isolable from f_j . Consequently, \mathcal{F} is *fulfilled* with \mathcal{R} if for each $(f_i, f_j) \in \mathcal{F}$ it holds that f_i is isolable from f_j with \mathcal{R} .

From Proposition 2 it can be deduced that to fulfill the isolability requirement it is necessary, and sufficient, to find for each fault pair $(f_i, f_j) \in \mathcal{F}$ an equation set $S_{f_i f_j} \subseteq E$ such that $S_{f_i f_j}$ is realizable with \mathcal{M} , and for which $e_{f_i} \in S_{f_i f_j}$ and $e_{f_j} \notin S_{f_i f_j}$. Given the equation subsets $S_{f_i f_j}$, a set of residual generators fulfilling \mathcal{F} can be constructed as

$$\mathcal{R} = \{\mathcal{M}(S_{f_i f_j}) : \forall (f_i, f_j) \in \mathcal{F}\}. \quad (2)$$

B. Candidate Equation Set

If E is a small set, it may be tractable to evaluate all subsets of E in the search for the sets $S_{f_i f_j}$ in (2) but in the general case, however, it is not. In order to reduce the search-space, all subsets of E that not by necessity are realizable are discarded. To this end, the notions of *necessary realizability criterion* and *candidate equation set* are introduced.

Definition 7 (Necessary Realizability Criterion for method \mathcal{M}). *Let S be an equation set and \mathcal{M} a residual generation method. A constraint on S is a necessary realizability criterion for \mathcal{M} if the constraint is satisfied when S is realizable with \mathcal{M} .*

Definition 8 (Candidate Equation Set for method \mathcal{M}). *Let S be an equation set and \mathcal{M} a residual generation method for which a necessary realizability criterion is defined. Then S is a candidate equation set for \mathcal{M} if S fulfills the necessary realizability criterion for \mathcal{M} .*

Regarding the choice of necessary realizability criterion for a given residual generation method, it is desirable that it fulfills at least two requirements. First of all, in order to be meaningful, the necessary realizability criterion should significantly reduce the search-space in terms of number of discarded non-realizable subsets of model equations. Secondly, in order to be of practical use, it should be possible to extract all candidate equation sets for a method, given a model, in an efficient way.

As an example, a candidate equation set for several observer-based residual generation methods is an equation set in, or that trivially can be cast in, state-space form, see e.g., [1], [7] and references therein. An additional example is given by the class of methods referred to as sequential residual generation, see e.g., [8], [9], [10], [1], [5], for which Minimal Structurally Over-determined (MSO) sets of equations [11], [12], [13], constitute candidate equation sets.

C. Formalization of the Selection Problem

Consider now the isolability requirement \mathcal{F} and let $\mathcal{S}_{\mathcal{M}} \subseteq 2^E$ be the set of all candidate equation sets for the residual generation method \mathcal{M} .

Define the *isolability class*, $I_{f_i f_j}$, of $\mathcal{S}_{\mathcal{M}}$ for the fault pair $(f_i, f_j) \in \mathcal{F}$ as the collection of all candidate equation sets in $\mathcal{S}_{\mathcal{M}}$ containing fault f_i but not fault f_j , that is,

$$I_{f_i f_j} = \{S \in \mathcal{S}_{\mathcal{M}} : e_{f_i} \in S \wedge e_{f_j} \notin S\}. \quad (3)$$

Let the set

$$\mathcal{J} = \{I_{f_i f_j} : \forall (f_i, f_j) \in \mathcal{F}\}, \quad (4)$$

contain the isolability classes of $\mathcal{S}_{\mathcal{M}}$ for all fault pairs in \mathcal{F} .

The next result formulates the problem of fulfilling the isolability requirement in terms of properties of the candidate equation sets.

Lemma 1. *Let $M = (E, X, Z, F)$ be a model, \mathcal{M} a residual generation method, and \mathcal{F} an isolability requirement. Also, let $\mathcal{S}_{\mathcal{M}}$ be the set of all candidate equation sets for \mathcal{M} and \mathcal{J} the set of all isolability classes of $\mathcal{S}_{\mathcal{M}}$ for \mathcal{F} , defined according to (3) and (4). Then, for each $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$ where all $S \in \mathcal{S}$ are realizable with \mathcal{M} it holds that \mathcal{F} is fulfilled with*

$$\mathcal{R} = \{\mathcal{M}(S) : \forall S \in \mathcal{S}\}, \quad (5)$$

if and only if

$$\forall I \in \mathcal{J}, \quad \mathcal{S} \cap I \neq \emptyset. \quad (6)$$

Proof: Assume first that \mathcal{F} is fulfilled with \mathcal{R} defined according to (5). First note that this implies that for each $(f_i, f_j) \in \mathcal{F}$ there exists a residual generator $R \in \mathcal{R}$ such that f_i is isolable from f_j with R . This, Proposition 2, and (5), imply that for each $(f_i, f_j) \in \mathcal{F}$ there exists a $S \in \mathcal{S}$ such that $R = \mathcal{M}(S) \in \mathcal{R}$, $e_{f_i} \in S$, and $e_{f_j} \notin S$. This implies, since $S \in \mathcal{S}$ and $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$, that $S \cap I_{f_i f_j} \neq \emptyset$ where $I_{f_i f_j}$ is defined according to (3). Hence, for each $(f_i, f_j) \in \mathcal{F}$ there exists $S \in \mathcal{S}$ such that $S \cap I_{f_i f_j} \neq \emptyset$. Since (4) holds, this implies that (6) is satisfied and the first part of the proof is complete. For the converse, assume that (6) is satisfied. This, (3) and (4) implies that for each $(f_i, f_j) \in \mathcal{F}$ there exists $S \in \mathcal{S}$ such that $e_{f_i} \in S$ and $e_{f_j} \notin S$. This and the fact that all $S \in \mathcal{S}$ are realizable with \mathcal{M} , implies via Proposition 2 that for each $(f_i, f_j) \in \mathcal{F}$ there exists $S \in \mathcal{S}$ such that f_i is isolable from f_j with $R = \mathcal{M}(S)$. Thus, if $\mathcal{R} = \{\mathcal{M}(S) : \forall S \in \mathcal{S}\}$ there exists $R \in \mathcal{R}$ such that f_i is isolable from f_j with R for each $(f_i, f_j) \in \mathcal{F}$ and the proof is complete. ■

For the set of residual generators \mathcal{R} to fulfill also the stated minimal cardinality requirement, the cardinality of the set \mathcal{S} in Lemma 1 should be minimized. Thus, the residual generator selection problem can be stated as the problem of finding the smallest set within $\mathcal{S}_{\mathcal{M}}$ which satisfies (6). To conclude, the selection problem is stated as the minimization problem

$$\min_{\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}} |\mathcal{S}| \quad (7a)$$

$$\text{s.t. } \forall S \in \mathcal{S}, \quad \mathcal{M}(S) \neq \emptyset \quad (7b)$$

$$\forall I \in \mathcal{J}, \quad \mathcal{S} \cap I \neq \emptyset, \quad (7c)$$

where $|\cdot|$ is the cardinality of a set.

V. MINIMAL HITTING SET BASED SELECTION

A *hitting set* is a set that has a non-empty intersection with every set in a collection of sets. In fact, the isolability requirement, given by (7c), on the set of candidate equation sets \mathcal{S} implies that \mathcal{S} should be a hitting set for the collection of sets \mathcal{J} . Further, to also fulfill the minimal cardinality requirement (7a), \mathcal{S} should be a hitting set for \mathcal{J} of minimal cardinality, i.e., a so called *minimal cardinality hitting set*. By necessity, a minimal cardinality hitting set is a *minimal hitting set*, i.e., a hitting set of which no proper subset is a hitting set.

This fact suggests the following naive, but nevertheless simple, approach for solving the selection problem (7). First find the collection of all minimal hitting sets for \mathcal{J} , denoted \mathcal{H} , and then find the smallest set $H \in \mathcal{H}$, where all candidate equation sets $S \in H$ are realizable.

A. MHS-Based Selection Algorithm

The naive selection approach outlined above is the basis for the procedure SELECTRESGENMHS presented in Algorithm 1, taking as input a model M , a residual generation method \mathcal{M} , and an isolability requirement \mathcal{F} . The output is a set of residual generators \mathcal{R} .

Algorithm 1 MHS-Based Selection of Residual Generators

Input: Model M , residual generation method \mathcal{M} , isolability requirement \mathcal{F}

Output: Set of residual generators \mathcal{R}

```

1: procedure SELECTRESGENMHS( $M, \mathcal{M}, \mathcal{F}$ )
2:    $\mathcal{S} \leftarrow \emptyset$ 
3:    $\mathcal{R} \leftarrow \emptyset$ 
4:    $\mathcal{S}_{\mathcal{M}} \leftarrow \text{FINDCES}(M, \mathcal{M})$ 
5:    $\mathcal{J} \leftarrow \text{ISOLCLASSES}(\mathcal{S}_{\mathcal{M}}, \mathcal{F})$ 
6:    $\mathcal{H} \leftarrow \text{FINDMHS}(\mathcal{J})$ 
7:   while  $\mathcal{H} \neq \emptyset$  do
8:      $H^* \leftarrow \arg \min_{H \in \mathcal{H}} |H|$ 
9:     for all  $S \in H^*$  do
10:       $R \leftarrow \mathcal{M}(S)$ 
11:      if  $R \neq \emptyset$  then
12:         $\mathcal{S} \leftarrow \mathcal{S} \cup \{S\}$ ,  $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$ 
13:      else
14:         $\mathcal{H} \leftarrow \mathcal{H} \setminus \{H^*\}$ 
15:         $\mathcal{S} \leftarrow \emptyset$ ,  $\mathcal{R} \leftarrow \emptyset$ 
16:      break
17:     end if
18:   end for
19:   if  $\mathcal{R} \neq \emptyset$  then
20:     break
21:   end if
22: end while
23: return  $\mathcal{R}$ 
24: end procedure

```

The other procedures used in Algorithm 1 are listed below:

- FINDCES finds all candidate equation sets for the method \mathcal{M} given a model M and a necessary realizability criterion for \mathcal{M} .
- ISOLCLASSES returns the set of all isolability classes of a set of candidate equation sets $\mathcal{S}_{\mathcal{M}}$ for the isolability requirement \mathcal{F} according to (3) and (4).
- FINDMHS finds all minimal hitting sets for the collection of sets \mathcal{J} given as input.

