A Greedy Approach for Selection of Residual Generators

Carl Svärd[1,2], **Mattias Nyberg**[1,2], and **Erik Frisk**[1]

[1] *Department of Electrical Engineering, Linköping University, Sweden*
carl@isy.liu.se
frisk@isy.liu.se
[2] *Scania CV AB, Södertälje, Sweden*
mattias.nyberg@scania.com

ABSTRACT

This paper considers the problem of selecting a set of residual generators, fulfilling requirements in terms of fault isolability and minimal cardinality, for inclusion in a model-based FDI-system. Two novel algorithms for solving the selection problem are proposed. The first one provides an exact solution fulfilling both requirements and is suitable for small problems. The second one, which is the main contribution and suitable for large problems, relaxes the minimal cardinality requirement and provides an approximate solution by means of a greedy heuristic. Both algorithms take the realizability properties of the considered residual generation method into account, but are general in the sense that they support any computerized residual generation method. In a case study the greedy algorithm is applied to the problem of finding a suitable set of residual generators for detection and isolation of faults in a complex truck diesel engine system. In this study a prior known sequential residual generation method is considered.

1 INTRODUCTION

In the FDI-approach to model-based fault diagnosis, a diagnosis system typically contains three sub-systems: residual generation, residual evaluation, and fault isolation, see e.g. (Blanke *et al.*, 2006). In this work, as in for example (Svärd and Nyberg, 2010), (Nyberg, 1999), (Nyberg and Krysander, 2008), 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, expressed as subsets of the model equations, are found and in the second step, the equation sets considered most suitable are selected and residual generators created from these. This paper addresses the selection problem.

The selection problem is formulated by considering two different requirements on the sought set of residual generators. Firstly, it is required that the set of residual generators fulfills an isolability requirement stating which faults that should be isolated from each other. Secondly, regarding implementation aspects such as complexity and computational load, a set of residual generators of low cardinality is preferred before a set of high cardinality, given that the two sets have equal isolability properties. Therefore, it is desirable 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 one provides an exact solution fulfilling both the isolability and the minimal cardinality requirements and is suitable for small problems. The second one, which is the main contribution, relaxes the minimal cardinality requirement and provides an approximate solution by means of a greedy-heuristic in an iterative manner. This algorithm is suitable for large, real-world, problems for which the approach used in the first algorithm is intractable.

Both algorithms exploit a novel formulation of the selection problem, in the form of an optimization problem, which enables an efficient reduction of the search-space by taking the realizability properties of the considered residual generation method into account. In this formulation the isolability requirement is equivalently stated in terms of properties of subsets of the model equations. The proposed algorithms are general in the sense that they support any computerized residual generation method.

The residual generator selection problem is formally stated, in the form mentioned above, in Section 2. The first selection algorithm is presented and discussed in Section 3. The second, greedy, algorithm is presented and justified in Section 4. Section 5 presents an application example in which a specific, and prior known, residual generation method is considered and the greedy selection algorithm is applied to a truck diesel engine system. The paper is concluded in Section 6.

2 THE RESIDUAL GENERATOR SELECTION PROBLEM

In this section, the selection problem is formulated. Prerequisites for the selection is a method for residual generation, an isolability requirement, and a model. The *residual generation method* used for design of residual generators plays a central role and before formulating the selection problem, the important concepts of *isolability* and *realizability*, given a method, are considered.

2.1 Isolability and Realizability

In this work, model-based residual generators are of interest. A model-based *residual generation method*, G, is defined to be a procedure taking as input a set of equations S and giving as output a residual generator R, or an empty set $\emptyset$. An equation set, S, is said to be *realizable* with the method G if the output from G is not the empty set when S is input.

Let G be a residual generation method and consider a *model*, $M = (E, X, Z, F)$, defined in terms of a set of equations E containing the unknown variables X, known variables Z, and considered faults F. A fault $f_i \in F$ is *isolable* from fault $f_j \in F$ in the model M with the method G if there exists an equation set $S \in E$ such that S is realizable with G and the residual generator R, obtained as the output from G when S is input, is sensitive to f_i but not to f_j. Further, a residual generator R is said to be *sensitive* to fault f if the residual r, the output from R, may respond, i.e., deviate from zero, when f is present.

To exemplify these concepts, consider a model containing the following set of differential and algebraic equations

$$\begin{aligned} e_1: &\quad \dot{x}_1 - x_1 - u - f_1 = 0 \\ e_2: &\quad x_1 - y_1 - f_2 = 0 \\ e_3: &\quad x_1 - y_2 - f_3 = 0, \end{aligned}$$

where x_1 is an unknown variable, $\{u, y_1, y_2\}$ known variables, and $\{f_1, f_2, f_3\}$ the considered faults. It can be shown that the equation sets $S_1 = \{e_1, e_2\}$, $S_2 = \{e_1, e_3\}$, and $S_3 = \{e_2, e_3\}$, all are realizable with the sequential residual generation method described in (Svärd and Nyberg, 2010). The output from this method is the residual generator $R_1 = \{\{x_1 = y_1\}, \{r = y_2 - x_2\}\}$, when S_3 is input. Residual generator R_1 is sensitive to both fault f_1 and fault f_2, but not to fault f_3. Thus, fault f_1 and fault f_2 are both isolable from fault f_3 in the above model with the considered sequential residual generation method.

Assuming that each fault occurs in only one 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 1. *Let S be an equation set realizable with the residual generation method G, and R the residual generator obtained as the output from G when S is input. Then, if $e_f \in S$ then R is sensitive to fault f. Moreover, if $e_f \notin S$ then R is not sensitive to f.*

Note that if a fault f occurs 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. This added equation will then be the only equation where f occurs.

Also note that even though additive faults were considered in the example above, the framework in this paper is general and independent on the fault model.

The following result establishes necessary and sufficient condition for fault isolability, given a model and a residual generation method.

Proposition 1. *Let G be a residual generation method and $M = (E, X, Z, F)$ a model. Fault $f_i \in F$ is isolable from fault $f_j \in F$ in M with G if and only if there exists an equation set $S \subseteq E$ that is realizable with G, and for which $e_{f_i} \in S$ and $e_{f_j} \notin S$.*

2.2 Formulation of the Selection Problem

Define an *isolability requirement* as a set, $\mathcal{F}$, of ordered pairs $(f_i, f_j) \in F \times F$, where the interpretation of (f_i, f_j) is that f_i should isolable from f_j. From Proposition 1 it is clear that to be able to satisfy the isolability requirement $\mathcal{F}$ it is necessary, and sufficient, to find for each $(f_i, f_j) \in \mathcal{F}$ a realizable set $S \subseteq E$ for which $e_{f_i} \in S$ and $e_{f_j} \notin S$. If E is a small set, it may be tractable to evaluate all subsets of E in the search for these sets. 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, consider a residual generation method G and an equation set S. A constraint on the equation set S is said to be a *necessary realizability criterion* for the method G if the constraint is satisfied when S is realizable with G. Assume now that for the method G, the necessary realizability criterion is known and well-defined. A *candidate equation set* for the method G is defined as an equation set S that fulfills the necessary realizability criterion for G.

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. (Blanke *et al.*, 2006), (Chen and Patton, 1999) and references therein. An additional example is given by the class of methods referred to as sequential residual generation, see e.g. (Staroswiecki and Declerck, 1989), (Cassar and Staroswiecki, 1997), (Pulido and Alonso-González, 2004), (Ploix *et al.*, 2005), (Travé-Massuyès *et al.*, 2006), (Blanke *et al.*, 2006), (Svärd and Nyberg, 2010), for which over-determined sets of equations, as for example Minimal Structurally Overdetermined (MSO) sets (Krysander *et al.*, 2008) or Minimal Test Equation Supports (MTES) (Krysander *et al.*, 2010), constitute candidate equation sets.