Note that in an efficient implementation of Algorithm 1, it is preferable to keep track of those candidate equation sets that have been realized, successfully or not, in previous iterations in order to avoid unnecessary calls to the procedure $\mathcal{M}(\cdot)$, which may be expensive.

B. Properties of the MHS-Based Selection Algorithm

Algorithm 1 is formally justified by Theorem 1 below. The theorem states that if, and only if, the given isolability requirement can be fulfilled with any set of residual generators created with the given method, then Algorithm 1 finds a set of residual generators fulfilling the requirement. In addition, it is guaranteed that this set of residual generators is of minimal cardinality, i.e., there is no residual generator set of lower cardinality that fulfills the isolability requirement.

Theorem 1. *Let $M = (E, X, Z, F)$ be a model, \mathcal{M} a residual generation method, and \mathcal{F} an isolability requirement. Further, let M , \mathcal{M} , and \mathcal{F} be input to Algorithm 1 and \mathcal{R} the output. Then, \mathcal{F} is fulfilled in M with \mathcal{M} if and only if \mathcal{F} is fulfilled with \mathcal{R} . Further, if \mathcal{F} is fulfilled with \mathcal{R} then \mathcal{R} is of minimal cardinality.*

Proof: Consider first the claim concerning the isolability requirement \mathcal{F} and assume that $\mathcal{R} \neq \emptyset$. Due to rows 10-17 in Algorithm 1, and the fact that $\mathcal{R} \neq \emptyset$, it holds that \mathcal{R} equals (5) and consequently there is a $\mathcal{S} \in \mathcal{H}$ where all $S \in \mathcal{S}$ is realizable with \mathcal{M} . From rows 4-6 and 7 and the definition of \mathcal{J} , see (3) and (4), it can also be deduced that $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$. Hence, \mathcal{S} fulfills the prerequisites of Lemma 1. Further, due to rows 4-6, it can be concluded that \mathcal{S} is a (minimal) hitting set for \mathcal{J} and thus \mathcal{S} fulfills (6). From Lemma 1 it then follows that this property of \mathcal{S} is equivalent to that \mathcal{F} is fulfilled with \mathcal{R} which, according to Proposition 1, is equivalent to that \mathcal{F} is fulfilled in M with \mathcal{M} .

If instead $\mathcal{R} = \emptyset$, rows 4-7 and 10-17 implies that there is no minimal hitting set in \mathcal{H} where all candidate equation sets are realizable with \mathcal{M} . Hence, there is no $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$, where all $S \in \mathcal{S}$ are realizable with \mathcal{M} , that fulfills (6). This is, due to Lemma 1, equivalent to that \mathcal{F} not is fulfilled with \mathcal{R} which is equivalent to that \mathcal{F} not is fulfilled in M with \mathcal{M} , due to Proposition 1. This completes the part of the proof considering the isolability requirement.

Regarding the cardinality of \mathcal{R} , or equivalently \mathcal{S} , it is first noted that a minimal cardinality hitting set also is a minimal hitting set, that is, a hitting set of which no proper subset is a hitting set. Thus, a minimal cardinality hitting set is by necessity found within the collection \mathcal{H} of all minimal hitting sets computed in row 6. Since the search for a realizable minimal hitting set in \mathcal{H} , rows 7-22, is exhaustive and performed by considering the sets in \mathcal{H} in increasing order with respect to cardinality, row 8, it is guaranteed that the first found, and then returned, realizable minimal hitting set is of minimal cardinality. ■

The minimal hitting set problem, or the equivalent minimal set covering problem [14], is unfortunately known to be NP-complete, see e.g., [15], [16], [17]. Thus, for large problems, that is, cases where the number of candidate equation sets $|\mathcal{S}_{\mathcal{M}}|$, as well as the number of isolability classes $|\mathcal{J}|$, is large, it may be intractable to obtain the collection of all minimal hitting sets for \mathcal{J} . Two possible improvements of Algorithm 1, which may overcome this complexity issue, are discussed below.

1) *Using an Approximate MHS Algorithm:* There are several algorithms that give approximate solutions, typically in the

form of a subset of all minimal hitting sets, to the NP-complete minimal hitting set problem, see for example [18] and references therein. A complicating issue is however that for large and complex models, typically, only a fraction of the candidate equation sets are realizable. Indeed, this situation applies to the automotive engine system considered in Section VIII. Typical causes of non-realizability are non-invertible functions in the model, see for example [5], but also numerical issues or instability. For Algorithm 1, this implies that a vast amount of the found minimal hitting sets, possibly all, would be discarded since only a fraction of the found minimal hitting sets contain realizable candidate equation sets. To maximize the possibilities of finding a minimal hitting set in which all candidate equation sets are realizable, it is important to start with as many minimal hitting sets as possible. The reduced number of minimal hitting sets found by an approximate algorithm may therefore not be large enough.

2) *Reducing the Problem Size:* Another alternative approach is to find the realizable subset of all candidate equation sets, $\mathcal{S}'_{\mathcal{M}} = \{S \in \mathcal{S}_{\mathcal{M}} : \mathcal{M}(S) \neq \emptyset\}$, calculate \mathcal{J}' according to (3) and (4) using $\mathcal{S}'_{\mathcal{M}}$ instead of $\mathcal{S}_{\mathcal{M}}$, and then apply a minimal hitting set algorithm to \mathcal{J}' to obtain \mathcal{S} . In general, it holds that $|\mathcal{S}'_{\mathcal{M}}| < |\mathcal{S}_{\mathcal{M}}|$ and $|\mathcal{J}'| < |\mathcal{J}|$, and therefore it is more likely that the set of all minimal hitting sets can be computed for \mathcal{J}' than for \mathcal{J} . The set $\mathcal{S}'_{\mathcal{M}}$ can be computed by applying $\mathcal{M}(\cdot)$ to each $S \in \mathcal{S}_{\mathcal{M}}$. However, realization of an equation set may be a computational demanding task, see Section VIII-B1 for an example. It is therefore desirable to keep the number of realizations, or realization attempts, at a minimum. Consequently, this approach may not be preferable if $\mathcal{S}_{\mathcal{M}}$ is a large set.

It should however be noted that for small problems, where all minimal hitting set can be found, Algorithm 1 works satisfactory and in those cases it provides the globally optimal solution to the selection problem.

VI. GREEDY SELECTION

Taking into account the complexity issues associated with finding all minimal hitting sets, and the objective of keeping the number of realizations at a minimum, a more appealing approach is instead to build the set of candidate equation sets \mathcal{S} iteratively, and only realize those candidate equation sets that are likely to be part of \mathcal{S} . To employ this iterative approach, a heuristic is needed for identifying and selecting a candidate equation set in each iteration.

A. Greedy Heuristic

For the general minimal hitting set problem, or the equivalent set covering problem, a greedy heuristic [19] has shown [20], [21], [22] to provide an approximate solution at a reasonable cost. Using a greedy approach, the candidate equation set with the largest *utility*, is selected in each iteration of the algorithm and added to the solution if it is realizable. The iterations continue until the solution is *complete*. In order to use this approach, a utility function that evaluates the usefulness of a given candidate equation set must be defined, and the

properties of a complete solution to the selection problem must be stated to know when to stop the iterations.

Given the set of isolability classes \mathcal{J} of the candidate equation sets $\mathcal{S}_{\mathcal{M}}$ for the isolability requirement \mathcal{F} , define the *isolability class coverage* of a set $\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}$ as

$$\sigma_{\mathcal{J}}(\mathcal{S}) = \{I \in \mathcal{J} : \exists S \in \mathcal{S}, S \in I\}. \quad (8)$$

Basically, $\sigma_{\mathcal{J}}(\mathcal{S})$ states which of the isolability classes in \mathcal{J} that are covered by the candidate equation sets in \mathcal{S} .

1) *Complete Solution*: A complete solution to the selection problem is characterized as a set of candidate equation sets \mathcal{S} that fulfills (7b) and (7c). The hitting set requirement (7c) can with the isolability class coverage notion be formulated as $\sigma_{\mathcal{J}}(\mathcal{S}) = \mathcal{J}$.

2) *Utility Function*: The aim is to fulfill the isolability requirement, formalized by (7b) and (7c), with as few candidate equation sets as possible (7a). In line with this, the following utility function will be used to evaluate a specific candidate equation set,

$$\mu_{\mathcal{J}}(S) = |\sigma_{\mathcal{J}}(\{S\})|, \quad (9)$$

reflecting how many of the isolability classes in \mathcal{J} that are covered by the candidate equation set $S \in \mathcal{S}_{\mathcal{M}}$. According to the greedy approach the candidate equation set that maximizes $\mu_{\mathcal{J}}(S)$, i.e., covers most isolability classes, should be selected in each iteration.