Let $\mathcal{S}_G \subseteq 2^E$ be the set of all candidate equation sets for a residual generation method G. Given an isolability requirement $\mathcal{F}$, define the *isolation class* of $\mathcal{S}_G$ for $(f_i, f_j) \in \mathcal{F}$ as

$$I_{f_i f_j} = \left\{ S \in \mathcal{S}_G : e_{f_i} \in S \wedge e_{f_j} \notin S \right\}, \tag{1}$$

and the set

$$\mathcal{I} = \left\{ I_{f_i f_j} : \forall (f_i, f_j) \in \mathcal{F} \right\}, \tag{2}$$

containing all isolation classes of $\mathcal{S}_G$ for $\mathcal{F}$. The next result formulates the problem of fulfilling an isolability requirement in terms of candidate equation sets.

Theorem 1. *Let $\mathcal{F}$ be an isolation requirement, $M = (E, X, Z, F)$ a model, and G a residual generation method. Also, let $\mathcal{S}_G \subseteq 2^E$ be the set of all candidate equation sets for M and $\mathcal{I}$ the set of all isolation classes of $\mathcal{S}_G$ for $\mathcal{F}$, defined according to (1) and (2). Then, the isolation requirement $\mathcal{F}$ is satisfied for the model M with the method G if and only if there exists $\mathcal{M} \subseteq \mathcal{S}_G$ such that*

$$\forall I \in \mathcal{I}, \quad \mathcal{M} \cap I \neq \emptyset, \tag{3}$$

and each $S \in \mathcal{M}$ is realizable with the method G.

In order to satisfy also the requirement regarding the cardinality of the set of sought residual generators, the cardinality of set $\mathcal{M}$ in Theorem 1 should be minimized. To summarize this discussion, the selection problem is stated as the minimization problem

$$\min_{\mathcal{M} \subseteq \mathcal{S}_G} \quad |\mathcal{M}| \tag{4a}$$

$$\text{s.t.} \quad \forall I \in \mathcal{I}, \quad \mathcal{M} \cap I \neq \emptyset \tag{4b}$$

$$\forall S \in \mathcal{M}, \quad \text{realize}_G(S) \neq \emptyset, \tag{4c}$$

where $|\cdot|$ returns the cardinality of a set, and the function $\text{realize}_G(S)$, taking an equation set as input and returning a residual generator, denotes the procedure of creating a residual generator with method G. In particular, $\text{realize}_G(S) \neq \emptyset$ denotes that S is realizable with G.

In fact, the isolability requirement on $\mathcal{M}$, given by (4b), implies that $\mathcal{M}$ should be a *hitting set* for the set $\mathcal{I}$. To also satisfy the minimal cardinality requirement (4a), $\mathcal{M}$ should be a *minimal cardinality hitting set* for $\mathcal{I}$.

3 A NAIVE SELECTION APPROACH

The observation that $\mathcal{M}$ should be a minimal cardinality hitting set for $\mathcal{I}$ suggests the following naive, but nevertheless simple, selection approach. First find the collection of all minimal hitting sets for $\mathcal{I}$, denoted $\mathcal{H}$, and then find a set $H \in \mathcal{H}$, of minimal cardinality, where all candidate equation sets $S \in H$ are realizable.

3.1 A MHS-Based Selection Algorithm

The naive selection approach outlined above is the basis for the algorithm SELECTRESGENMHS presented below, taking as input a residual generation method G, a model in terms of a set of equations E, and an isolation requirement $\mathcal{F}$. The output from the algorithm is a set of residual generators $\mathcal{R}$.

To make the realizability evaluation of the minimal hitting sets in $\mathcal{H}$ a bit more effective in SELECTRESGENMHS, the candidate equation sets in a minimal hitting set $H \in \mathcal{H}$ are evaluated in decreasing order, with respect to cardinality. This relies on the assumption, which indeed have shown to be valid in practice and in particular in the application example presented in Section 5, that a candidate equation set of low cardinality is more likely to be realizable than a set of high cardinality.

```
function SELECTRESGENMHS(G, E, F)
    M := ∅
    R := ∅
    S_G := FINDCES(G, E)
    I := ISOLCLASSES(S_G, F)
    H := FINDMHS(I)
    while H ≠ ∅ do
        H* := arg min_{H∈H} |H|,  M := H*
        while H* ≠ ∅ do
            S* := arg max_{S∈H*} |S|
            R := REALIZERESGEN(G, S*)
            if R ≠ ∅ then
                R := R ⋃ {R},  H* := H* \ {S*}
                if H* = ∅ then
                    return R
                end if
            else
                H := H \ {M}
                M := ∅,  R := ∅,  H* := ∅
            end if
        end while
    end while
    return R
end function
```

The function FINDCES is assumed to find all candidate equation sets for the method G given a set of equations E. The function ISOLCLASSES is assumed to return the set of all isolation classes of a set of candidate equation sets $\mathcal{S}_G$ for the isolation requirement $\mathcal{F}$ according to (1) and (2). The function FINDMHS is assumed to find all minimal hitting sets for the collection of sets $\mathcal{I}$ given as input. The function REALIZERESGEN is assumed to invoke a call to a user-provided function for realizing a candidate equation set with the residual generation method G. That is, it implements the function $\text{realize}_G(\cdot)$. The input to this function is, besides the residual generation method G, a candidate equation set S for G. The output is a residual generator R, if S is realizable with G, otherwise $\emptyset$. This function will be exemplified, for a specific residual generation method, in Section 5.

Note that in an efficient implementation of algorithm SELECTRESGENMHS, it is preferable to keep book of those candidate equation sets that have been realized, successfully or not, in previous iterations in order to avoid unnecessary calls to REALIZERESGEN.

3.2 Properties of the MHS-Based Algorithm

It is easy to verify that if the output $\mathcal{R}$ from algorithm SELECTRESGENMHS is a non-empty set, the corresponding set $\mathcal{M}$ of selected candidate equation sets is indeed a solution to the selection problem (4). The minimal hitting set problem, or the equivalent set covering problem, is however known to be NP-complete, see e.g. (Karp, 1972), (Aho *et al.*, 1974), (Garey and Johnson, 1979). Thus, for large problems, that is, cases when the number of candidate equation sets $|\mathcal{S}_G|$, as well as the number of isolation classes $|\mathcal{I}|$, is large it may be impossible, or at least intractable, to obtain the collection of all minimal hitting sets for $\mathcal{I}$.

There are nevertheless several algorithms that give approximate solutions, typically a subset of all minimal hitting sets, see for example (Abreu and van Gemund, 2009) 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 described above applies to the truck diesel engine system considered in Section 5. For additional examples and a discussion regarding this, see (Svärd and Nyberg, 2010). For the algorithm SELECTRESGENMHS, 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.

One solution to the complexity issues is to find the realizable subset of all candidate equation sets $\mathcal{S}'_G = \{S \in \mathcal{S}_G : \text{realize}_G(S) \neq \emptyset\}$, calculate $\mathcal{I}'$ according to (1) and (2) using $\mathcal{S}'_G$ instead of $\mathcal{S}_G$, and then apply a minimal hitting set algorithm to $\mathcal{I}'$ to obtain $\mathcal{M}$. The set $\mathcal{S}'_G$ can be computed by applying the function REALIZERESGEN to each $S \in \mathcal{S}_G$. Realization of an equation set in general requires analysis and manipulation of the equations in the set which may be a complex and computational demanding task. It is therefore desirable to keep the number of realizations, or realization attempts, at a minimum. Consequently, this approach may not be preferable if $\mathcal{S}_G$ is a large set.

It should however be noted that for small problems, where all minimal hitting set can be found, algorithm SELECTRESGENMHS works satisfactory and in those cases it provides an exact, and yet straightforward and simple, solution to the selection problem (4).