B. Greedy Selection Algorithm

The procedure `SELECTRESGENGREEDY` for greedy selection of residual generators is presented in Algorithm 2. Input to the algorithm is a model M , a residual generation method \mathcal{M} , and an isolability requirement \mathcal{F} . The output is a set of residual generators \mathcal{R} . The procedures `FINDCES` and `ISOLCLASSES` are the same as in Algorithm 1 and described in Section V-A.

The procedure `PICKCES`, taking a set H containing candidate equation sets as input, returns one of the equation sets in H . This function enables usage of an additional, user-provided, heuristic for selecting one single candidate equation set among candidate equation sets of equal utility by analyzing both structural and analytical properties of equation sets. For instance, `PICKCES` can be used to pick the candidate equation set of lowest cardinality, i.e., containing fewest equations or to pick a candidate equation set not containing a troublesome non-linearity.

Note that the complexity of Algorithm 2 is linear in the number of elements of $\mathcal{S}_{\mathcal{M}}$, in comparison with the NP-completeness of Algorithm 1 originating from the search for all minimal hitting sets. To further understand the complexity of Algorithm 2, the complexity of the procedure `FINDCES` is of most interest. The complexity of `FINDCES` is however dependent of the actual method used for residual generation. For the method employed in Section VIII, the procedure corresponding to `FINDCES` has attractive complexity properties [11].

C. Properties of the Greedy Selection Algorithm

This section explores the properties of Algorithm 2 in terms of providing a solution to the residual generator selection

Algorithm 2 Greedy Selection of Residual Generators

Input: Model M , residual generation method \mathcal{M} , isolability requirement \mathcal{F}

Output: Set of residual generators \mathcal{R}

```

1: procedure SELECTRESGENGREEDY( $\mathcal{M}, E, \mathcal{F}$ )
2:    $\mathcal{S} \leftarrow \emptyset$ 
3:    $\mathcal{R} \leftarrow \emptyset$ 
4:    $\mathcal{S}_{\mathcal{M}} \leftarrow \text{FINDCES}(\mathcal{M}, E)$ 
5:    $\mathcal{J} \leftarrow \text{ISOLCLASSES}(\mathcal{S}_{\mathcal{M}}, \mathcal{F})$ 
6:   while  $\mathcal{J} \neq \emptyset$  do
7:     if  $\mathcal{S}_{\mathcal{M}} \neq \emptyset$  then
8:        $H \leftarrow \{S' \in \mathcal{S}_{\mathcal{M}} : S' = \arg \max_{S \in \mathcal{S}_{\mathcal{M}}} \mu_{\mathcal{J}}(S)\}$ 
9:        $S^* \leftarrow \text{PICKCES}(H)$ 
10:       $R \leftarrow \mathcal{M}(S^*)$ 
11:      if  $R \neq \emptyset$  then
12:         $\mathcal{R} \leftarrow \mathcal{R} \cup \{R\}$ 
13:         $\mathcal{S} \leftarrow \mathcal{S} \cup \{S^*\}$ 
14:         $\mathcal{J} \leftarrow \mathcal{J} \setminus \sigma_{\mathcal{J}}(\{S^*\})$ 
15:      end if
16:       $\mathcal{S}_{\mathcal{M}} \leftarrow \mathcal{S}_{\mathcal{M}} \setminus \{S^*\}$ 
17:    else
18:      return  $\mathcal{R}$ 
19:    end if
20:  end while
21:  return  $\mathcal{R}$ 
22: end procedure

```

problem, i.e., return a set of residual generators fulfilling the isolability and minimal cardinality requirements. The following result justifies Algorithm 2 with regard to the isolability requirement. That is, if, and only if, the isolability requirement can be fulfilled with the given method, then Algorithm 2 finds a set of residual generators with which the isolability requirement is fulfilled.

Theorem 2. *Let $M = (E, X, Z, F)$ be a model, \mathcal{M} a residual generation method, and \mathcal{F} an isolability requirement. Further, let M , \mathcal{M} , and \mathcal{F} be input to Algorithm 2 and \mathcal{R} the output. Then, \mathcal{F} is fulfilled for M with \mathcal{M} if and only if \mathcal{F} is fulfilled with \mathcal{R} . If \mathcal{F} is not fulfilled for M with \mathcal{M} , then \mathcal{R} gives the maximum attainable isolability for M with \mathcal{M} , with respect to \mathcal{F} .*

Proof: According to rows 5, 6, 14, and 21, and rows 4, 7, 16, and 18, there are two different termination conditions in Algorithm 2; either $\mathcal{J} = \emptyset$ or $\mathcal{S}_{\mathcal{M}} = \emptyset$.

Consider first the case when Algorithm 2 terminates because of the condition on row 6, i.e., $\mathcal{J} = \emptyset$, and let n denote the total number of iterations performed by Algorithm 2 in which the condition on row 11 is met. Further let \mathcal{S}_i , \mathcal{R}_i , \mathcal{J}_i , S_i^* , and R_i , denote the values of the variables \mathcal{S} , \mathcal{R} , \mathcal{J} , S^* , and R , respectively, after iteration i . By assumption, and due to row 6, it holds that $\mathcal{J}_n = \emptyset$. Further, it holds that $\mathcal{S}_0 = \mathcal{R}_0 = \emptyset$, and $\mathcal{J}_0 = \mathcal{J}$. By assumption also $\mathcal{R} \neq \emptyset$ and therefore $\mathcal{R}_n \neq \emptyset$ and $\mathcal{S}_n \neq \emptyset$, due to rows 12 and 13. In fact, due to rows 10-12, it can be concluded that $\mathcal{R}_n = \bigcup_{i=1}^{n-1} \{R_i\}$, and $\mathcal{S}_n = \bigcup_{i=1}^{n-1} \{S_i^*\}$, where $R_i = \mathcal{M}(S_i^*)$, and thus each $S_i^* \in \mathcal{S}_n$ is realizable

with \mathcal{M} and the relation between \mathcal{R}_n and \mathcal{S}_n is the same as between \mathcal{R} and \mathcal{S} in (5). Moreover, due to rows 7-9, it holds that each $S_i^* \in \mathcal{S}_n$ is contained in $\mathcal{S}_{\mathcal{M}}$ and therefore \mathcal{S}_n fulfills the prerequisites of Lemma 1. From row 14 it can be deduced that $\mathcal{J}_0 = \bigcup_{i=1}^{n-1} \sigma_J(\{S_i^*\})$. From (8), it follows that for $i = 1, 2, \dots, n-1$ and for all $I \in \sigma_J(\{S_i^*\})$ it holds by definition that $S_i^* \in I$. Therefore, since $\mathcal{S}_n = \bigcup_{i=1}^{n-1} \{S_i^*\}$, it holds that $\mathcal{S}_n \cap I \neq \emptyset$ for all $I \in \mathcal{J}_0 = \bigcup_{i=1}^{n-1} \sigma_J(\{S_i^*\})$. According to Lemma 1, this property of $\mathcal{S} = \mathcal{S}_n$ is equivalent to that \mathcal{F} is fulfilled with $\mathcal{R} = \mathcal{R}_n$ which, due to Proposition 1, is equivalent to that \mathcal{F} is fulfilled in M with \mathcal{M} .

Consider now instead the case when Algorithm 2 terminates because of the condition on row 7 and let n denote the total number of iterations in which the condition on row 11 is met. With similar arguments and notations as above, it holds that $\mathcal{R}_n = \bigcup_{i=1}^{n-1} \{R_i\}$ and $\mathcal{S}_n = \bigcup_{i=1}^{n-1} \{S_i^*\}$, where $R_i = \mathcal{M}(S_i)$. Since termination of Algorithm 2 by assumption was due to the condition on row 7, it holds that $\mathcal{J}_n = \mathcal{J}_0 \setminus \left\{ \bigcup_{i=1}^{n-1} \sigma_J(\{S_i^*\}) \right\} \neq \emptyset$. Thus, there exists $I \in \mathcal{J}_0$ such that $\mathcal{S}_n \cap I = \emptyset$ and consequently, by Lemma 1, it can be deduced that \mathcal{F} not is fulfilled with $\mathcal{R} = \mathcal{R}_n$. However, if $\mathcal{J}' = \bigcup_{i=1}^{n-1} \sigma_J(\{S_i^*\})$ and $\mathcal{F}' = \{(f_i, f_j,) \in \mathcal{F}' : I_{f_i, f_j} \in \mathcal{J}'\}$, Lemma 1 implies that \mathcal{F}' is fulfilled with \mathcal{R} . By assumption and row 7, it holds that $\mathcal{S}_{\mathcal{M}}^n = \emptyset$. Therefore, there are no $S \in \mathcal{S}_{\mathcal{M}}^n$ that can be used to isolate the fault pairs in $\mathcal{F} \setminus \mathcal{F}'$ and thus \mathcal{F}' is the maximum attainable isolability for M with \mathcal{M} . ■

Note that if the isolability requirement not can be fulfilled, the MHS-based Algorithm 1 will return an empty set due to the non-existence of minimal hitting sets. Algorithm 2 will instead provide the best possible solution, in terms of fault isolability, with regard to the given method. However, if the output from Algorithm 2 is an empty set, there are no realizable candidate equation sets that contribute to fulfill the stated isolability requirement.

1) *The Minimal Cardinality Requirement:* Theorem 2 does not consider the minimal cardinality requirement and the purpose of this section is to analyze cardinality properties of the output of Algorithm 2.

To this end, consider the optimization problem formulation (7) of the residual generator selection problem. To be able to exploit a previous result regarding the qualification of the greedy heuristic used in Algorithm 2, the underlying minimal hitting set problem, given by (7a) and (7c), is rewritten as a set-covering problem. Define the set

$$\mathcal{U}_{\mathcal{M}} = \{\sigma_J(\{S\}) : \forall S \in \mathcal{S}_{\mathcal{M}}\}, \quad (10)$$

that is, $\mathcal{U}_{\mathcal{M}}$ is the collection of all isolability classes covered by each candidate equation set in $\mathcal{S}_{\mathcal{M}}$. Consider now the problem of finding a set $\mathcal{U} \subseteq \mathcal{U}_{\mathcal{M}}$ of minimal cardinality that covers $\mathcal{U}_{\mathcal{M}}$, i.e.,

$$\min_{\mathcal{U} \subseteq \mathcal{U}_{\mathcal{M}}} |\mathcal{U}|, \quad \text{s.t.} \quad \bigcup_{U \in \mathcal{U}} U = \bigcup_{U \in \mathcal{U}_{\mathcal{M}}} U. \quad (11)$$

The problem (11) is referred to as a *set covering problem*, and can be shown to be equivalent to the previously considered minimal hitting set problem

$$\min_{\mathcal{S} \subseteq \mathcal{S}_{\mathcal{M}}} |\mathcal{S}|, \quad \text{s.t.} \quad \forall I \in \mathcal{J}, \quad \mathcal{S} \cap I \neq \emptyset, \quad (12)$$

that is, the selection problem (7) with the realizability condition (7b) relaxed. In fact, if \mathcal{U}^* is a solution to the set covering problem (11), a solution \mathcal{S}^* to the minimal hitting set problem (12) can be constructed by finding for each $U \in \mathcal{U}^*$ a $S \in \mathcal{S}_{\mathcal{M}}$ such that $\sigma_J(\{S\}) = U$. The converse is given by (10) with $\mathcal{U}_{\mathcal{M}}$ and $\mathcal{S}_{\mathcal{M}}$ replaced by \mathcal{U}^* and \mathcal{S}^* , respectively.

Consider now solving (11) approximately with a greedy heuristic equivalent to the one described in Section VI. Namely, in each iteration, until all isolability classes in $\mathcal{U}_{\mathcal{M}}$ are covered, select the one $U \in \mathcal{U}_{\mathcal{M}}$ that covers most uncovered isolability classes, i.e., the $U \in \mathcal{U}_{\mathcal{M}}$ of highest cardinality. Denote the resulting solution \mathcal{U} . It can be shown [20], [21], that

$$\frac{|\mathcal{U}|}{|\mathcal{U}^*|} \leq \sum_{j=1}^k \frac{1}{j} \leq \ln k + 1, \quad (13)$$

where \mathcal{U}^* is an exact solution to (11) and k is the cardinality of the largest set in $\mathcal{U}_{\mathcal{M}}$.

As said, the greedy heuristic described above for solving problem (11) coincide with the heuristic described in Section VI for solving problem (12). Since the two problems are equivalent, it can be concluded that the worst case bound (13) also holds for approximate solutions to (12) obtained by usage of the greedy heuristic described in Section VI. This fact is summarized in the following result.

Theorem 3. *Let $M = (E, X, Z, F)$ be a model, \mathcal{M} a residual generation method, and \mathcal{F} an isolability requirement. Further, let M , \mathcal{M} , and \mathcal{F} be input to Algorithm 2 and \mathcal{R} a non-empty output. Then,*

$$\frac{|\mathcal{R}|}{|\mathcal{R}^*|} \leq \sum_{j=1}^k \frac{1}{j} \leq \ln k + 1, \quad (14)$$

where \mathcal{R}^* is the exact solution to the residual generator selection problem, and k is the cardinality of the largest set in $\mathcal{U}_{\mathcal{M}}$, defined according to (10).

Theorem 3 provides a measure, by means of a worst-case error bound, of how well the minimal cardinality requirement is met when solving the selection problem with Algorithm 2. Theorem 3 and Theorem 2 together provide a theoretical justification of Algorithm 2.

Note that if each candidate equation set in $\mathcal{S}_{\mathcal{M}}$ only covers a few of the isolability classes in \mathcal{J} , i.e., k is small, then Algorithm 2 performs well in the sense that the cardinality of its output is close to the cardinality of the exact solution to the selection problem. However, the larger the coverage, the worse the performance. Nevertheless, the approximation ratio (14) increases slowly with k , due to the function $\ln(\cdot)$.

VII. SEQUENTIAL RESIDUAL GENERATION

The purpose of this section is to briefly describe the residual generation method [5], which is employed in the application study presented in Section VIII, and discuss its use in the framework of Section III. For a more thorough description of the method, as well as for technical details, see [5]. It is however noted that the algorithms developed in Sections V and VI are general in the sense that they are aimed

at supporting any computerized residual generation method fulfilling Assumption 2, and not only the particular method [5].

The considered residual generation method belongs to a class of methods referred to as *sequential residual generation*, which has shown to be successful for real applications and also has the potential to be automated to a high extent. Sequential residual generation is based upon the ideas originally described in [8], where unknown variables in a model are computed by solving equation sets one at a time in a sequence and a residual is obtained by evaluating a redundant equation. Similar approaches are described and exploited in for example [9], [23], [10], [13], [1].

A. Computation Sequence

Recall the model $M = (E, X, Z, F)$ considered in Section III, where E is a set of equations, X a set of unknown variables, Z a set of known variables, and F a set of fault variables. An essential component in the design of a sequential residual generator is a *computation sequence*, describing the order and from which equations variables are computed. In [5] a computation sequence is defined as an ordered set of variable and equation pairs

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \dots, (V_k, E_k)), \quad (15)$$

where $V_i \subseteq X \cup D$, $E_i \subseteq E$, and D contains the first-order derivatives of the variables in X . The computation sequence \mathcal{C} implies that first the variables in V_1 are computed from equations E_1 , then the variables in V_2 from equations E_2 and so forth.

B. Sequential Residual Generator

Having computed the unknown variables in $V_1 \cup V_2 \cup \dots \cup V_k$ according to the computation sequence \mathcal{C} in (15), a residual can be obtained by evaluating a redundant equation e , i.e., $e \in E \setminus E_1 \cup E_2 \dots \cup E_k$ with $\text{var}_X(e) \subseteq \text{var}_X(E_1 \cup E_2 \dots \cup E_k)$, where the operator $\text{var}_X(\cdot)$ returns the unknown variables that are contained in an equation set. A residual generator based on a computation sequence \mathcal{C} and redundant residual equation e is referred to as a *sequential residual generator*.

For an example, consider again the model (1) considered in Section III, where $E = \{e_1, e_2, e_3\}$, $X = \{x_1\}$, $Z = \{u, y_1, y_2\}$, and $F = \{f_1, f_2, f_3\}$. A computation sequence for the unknown variable x_1 is given by $\mathcal{C}_1 = ((\{\dot{x}_1\}, \{e_1\}))$. Given \mathcal{C}_1 , e_2 is a redundant residual equation and the corresponding sequential residual generator is

$$\dot{x}_1 = -x_1 + u \quad (16a)$$

$$r = y_1 - x_1. \quad (16b)$$

In fact, also $\mathcal{C}_2 = ((\{x_1\}, \{e_2\}))$ and $\mathcal{C}_3 = ((\{x_1\}, \{e_3\}))$ are computation sequences for x_1 . For instance, the sequential residual generator corresponding to \mathcal{C}_2 and the residual equation e_3 is

$$x_1 = y_1 \quad (17a)$$

$$r = y_2 - x_1. \quad (17b)$$

C. Residual Generation Method

Algorithm 3, see [5] for more details, constructs a sequential residual generator given an equation set S . The output from the algorithm is a sequential residual generator R , if S is realizable with the method, else an empty set.

Algorithm 3 Realization of a Sequential Residual Generator

Input: Equation set S

Output: Sequential residual generator R

```

1: procedure SEQUENTIALRESIDUALGENERATION( $S$ )
2:    $X \leftarrow \text{var}_X(S)$ 
3:   for all  $e \in S$  do
4:      $S' \leftarrow S \setminus \{e\}$ 
5:      $\mathcal{C} \leftarrow \text{FINDCOMPUTATIONSEQUENCE}(S', X)$ 
6:     if  $\mathcal{C} \neq \emptyset$  then
7:        $R \leftarrow \{\mathcal{C} \cup e\}$ 
8:     return  $R$ 
9:   end if
10: end for
11: return  $\emptyset$ 
12: end procedure

```

In Algorithm 3, given the over-determined equation set S , one equation $e \in S$ is removed from S and a *minimal and irreducible* computation sequence, \mathcal{C} , is sought for the unknown variables X in the remaining just-determined set of equations $S' = S \setminus \{e\}$. Whether it is possible or not to find a computation sequence depends naturally on the structural and analytical properties of the equations in the set S' . Equally important are however prerequisites in terms of *causality assumption*, i.e., regarding integral and/or derivative causality, and the properties of the *computational tools*, that are available for use. For more details and a full description of the procedure FINDCOMPUTATIONSEQUENCE, see [5].

D. Fault Sensitivity

In Section III, it was assumed that a residual generation method satisfies Assumption 2. If a residual generation method \mathcal{M} satisfies Assumption 2 it is guaranteed that the residual generator $R = \mathcal{M}(S)$ is sensitive to a fault f if $e_f \in S$. Thus, to verify that the residual generation method given as Algorithm 3 satisfies Assumption 2 it must be shown that a non-empty output R from Algorithm 3 is sensitive to fault f if and only if $e_f \in S$, when $S \subseteq E$ is input.

Assume first that $e_f \notin S$ and note that this implies that no equation in S is affected if fault f is present. Since only equations in S are used in the sequential residual generator $R = \mathcal{M}(S)$ it follows that R can not be sensitive to f .