4 A GREEDY SELECTION APPROACH

Taking the issues regarding complexity and realizability discussed above into account, a more appealing approach is instead to build the set of candidate equation sets $\mathcal{M}$ iteratively, and only realize those candidate equation sets that are likely to be part of $\mathcal{M}$. In this way, the complexity problem associated with finding minimal hitting sets is overcome and the number of needed realizations, or realization attempts, is reduced. To employ this iterative approach, a heuristic is needed for identifying and selecting a candidate equation set in each iteration.

4.1 Greedy Heuristic

For the general minimal hitting set problem, or the equivalent set covering problem, a greedy heuristic (Black, 2005) has shown (Johnsson, 1974), (Lovász, 1975), (Chvatal, 1979) 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. To fulfill the realizability requirement, only realizable candidate equation sets are allowed to be added to the solution. The iterations continue until the solution is *complete*. In order to use this approach, a utility function must be defined, i.e., a function that evaluates the usefulness of a given candidate equation set, and the properties of a complete solution to the selection problem, in this framework, must be stated.

Properties of a Complete Solution

Given a set of isolation classes $\mathcal{I}$, define the *isolation class coverage* of a set $\mathcal{M} \subseteq \mathcal{S}_G$ as

$$\sigma_{\mathcal{I}}(\mathcal{M}) = \{I \in \mathcal{I} : \exists S \in \mathcal{M}, S \in I\}. \tag{5}$$

Basically, $\sigma_{\mathcal{I}}(\mathcal{M})$ states which of the isolation classes in $\mathcal{I}$ that are covered by the candidate equation sets in $\mathcal{M}$. The isolability requirement (4b), or hitting set property, can with the isolation class coverage notion be formulated as $\sigma_{\mathcal{I}}(\mathcal{M}) = \mathcal{I}$, which then characterizes a complete solution to the selection problem.

Utility Function

To evaluate a specific candidate equation set, it is desirable to take into account the isolability and cardinality requirements. In line with this, the following utility function will be used

$$\mu_{\mathcal{I}}(S) = |\sigma_{\mathcal{I}}(\{S\})|, \tag{6}$$

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

As in the algorithm FINDRESGENMHS, it is assumed that a candidate equation set of low cardinality is more likely to be realizable than a set of high cardinality. Consequently, if there are several candidate equation sets with equal utility, the one among these of lowest cardinality will be selected.

4.2 A Greedy Selection Algorithm

The algorithm SELECTRESGENGREEDY for greedy selection and design of residual generators is given below. Input to the algorithm is a method G, model in terms of an equation set E, and an isolation requirement $\mathcal{F}$. The output from the algorithm is a set of residual generators $\mathcal{R}$. The functions FINDCES, ISOLCLASSES, and REALIZERESGEN are the same as in algorithm SELECTRESGENMHS.

1: **function** SELECTRESGENGREEDY$(G, E, \mathcal{F})$
2: $\quad \mathcal{M} := \emptyset$
3: $\quad \mathcal{R} := \emptyset$
4: $\quad \mathcal{S}_G :=$ FINDCES(G, E)
5: $\quad \mathcal{I} :=$ ISOLCLASSES$(\mathcal{S}_G, \mathcal{F})$
6: $\quad$ **while** $\mathcal{I} \neq \emptyset$ **do**
7: $\quad\quad$ **if** $\mathcal{S}_G \neq \emptyset$ **then**
8: $\quad\quad\quad \mathrm{H} := \{S' \in \mathcal{S}_G : S' = \arg\max_{S \in \mathcal{S}_G} \mu_{\mathcal{I}}(S)\}$
9: $\quad\quad\quad S^* := \arg\min_{S \in \mathrm{H}} |S|$
10: $\quad\quad\quad R :=$ REALIZERESGEN(G, S^*)
11: $\quad\quad\quad$ **if** $R \neq \emptyset$ **then**
12: $\quad\quad\quad\quad \mathcal{R} := \mathcal{R} \bigcup \{R\}$
13: $\quad\quad\quad\quad \mathcal{M} := \mathcal{M} \bigcup \{S^*\}$
14: $\quad\quad\quad\quad \mathcal{I} := \mathcal{I} \setminus \sigma_{\mathcal{I}}(\{S^*\})$
15: $\quad\quad\quad$ **end if**
16: $\quad\quad\quad \mathcal{S}_G := \mathcal{S}_G \setminus \{S^*\}$
17: $\quad\quad$ **else**
18: $\quad\quad\quad$ **return** $\mathcal{R}$
19: $\quad\quad$ **end if**
20: $\quad$ **end while**
21: $\quad$ **return** $\mathcal{R}$
22: **end function**