For the converse, assume that $e_f \in S$ and note that a sequential residual generator consists of a computation sequence and a residual equation. It therefore holds that $R = \{\mathcal{C} \cup e\}$, where \mathcal{C} is a computation sequence for $\text{var}_X(S)$ and e a residual equation. For $R = \{\mathcal{C} \cup e\}$ to be sensitive to fault f , it is necessary that $e = e_f$ or that e_f is contained in any of the equations in \mathcal{C} , i.e., $e_f \in E_1 \cup E_2 \cup \dots \cup E_k$ where $E_i \subseteq E$ when \mathcal{C} is given by (15). Since the former case is trivial due to the fact that $e \in S$, consider the latter and

assume that e_f is *not* used in \mathcal{C} . This implies that there exists a computation sequence \mathcal{C}' for $\text{var}_X(S)$ such that $\mathcal{C}' \subset \mathcal{C}$. However, according to Theorem 4 in [5], a non-empty \mathcal{C} returned by `FINDCOMPUTATIONSEQUENCE` in Algorithm 3 is a *minimal* and *irreducible* computation sequence for $\text{var}_X(S)$. Therefore $\mathcal{C}' \subset \mathcal{C}$ contradicts the minimality of \mathcal{C} and it follows that e_f must be used in \mathcal{C} .

It then remains to show that $R = \{\mathcal{C} \cup e\}$ is sensitive to f if $e_f \in E_1 \cup E_2 \cup \dots \cup E_k$, where $E_i \subseteq E$, or $e_f = e$. Since no restrictions are placed on the model equations E , nothing can in general be guaranteed regarding the analytical properties of the equations in $E_1 \cup E_2 \cup \dots \cup E_k \cup e$. In particular, nothing can be said regarding how the fault f influences the equation e_f in $E_1 \cup E_2 \cup \dots \cup E_k \cup e$ and consequently nor how f influences the residual generator $R = \{\mathcal{C} \cup e\}$. In addition, the effect of f in R is highly dependent on the size and temporal properties of f , and also on for example the current operating conditions. In order to verify that R is sensitive to f , it is thus necessary to implement and run R using representative data from relevant fault cases.

In conclusion, it is hard to theoretically verify that R is sensitive to fault f , given the prerequisites and the general model class considered in this work. It should though be noted that under the idealized assumption that $R = \{\mathcal{C} \cup e\}$ is sensitive to f if $e_f \in \mathcal{C}$ or $e_f = e$, the residual generation method given as Algorithm 3 satisfies Assumption 2. Empirical studies have however shown that Assumption 2 mostly holds in practice. In particular, this is true for the automotive engine system considered in Sections II and VIII. This is discussed in Section VIII-E and exemplified in Figure 7.

E. Necessary Realizability Criterion

In [5] (Theorem 2), it is shown that the equations in a minimal and irreducible computation sequence together with a redundant residual equation, in fact correspond to a Minimal Structurally Overdetermined (MSO) set, see [11]. As said above, a non-empty computation sequence returned by `FINDCOMPUTATIONSEQUENCE` in Algorithm 3 is indeed minimal and irreducible. This means that if an equation set S is realizable with the particular sequential residual generation method [5], then S is an MSO set. Consequently, a necessary realizability criterion for the method [5] is that the equation set used as input is an MSO set and hence an MSO set is a candidate equation set for the method. There are efficient algorithms for finding all MSO sets in a large set of equations, see e.g., [11].

For the model (1), it is possible to find in total three MSO sets. These are given by $S_1 = \{e_1, e_2\}$, $S_2 = \{e_1, e_3\}$, and $S_3 = \{e_2, e_3\}$. In fact, the sequential residual generators (16) and (17) are created from the MSO sets S_1 and S_3 , respectively.

As a side remark, note that the maximum number of sequential residual generators that can be constructed from an MSO set equals the number of equations in the set. All residual generators created from the same MSO set however have equal fault sensitivity properties according to Assumption 2. Nevertheless, their actual fault sensitivity may differ for example due to different sensitivity for noise. To make the

final selection of which of the residual generators created from an MSO set should be included in the final diagnosis system, evaluation by means of real measurements from different fault cases might be needed. For this purpose, Algorithm 3 can be trivially modified to return all residual generators that can be created from the MSO set used input, and not only one.

VIII. APPLICATION EXAMPLE

In this section, the selection algorithms presented in Section V and VI are applied to the automotive engine system introduced in Section II. The residual generation method considered in this study is briefly outlined in Section VII.

A. The Automotive Engine System

Consider again the Scania truck diesel engine system introduced in Section II, which is shown in Figure 1. The main incentive for diagnosis of this system is the stricter emission legislation requirements for heavy-duty trucks, which in turn implies stricter on-board diagnosis (OBD) legislation requirements. The OBD-legislation states that all manufactured vehicles must be equipped with a diagnosis system capable of detecting and isolating faults in all components that, if broken, result in emissions above pre-defined OBD-thresholds during a specified test cycle.

For the considered system, emission critical components include all actuators and sensors, and to meet the OBD-requirements it is desirable that, at least, single faults in these can be detected and isolated. Other emission critical components are pipes and hoses. In particular, a broken pipe or hose may lead to gas-leakage which may increase emissions. Leakages in or near the intercooler, intake manifold, and exhaust manifold are particularly critical and it is desirable that these leakages can be detected and isolated, from each other, but also from all sensor and actuator faults. In total, there are here 12 emission critical components considered and consequently 12 faults that should be isolated from each other in the system. All the 12 considered faults for the system, along with their description, can be found in Table I.

TABLE I
CONSIDERED FAULTS

Fault	Description
$f_{W_{ic}}$	Leakage, intercooler
$f_{W_{im}}$	Leakage, intake manifold
$f_{W_{em}}$	Leakage, exhaust manifold
$f_{u_{x_{th}}}$	Fault, throttle position actuator
$f_{u_{x_{egr}}}$	Fault, EGR-valve position actuator
$f_{u_{x_{vgt}}}$	Fault, VGT-valve position actuator
$f_{y_{p_{amb}}}$	Fault, ambient pressure sensor
$f_{y_{T_{amb}}}$	Fault, ambient temperature sensor
$f_{y_{p_{ic}}}$	Fault, intercooler pressure sensor
$f_{y_{p_{im}}}$	Fault, intake manifold pressure sensor
$f_{y_{T_{im}}}$	Fault, intake manifold temperature sensor
$f_{y_{p_{em}}}$	Fault, exhaust manifold pressure sensor

1) *The Model*: The model of the system used in this work is described in detail in [24] and relies on both fundamental first principle physics and gray-box modeling. The model describes the behavior of the system in the no-fault case, i.e., it is a *nominal* model.

To incorporate fault information in the nominal model, faults are modeled as additive signals in corresponding equations. For example, fault $f_{y_{p_{im}}}$, representing a fault in the intake manifold pressure sensor $y_{p_{im}}$, is modeled by simply adding $f_{y_{p_{im}}}$ to the equation describing the relation between the sensor value $y_{p_{im}}$ and the actual intake manifold pressure p_{im} according to $y_{p_{im}} = p_{im} + f_{y_{p_{im}}}$. It is noted that even though additive fault signals are considered in this application study, the framework in this paper is general and independent on how fault information is incorporated in the model.

In summary, the model contains in total 46 equations, 43 unknown variables, 11 known variables, and the 12 faults in Table I. Of the 11 known variables, 3 are actuators, 6 are sensors, and 2 are control inputs. Of the 46 equations, 5 are differential equations and the rest are algebraic equations. The model contains several non-linear functions.

2) *The Isolability Requirement*: Since it is required that the 12 considered faults can be isolated from each other, the isolability requirement \mathcal{F} for the truck diesel engine system consists of all unique pairwise combinations of the faults in Table I. That is,

$$\mathcal{F} = \left\{ (f_{W_{ic}}, f_{W_{im}}), (f_{W_{ic}}, f_{W_{em}}), \dots, (f_{y_{T_{im}}}, f_{y_{p_{em}}}) \right\}, \quad (18)$$

with $|\mathcal{F}| = 12 \times 11 = 132$.

B. Appliance of the MHS-Based Algorithm

There exists in total 270 candidate equation sets, here MSO sets, for the considered sequential residual generation method in the truck diesel engine system model, i.e., $|\mathcal{S}_{\mathcal{M}}| = 270$. The MSO sets were found using the algorithm in [11], which was implemented as the procedure FINDCES.

As said in Section VII-E, the largest possible number of sequential residual generators that can be constructed from an MSO set equals the number of equations in the set. Thus, the maximum number of residual generators that can be constructed from a set of MSO sets is the sum of the number of equations for all MSO sets. From the set of 270 MSO sets found in the automotive engine system model this number equals 14,242. This is the rationale behind the total number of candidate residual generators mentioned in Section II.

Given the 270 candidate equation sets and the isolability requirement \mathcal{F} defined in (18), 132 isolability classes were created according to (3) and (4), that is, $|\mathcal{J}| = 132$. Due to the complexity of the selection problem, in terms of the cardinalities of the sets $\mathcal{S}_{\mathcal{M}}$ and \mathcal{J} , it was impossible to find the collection of all minimal hitting sets for \mathcal{J} and consequently the MHS-based Algorithm 1 could not be used to solve the automotive engine selection problem.

Some insight regarding the complexity of the selection problem can be gained by studying the total number of minimal hitting sets for smaller instances of the problem. One simple way to reduce the size of the selection problem is to consider

only a subset of the 12 faults in Table I, and then calculate \mathcal{F} and \mathcal{J} for this smaller set of faults. By using a C++ implementation of the minimal hitting set algorithm presented in [25], the set of all minimal hitting sets was computed for several randomized subsets, of various cardinality, of the faults in Table I. Figure 3 shows the total number of minimal hitting sets, in average, as a function of the number of considered faults. From Figure 3, it can be seen that the number of minimal hitting sets grows rapidly with the number of faults, and that the total number of minimal hitting sets is over 30,000 already for 7 faults. Given this, it is not that surprising that the problem with 12 faults was not possible to solve.