Note that the complexity of algorithm SELECTRESGENGREEDY is linear in the number of elements of $\mathcal{S}_G$, in comparison with the NP-completeness of algorithm SELECTRESGENMHS originating from the search for all minimal hitting sets. For a further comparison of the complexity of algorithm SELECTRESGENGREEDY with the complexity of algorithm SELECTRESGENMHS, only the complexity of FINDCES is of interest since both ISOLCLASSES and REALIZERESGEN are used in similar manner in both algorithms. For the residual generation method exemplified in Section 5, the function corresponding to FINDCES has nice complexity properties, see (Krysander *et al.*, 2008).

4.3 Properties of the Greedy Selection Algorithm

This section explores the properties of algorithm SELECTRESGENGREEDY in terms of providing a solution to the selection problem (4). The algorithm is partly justified by the following result.

Proposition 2. *Let G be a residual generation method, E a set of equations, and $\mathcal{F}$ an isolability requirement. Further, let G, E, and $\mathcal{F}$, be input to* SELECTRESGENGREEDY *and the output be a non-empty $\mathcal{R}$. If the isolability requirement $\mathcal{F}$ can be fulfilled with G, then the set $\mathcal{M}$, where $\mathcal{R} = \{R : R = \mathrm{realize}_G(S), \forall S \in \mathcal{M}\}$, fulfills the isolability and realizability requirements (4b) and (4c), respectively. If the isolability requirement $\mathcal{F}$ not can be fulfilled with G, then the set $\mathcal{M}$ fulfills the realizability requirement and gives the maximum attainable isolability for G, with respect to $\mathcal{F}$.*

Proof. Due to space limitations, only the first claim is shown. Let n denote the total number of iterations performed by SELECTRESGENGREEDY in which the condition on line 11 is met. Further let $\mathcal{M}_i$, $\mathcal{R}_i$, $\mathcal{I}_i$, S_i^*, and R_i, denote the values of the variables $\mathcal{M}$, $\mathcal{R}$, $\mathcal{I}$, S^*, and R, respectively, after iteration i. It then holds that $\mathcal{M}_0 = \mathcal{R}_0 = \emptyset$, and $\mathcal{I}_0 = \mathcal{I}$. By assumption $\mathcal{R} \neq \emptyset$, and therefore $\mathcal{R}_n = \mathcal{R} \neq \emptyset$, $\mathcal{M}_n = \mathcal{M} \neq \emptyset$, and $\mathcal{I}_n = \emptyset$, due to rows 6, 12, and 13, in SELECTRESGENGREEDY. In fact, due to rows 12 and 13 it can be concluded that $\mathcal{R}_n = \bigcup_{i=1}^{n-1}\{R_i\}$, and $\mathcal{M}_n = \bigcup_{i=1}^{n-1}\{S_i^*\}$. Consider now the realizability requirement (4c). From rows 10,11, and