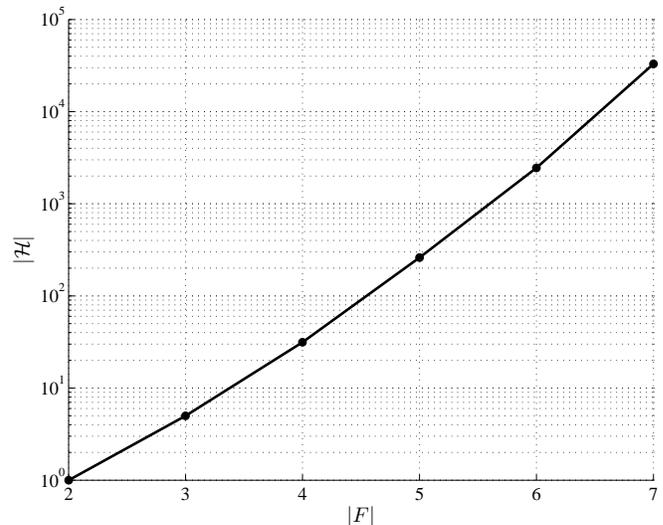


Fig. 3. The total number of minimal hitting sets, $|\mathcal{J}_c|$, as a function of the number of considered faults, $|F|$. Clearly, the number of minimal hitting sets grows exponentially with the number of faults.

1) *Using Improvements of the Algorithm*: Two possible improvements of Algorithm 1 were suggested in Section V-B. One of the proposed improvements was to consider the realizable subset of all candidate equation sets and thereby reduce the size of the involved minimal hitting set problem.

This approach however requires that the realizability of all candidate equation sets are evaluated which, as argued in Section V-B2, may be a computational demanding task. With a MATLAB implementation of the sequential residual generation method outlined in Section VII, the realizability evaluation required 15,778 s \approx 4.38 h on a 2.4 GHz Intel Core 2 Duo PC running Windows XP. In total, only 59 of the 270 candidate equation sets (21.9%) were realizable with the considered sequential residual generation method. The main cause of this relatively large fraction of non-realizable candidate equation sets is non-invertible non-linear functions in the automotive engine model, see [26] for a discussion of a similar result regarding a similar model.

By using the set of 59 realizable candidate equation sets, the size of the selection problem is substantially reduced. Even for this smaller problem, it was unfortunately not possible to compute the set of all minimal hitting sets within feasible time, no termination after 24 h, using the same C++ implementation as above of the minimal hitting set algorithm [25].

TABLE II
FAULT SIGNATURE MATRIX

	$f_{W_{ic}}$	$f_{W_{im}}$	$f_{W_{em}}$	$f_{y_{p_{amb}}}$	$f_{y_{T_{amb}}}$	$f_{y_{p_{ic}}}$	$f_{y_{p_{im}}}$	$f_{y_{T_{im}}}$	$f_{y_{p_{em}}}$	$f_{u_{x_{th}}}$	$f_{u_{x_{egr}}}$	$f_{u_{x_{vgt}}}$
R_1			X	X	X	X	X	X	X	X	X	X
R_2	X	X	X	X	X	X	X	X	X	X	X	X
R_3	X	X	X	X	X	X	X		X	X	X	X
R_4	X	X	X	X	X	X		X	X	X	X	X
R_5		X	X	X		X	X	X	X	X	X	X
R_6	X	X	X	X	X	X	X	X				X
R_7	X		X	X	X		X	X	X	X	X	X
R_8	X	X		X	X	X		X	X	X	X	X
R_9	X	X	X	X	X	X	X		X		X	X
R_{10}	X	X	X	X	X	X	X				X	X
R_{11}	X		X	X		X	X	X	X	X	X	X

The other improvement of Algorithm 1 suggested in Section V-B1 is to use an approximative MHS-algorithm to compute a subset of all minimal hitting sets. Neither this approach did succeed, since it was impossible to find a realizable minimal hitting set within feasible time due to the large number of non-realizable candidate equation sets.

C. Appliance of the Greedy Algorithm

Since it was impossible to use the MHS-based Algorithm 1, or any of the two suggested improvements, to solve the automotive engine selection problem, the greedy Algorithm 2 was employed. Algorithm 2 was implemented in MATLAB. The realization procedure $\mathcal{M}(\cdot)$ was implemented according to Algorithm 3, and the procedure FINDCOMPUTATIONSEQUENCE, for finding computation sequences, according to the corresponding algorithm in [5].

Given the isolability requirement (18) and the automotive engine system model, Algorithm 2 returned a set of 11 residual generators. All of the 11 residual generators were dynamic, 3 used only integral causality and the remaining 8 both integral and derivative causality, i.e., mixed causality. Before terminating, the algorithm discarded in total 119 non-realizable candidate equation sets, mainly due to non-invertible non-linear functions in the model.

Table II shows the fault signature matrix for the 11 selected residual generators with respect to the faults in Table I. The fault signature for a residual generator R contains an “x” in the column corresponding to fault f , if R is sensitive to f in the context of Assumption 2.

As seen in Table II, all of the 11 selected residual generators are sensitive to the faults $f_{y_{p_{amb}}}$ and $f_{u_{x_{vgt}}}$. This is also indicated in Table III, which shows the resulting *isolability matrix* for the set of selected residual generators. The isolability matrix contains an “x” in position (f_i, f_j) if fault f_i is not isolable from fault f_j . Clearly, faults $f_{y_{p_{amb}}}$ and $f_{u_{x_{vgt}}}$ are not isolable from the other faults and the isolability requirement \mathcal{F} , defined in (18), is not met. However, according to Theorem 2, Table III shows the maximum attainable isolability in the automotive engine model with the considered sequential residual generation method.

The rationale to all of the 11 residual generators being sensitive to fault $f_{u_{x_{vgt}}}$ is that all residual generators uses the

TABLE III
ISOLABILITY MATRIX

	$f_{W_{ic}}$	$f_{W_{im}}$	$f_{W_{em}}$	$f_{y_{p_{amb}}}$	$f_{y_{T_{amb}}}$	$f_{y_{p_{ic}}}$	$f_{y_{p_{im}}}$	$f_{y_{T_{im}}}$	$f_{y_{p_{em}}}$	$f_{u_{x_{th}}}$	$f_{u_{x_{egr}}}$	$f_{u_{x_{vgt}}}$
$f_{W_{ic}}$	X			X								X
$f_{W_{im}}$		X		X								X
$f_{W_{em}}$			X	X								X
$f_{y_{p_{amb}}}$				X								X
$f_{y_{T_{amb}}}$				X	X							X
$f_{y_{p_{ic}}}$				X		X						X
$f_{y_{p_{im}}}$				X			X					X
$f_{y_{T_{im}}}$				X				X				X
$f_{y_{p_{em}}}$				X					X			X
$f_{u_{x_{th}}}$				X						X		X
$f_{u_{x_{egr}}}$				X							X	X
$f_{u_{x_{vgt}}}$				X								X

Variable Geometry Turbine (VGT) actuator position signal, $u_{x_{vgt}}$, as input. This stems from the fact that each of the 11 residual generators contains the part of the diesel engine model that describes the dynamics of the VGT, by means of the turbine speed ω_t , and that the signal $u_{x_{vgt}}$ is necessary for calculating ω_t , which is not measured. Similarly, all 11 residual generators are sensitive to fault $f_{y_{p_{amb}}}$ due to that the ambient pressure sensor signal, $y_{p_{amb}}$, is used as input to all 11 residual generators since they all require information regarding the pressure of the ambient air, p_{amb} .

D. Analysis of the Cardinalities of Greedy Solutions

As said in Section VI-C, the greedy Algorithm 2 provides an approximate solution when it comes to fulfillment of the minimal cardinality requirement. Thus, the above mentioned solution to the automotive engine selection problem, i.e., the set of 11 residual generators, may therefore not be of minimal cardinality.

To investigate the performance of Algorithm 2 with respect to the minimal cardinality requirement, it is necessary to know the cardinality of an exact, i.e., minimal cardinality, solution to the selection problem. As said in Section VIII-B it is unfortunately not possible to find all minimal hitting sets for the selection problem when all 12 faults are considered and consequently not possible to find an exact solution using Algorithm 1. There are however algorithms [27] that are able to compute *one* minimal cardinality hitting set for this problem. In practice, this is not sufficient since the obtained minimal cardinality hitting set may contain non-realizable candidate equation sets, see Section V-B. However, from a theoretical point of view and for this investigation, this is sufficient.

For several different instances of the selection problem, and under the assumption that all candidate equation sets were realizable, one greedy solution and one exact, i.e., minimal cardinality, solution were computed. The different instances were obtained by using randomized subsets, of varying cardinality, of the 12 faults in Table I. Figure 4 shows the median cardinalities of the exact, $|\mathcal{R}^*|$, and greedy, $|\mathcal{R}|$, solutions as functions of the cardinality of the set of considered faults, $|F|$.

According to Figure 4, the median cardinalities of the greedy and exact solutions coincide in a majority of the cases. Consequently, it can be concluded that this selection problem suits the greedy selection approach well. Thus, it is likely that the set of 11 residual generators obtained as solution to the selection problem with 12 considered faults in Section VIII-C, is of minimal cardinality, or at least in close proximity.

Figure 5 shows the mean execution times, in logarithmic scale, for the exact and greedy algorithms for the runs described above. Both algorithms were implemented in MATLAB and executed on a 2.4 GHz Intel Core 2 Duo PC running Windows XP. Clearly, the greedy algorithm is magnitudes faster than the exact algorithm. Note that the execution time for computing a minimal cardinality hitting set for the problem with 12 faults is in the magnitude of hundreds of hours.

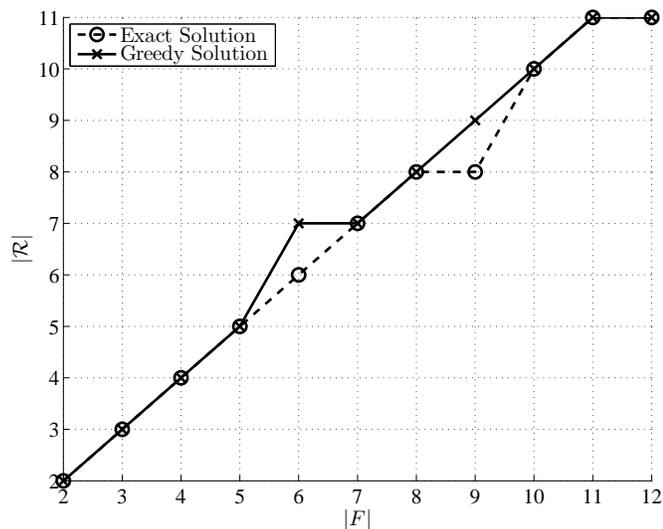


Fig. 4. Median cardinalities of exact and greedy solutions, as functions of the cardinality of the set of considered faults, to the automotive engine selection problem.

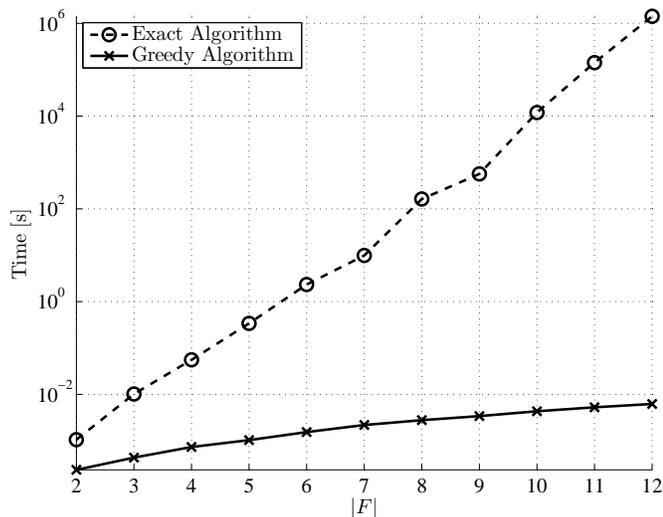


Fig. 5. Mean execution times for the exact and greedy minimal cardinality hitting sets algorithms, as functions of the cardinality of the set of considered faults, for the automotive engine selection problem.

It is also interesting to evaluate the greedy solution to the truck diesel engine selection problem by comparing it with the worst-case bound (14), given in Theorem 3. This bound, along with the median cardinalities of the greedy solutions are shown in Figure 6, for the same instances of the selection problem used above. It can be seen that the cardinalities of the greedy solution differ substantially from the worst-case bound. From this and the fact that the cardinalities of the greedy solutions are more or less equal to the cardinalities of the exact solutions, according to Figure 4, it can be concluded that for the automotive engine selection problem, the bound (14) is very conservative.

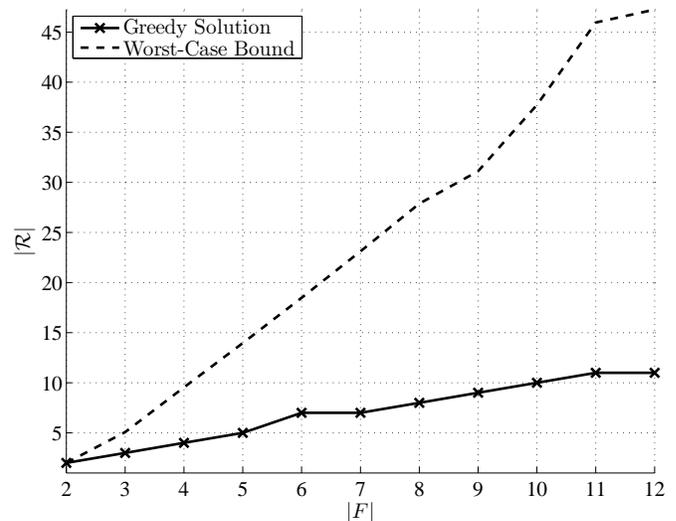


Fig. 6. The median cardinalities of the greedy solution to the truck diesel engine selection problem compared with the worst-case bound provided in Theorem 3.

E. Case Study of Fault Sensitivity

In this section it is shown that the considered approach for design of residual generators, i.e., the proposed selection algorithm together with the residual generation method [5], is applicable to real-world systems characterized by, e.g., uncertain models and noisy measurements. This is done by illustrating how two of the 11 residual generators obtained in Section VIII-C can be used to isolate a pair of faults from each other.

The first residual generator, denoted R_2 in Table II, adopts mixed causality with three state variables and two numerically differentiated measurement signals. The estimated derivatives are of first-order. The residual generator uses in total 11 of the 12 known variables as input. The second residual generator, denoted R_4 , contains 5 state variables and uses 9 known variables as input. This residual generator uses integral causality only.

The considered faults are $f_{y_{p_{im}}}$ and $f_{y_{p_{ic}}}$, i.e., faults in the intake manifold pressure sensor and intercooler pressure sensor, respectively. According to Table II, residual generator R_2 is sensitive to fault $f_{y_{p_{im}}}$ but not to fault $f_{y_{p_{ic}}}$. The residual generator R_4 , on the other hand, is sensitive to $f_{y_{p_{ic}}}$ but not to $f_{y_{p_{im}}}$. Note that the fault sensitivity in Table II is in the context

of Assumption 2, see Section VII-D for a further discussion regarding this.

The residual generators were implemented in a MATLAB/SIMULINK environment and run off-line. As input data, a set of measurements from an engine test bed during a World Harmonized Test Cycle (WHTC) was used. In two separate runs, faults in the intake manifold pressure sensor p_{im} and intercooler pressure sensor p_{ic} were injected. Both faults were in the form of a 20% positive gain of the corresponding pressure sensor signal, i.e., $y_{p_{im}} = 1.2 \times p_{im}$ and $y_{p_{ic}} = 1.2 \times p_{ic}$ where p_{im} and p_{ic} are the actual intake manifold pressure and intercooler pressure signals, respectively.

The residuals obtained as output from the residual generators R_2 and R_4 , for each of the faults $f_{y_{p_{im}}}$ and $f_{y_{p_{ic}}}$, are shown in Figure 7. From the figure it can be seen that residual generator R_2 (top figure) responds to the fault $f_{y_{p_{im}}}$ but not to fault $f_{y_{p_{ic}}}$, and that residual generator R_4 (bottom figure) responds to fault $f_{y_{p_{ic}}}$ but not to fault $f_{y_{p_{im}}}$. Clearly, for these fault cases, R_2 is indeed sensitive to $f_{y_{p_{im}}}$ but not to $f_{y_{p_{ic}}}$, and R_4 sensitive to $f_{y_{p_{ic}}}$ but not to $f_{y_{p_{im}}}$. Thus, fault $f_{y_{p_{im}}}$ is isolable from fault $f_{y_{p_{ic}}}$ and vice versa, with the residual generators R_2 and R_4 .

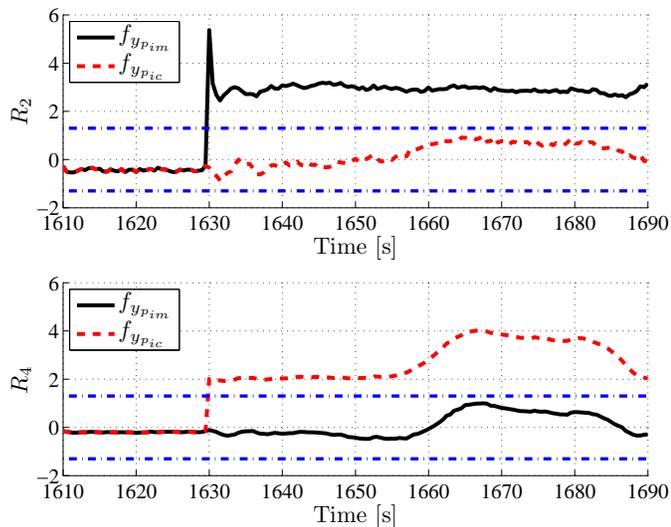


Fig. 7. Residuals from residual generator R_2 (top figure) and residual generator R_4 (bottom figure) for the fault cases $f_{y_{p_{im}}}$ (solid lines) and $f_{y_{p_{ic}}}$ (dashed lines). Both faults are injected at $t = 1630$ s. The dash dotted lines suggest how thresholds may be set in order to detect the faults.

IX. RELATED WORK

Residual generator selection and related problems have been studied in a number of other works. In [2], residual generator selection is considered within a probabilistic framework. Non-ideal fault sensitivity is handled, i.e., Assumption 2 is not needed, but on the other hand, all potential residual generators need to be realized and also evaluated with an extensive use of measurement data. Furthermore, only a very small set of potential residual generators can be handled since the optimal selection is found through a complete search.