13, it can be deduced that for each $S_i^* \in \mathcal{M}_n$ it holds that $\textsc{realizeResGen}\left(G, S_i^*\right) \neq \emptyset$ and thus each $S_i^* \in \mathcal{M}_n$ is realizable with G and (4c) is satisfied. Consider the isolability requirement (4b). From row 14, it can be deduced that $\mathcal{I} = \mathcal{I}_0 = \bigcup_{i=1}^{n-1} \sigma_{\mathcal{I}}\left(\{S_i^*\}\right)$. From (5), it follows that for $i = 1, 2, \ldots, n-1$ and for all $I \in \sigma_{\mathcal{I}}\left(\{S_i^*\}\right)$ it holds by definition that $S_i^* \in I$. Therefore, since $\mathcal{M} = \mathcal{M}_n = \bigcup_{i=1}^{n-1} \{S_i^*\}$, it holds that $\mathcal{M} \bigcap I \neq \emptyset$ for all $I \in \mathcal{I} = \bigcup_{i=1}^{n-1} \sigma_{\mathcal{I}}\left(\{S_i^*\}\right)$ and the proof is complete. □

The Minimal Cardinality Requirement

In Proposition 2, nothing is said about the cardinality of $\mathcal{M}$, that is, how well the objective (4a) is met when algorithm SELECTRESGENGREEDY is employed for solving the selection problem. The purpose of this section is to analyze this. To be able to exploit a previous result regarding the qualification of the greedy heuristic used in the algorithm, a different but equivalent formulation of the minimal hitting set problem is considered. To this end, define the set

$$\mathcal{U}_G = \sigma_{\mathcal{I}}\left(\mathcal{S}_G\right), \tag{7}$$

that is, $\mathcal{U}_G$ contains the isolation classes covered by each candidate equation set in $\mathcal{S}_G$. Consider now the problem of finding the smallest subset $\mathcal{U} \subseteq \mathcal{U}_G$ that covers $\mathcal{U}_G$, i.e.,

$$\min_{\mathcal{U} \subseteq \mathcal{U}_G} \quad |\mathcal{U}|, \quad \text{s.t.} \quad \bigcup_{U \in \mathcal{U}} U = \bigcup_{U \in \mathcal{U}_G} U. \tag{8}$$

The problem (8) is referred to as a *set covering problem*, and can be shown to be equivalent to the previously considered minimal hitting set problem

$$\min_{\mathcal{M} \subseteq \mathcal{S}_G} \quad |\mathcal{M}|, \quad \text{s.t.} \quad \forall I \in \mathcal{I}, \quad \mathcal{M} \bigcap I \neq \emptyset, \tag{9}$$

that is, the selection problem (4) with the realizability condition (4c) relaxed. In fact, if $\mathcal{U}^*$ is a solution to the set covering problem (8), then

$$\mathcal{M}^* = \{S \in \mathcal{S}_G : \sigma_{\mathcal{I}}\left(\{S\}\right) \in \mathcal{U}^*\} \tag{10}$$

is a solution to minimal hitting set problem (9). The converse is given by (7) with $\mathcal{U}_G$ and $\mathcal{S}_G$ replaced by $\mathcal{U}^*$ and $\mathcal{M}^*$, respectively.

Consider now solving (8) approximately with a greedy heuristic equivalent to the one described in Section 4. Namely, in each iteration, until all isolation classes in $\mathcal{U}_G$ are covered, select the one $U \in \mathcal{U}_G$ that covers most uncovered isolation classes, i.e., the $U \in \mathcal{U}_G$ of highest cardinality. Denote the resulting solution $\hat{\mathcal{U}}$. It can be shown (Johnsson, 1974), (Lovász, 1975), that

$$\frac{\left|\hat{\mathcal{U}}\right|}{\left|\mathcal{U}^*\right|} \leq \sum_{j=1}^{k} \frac{1}{j} \leq \ln k + 1, \tag{11}$$

where $\mathcal{U}^*$ is the exact solution to (8) and k is the cardinality of the largest set in $\mathcal{U}_G$.