In [25], the problem of sequential diagnosis is considered, i.e., iteratively selecting next measurement to finally uniquely identify the fault. Selecting measurements with different

possible outcomes is similar to selecting residual generators. Since the problem of uniquely identifying the fault corresponds to minimizing the entropy of the fault-probability distribution, a heuristic of expected entropy is efficiently utilized in a greedy selection of next measurement to make. However, the residual selection problem does not correspond to a minimization of the entropy of a probability distribution of faults. Therefore, even though selection of measurements is similar to selection of residual generators, an entropy based heuristic cannot be used.

A problem similar to sequential diagnosis is test sequencing, or test planning, which is considered in [28] by exploiting information theory and heuristic search, for instance AO*. A more general treatment of this problem, taking several important practical issues such as test setup operations and precedence constraints into account, is considered in [29], [30].

The problem of troubleshooting, i.e., iteratively selecting actions, e.g., making measurements, until a device can be repaired, is considered in [31]. The selection of next action is made by using a ratio of fault probability and action cost in a greedy search. However, the actions considered are assumed ideal, meaning that it is possible to directly and exactly always measure if a fault is present or not. This is in contrast to our case where each residual only give information whether a set of possible faults can be present or not.

The problem of determining sensor locations to achieve a certain diagnosability objective is a closely related, but not identical, problem to the one studied in this work. See for example [32], [13], [33], [34], [35] which all propose different methods for placing sensors to achieve a certain fault isolability requirement, similar to the objective in Section IV-B. Other works, e.g., [36] propose techniques to find optimal sensor locations.

In conclusion, the problem of residual generator selection has up until now only been studied in a few other papers, and these are unable to handle problems of the size considered here due to the complexity of the search algorithm. In contrast, Algorithm 2 in the present paper can efficiently handle large problems. There are also works that have studied closely related problems, such as the sensor placement problem. The main contribution in the present paper with respect to these works is that here, realizability of the residual generator, an aspect particularly important when analyzing dynamical models, is taken into consideration directly at the selection stage.

X. CONCLUSIONS

Two novel algorithms for solving the residual generator selection problem have been proposed. The foundation for both algorithms was a formulation of the selection problem, in the form of an optimization problem, where the isolability requirement was equivalently stated in terms of properties of subsets of the model equations. The formulation enabled an efficient reduction of the search-space by taking the realizability properties of equation subsets, with respect to the considered residual generation method, into account. Both algorithms are general in the sense that they are aimed at supporting any computerized residual generation method.

Algorithm 1, based on the naive approach of finding all minimal hitting sets, gives an exact solution fulfilling both the isolability and the minimal cardinality requirements but is intractable for large problems. Algorithm 2 is suitable for large, real-world, problems and is based on a greedy heuristic. It provides an approximate solution in terms of fulfilling the minimal cardinality requirement. A theoretical characterization of the approximation error, in the form of a worst-case bound, was given in Theorem 3, and that the output of Algorithm 2 indeed fulfills the isolability requirement was guaranteed by Theorem 2.

The problem of selecting a set of residual generators for detection and isolation of faults in a complex automotive engine system was considered as an industrial application example. Due to the significant complexity of this problem, it was not possible to use the exact MHS-based Algorithm 1 and instead the approximative greedy Algorithm 2 was employed. For this selection problem, the greedy algorithm provides a near-exact solution at a very low cost.

ACKNOWLEDGMENT

This work was sponsored by Scania and VINNOVA (Swedish Governmental Agency for Innovation Systems) and by the Swedish Research Council within The Linnaeus Center CADICS.

REFERENCES

- [1] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed. Springer, 2006.
- [2] M. Nyberg, "Automatic design of diagnosis systems with application to an automotive engine," *Control Engineering Practice*, vol. 87, no. 8, pp. 993–1005, August 1999.
- [3] M. Krysander, "Design and analysis of diagnosis systems using structural methods," Ph.D. dissertation, Linköpings universitet, June 2006.
- [4] M. Nyberg and M. Krysander, "Statistical properties and design criterions for AI-based fault isolation," in *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, 2008, pp. 7356–7362.
- [5] C. Svärd and M. Nyberg, "Residual generators for fault diagnosis using computation sequences with mixed causality applied to automotive systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 6, pp. 1310–1328, 2010.
- [6] R. Patton, P. Frank, and R. Clark, Eds., *Issues of Fault Diagnosis for Dynamic Systems*. Springer, 2000.
- [7] J. Chen and R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.
- [8] M. Staroswiecki and P. Declerck, "Analytical redundancy in non-linear interconnected systems by means of structural analysis," in *Proceedings of IFAC AIPAC'89*, Nancy, France, 1989, pp. 51–55.
- [9] J. Cassar and M. Staroswiecki, "A structural approach for the design of failure detection and identification systems," in *Proceedings of IFAC Control Ind. Syst.*, Belfort, France, 1997, pp. 841–846.
- [10] S. Ploix, M. Desinde, and S. Touaf, "Automatic design of detection tests in complex dynamic systems," in *Proceedings of 16th IFAC World Congress*, Prague, Czech Republic, 2005.
- [11] M. Krysander, J. Åslund, and M. Nyberg, "An efficient algorithm for finding minimal over-constrained sub-systems for model-based diagnosis," *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 38, no. 1, pp. 197–206, 2008.
- [12] E. Gelso, S. Castillo, and J. Armengol, "An algorithm based on structural analysis for model-based fault diagnosis," *Artificial Intelligence Research and Development*, vol. 184, pp. 138–147, 2008.
- [13] L. Travé-Massuyès, T. Escobet, and X. Olive, "Diagnosability analysis based on component-supported analytical redundancy relations," *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 36, no. 6, pp. 1146–1160, November 2006.
- [14] G. Ausiello, A. D'Atri, and M. Protasi, "Structure preserving reductions among convex optimization problems," *Journal of Computer and System Sciences*, vol. 21, no. 1, pp. 136 – 153, 1980.
- [15] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computation*, R. Miller. and J. Thatcher, Eds. New York: Plenum Pres, 1972, pp. 85–103.
- [16] A. Aho, J. Hopcroft, and J. Ullman, *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.
- [17] M. Garey and D. Johnson, *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
- [18] R. Abreu and A. van Gemund, "A low-cost approximate minimal hitting set algorithm and its application to model-based diagnosis," in *Proceedings of the Eighth Symposium on Abstraction, Reformulation, and Approximation*, Lake Arrowhead, California, USA, 2009, pp. 2–9.
- [19] P. Black, "Greedy algorithm," Dictionary of Algorithms and Data Structures (online), U.S. National Institute of Standards and Technology, February 2005, <http://tinyurl.com/3x5zpp>, Accessed: 2010-09-13.
- [20] D. Johnsson, "Approximation algorithms for combinatorial problems," *Journal of Computer and System Sciences*, vol. 9, pp. 256–278, 1974.
- [21] L. Lovász, "On the ratio of optimal integral and fractional covers," *Discrete Math*, 1975.
- [22] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of Operations Research*, vol. 4, no. 3, pp. 233–235, August 1979.
- [23] B. Pulido and C. Alonso-González, "Possible conflicts: a compilation technique for consistency-based diagnosis," *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, Special Issue on Diagnosis of Complex Systems, vol. 34, no. 5, pp. 2192–2206, 2004.
- [24] J. Wahlström and L. Eriksson, "Modeling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 225, no. 7, July 2011.
- [25] J. de Kleer and B. C. Williams, "Diagnosing multiple faults," *Artificial Intelligence*, vol. 32, no. 1, pp. 97–130, 1987.
- [26] C. Svärd, M. Nyberg, E. Frisk, and M. Krysander, "Residual evaluation for fault diagnosis by data-driven analysis of non-stationary probability distributions," in *Proceedings of the 50:th IEEE Conference on Decision and Control and European Control Conference (CDC'11)*, 2011, submitted.
- [27] J. de Kleer, "Hitting set algorithms for model-based diagnosis," in *Proceedings of 22nd International Workshop on Principles of Diagnosis (DX-11)*, Murnau, Germany, 2011.
- [28] K. Pattipati and M. Alexandridis, "Application of heuristic search and information theory to sequential fault diagnosis," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 20, no. 4, pp. 872 –887, jul/aug 1990.
- [29] V. Raghavan, M. Shakeri, and K. Pattipati, "Optimal and near-optimal test sequencing algorithms with realistic test models," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 29, no. 1, pp. 11 –26, jan 1999.
- [30] —, "Test sequencing problems arising in test planning and design for testability," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 29, no. 2, pp. 153 –163, mar 1999.
- [31] D. Heckerman, J. S. Breese, and K. Rommelse, "Decision-theoretic troubleshooting," *Communications of the ACM*, vol. 38, no. 3, pp. 49–57, 1995.
- [32] M. Bhushan and R. Rengaswamy, "Design of sensor network based on the signed directed graph of the process for efficient fault diagnosis," *Industrial & Engineering Chemistry Research*, vol. 39, no. 4, pp. 999–1019, 2000.
- [33] M. Krysander and E. Frisk, "Sensor placement for fault diagnosis," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 6, pp. 1398–1410, Nov. 2008.
- [34] E. Frisk, M. Krysander, and J. Åslund, "Sensor placement for fault isolation in linear differential-algebraic systems," *Automatica*, vol. 45, no. 2, pp. 364–371, 2009.
- [35] R. Raghuraj, M. Bhushan, and R. Rengaswamy, "Locating sensors in complex chemical plants based on fault diagnostic observability criteria," *AIChE Journal*, vol. 45, no. 2, pp. 310–322, 1999.
- [36] M. Basseville, A. Benveniste, G. Moustakides, and A. Rougee, "Optimal sensor location for detecting changes in dynamical behavior," *Automatic Control, IEEE Transactions on*, vol. 32, no. 12, pp. 1067 – 1075, dec 1987.