As said, the greedy heuristic described above for solving problem (8) coincide with the heuristic described in Section 4 for solving problem (9). Since the two problems are equivalent, it can be concluded that the worst case bound (11) also holds for approximate solutions to (9) obtained by usage of the greedy heuristic described in Section 4. This fact is summarized in the following result.

Proposition 3. *Let $\mathcal{M}^*$ denote the exact solution to (9), $\hat{\mathcal{M}}$ the solution obtained by appliance of algorithm* SELECTRESGENGREEDY*, and k the cardinality of the largest set in $\mathcal{U}_G$, defined according to (7). Then, it holds that*

$$\frac{\left|\hat{\mathcal{M}}\right|}{\left|\mathcal{M}^*\right|} \leq \sum_{j=1}^{k} \frac{1}{j} \leq \ln k + 1. \tag{12}$$

Proposition 3 provides a measure, by means of a worst-case error bound, of how well the objective (4a) is met when solving the selection problem (4) with the algorithm SELECTRESGENGREEDY. Proposition 3 and Proposition 2 together provide a theoretical justification of the algorithm SELECTRESGENGREEDY, with respect to how well its output solves the selection problem (4).

Note that if each candidate equation set in $\mathcal{S}_G$ only covers a few of the isolation classes in $\mathcal{I}$, i.e., k is small, then algorithm SELECTRESGENGREEDY performs well in the sense that the cardinality of its output is close to the cardinality of the exact solution to (4). However, the larger the coverage, the worse the performance. Nevertheless, the approximation ratio (12) increases slowly with k, due to the function $\ln()$.

5 APPLICATION EXAMPLE

In this section, the greedy selection algorithm presented in Section 4 is applied to a truck diesel engine system. The residual generation method used in this study is described in (Svärd and Nyberg, 2010) and 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. These methods are based upon the ideas originally described in (Staroswiecki and Declerck, 1989), 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 (Cassar and Staroswiecki, 1997), (Pulido and Alonso-González, 2004), (Ploix *et al.*, 2005), (Travé-Massuyès *et al.*, 2006), (Blanke *et al.*, 2006). Before presenting the actual application study, the residual generation method is briefly recapitulated and its use in the framework of Section 2 is discussed.

5.1 Sequential Residual Generation

Recall the model $M = (E, X, Z, F)$ considered in Section 2, where E is a set of equations, X a set of unknown variables, Z a set of known variables, and F the set of considered fault. As said above, the main idea in sequential residual generation is to compute unknown variables in the model by solving equation sets one at a time in a sequence, and then evaluate a redundant equation to obtain a residual. An essential component in the design of a residual generator is therefore a *computation sequence*, describing the order and from which equations variables are computed. In (Svärd and Nyberg, 2010), to which is referred for technical details, a computation sequence is defined as an ordered set of variable and equation pairs

$$\mathcal{C} = \left(\left(V_1, E_1\right), \left(V_2, E_2\right), \ldots, \left(V_k, E_k\right)\right), \tag{13}$$

where $V_i \subseteq X \bigcup D$, $E_i \subseteq E$, 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.

Sequential Residual Generator

Having computed the unknown variables in $V_1 \bigcup V_2 \bigcup \ldots \bigcup V_k$ according to the computation sequence $\mathcal{C}$ in (13), a residual can be obtained by evaluating a redundant equation e, i.e. $e \in E \setminus E_1 \bigcup E_2 \ldots \bigcup E_k$ with $\mathrm{var}_X(e) \subseteq \mathrm{var}_X(E_1 \bigcup E_2 \ldots \bigcup E_k)$, where the operator $\mathrm{var}_X(\cdot)$ returns the unknown variables that are contained in an equation set. A residual generator based on a computation sequence $\mathcal{C}$ and redundant equation e is referred to as a *sequential residual generator*.

Realization Algorithm

The algorithm REALIZESEQRESGEN presented below realizes, or attempts to realize, an equation set, S, with the sequential residual generation method briefly outlined above.

```
function REALIZESEQRESGEN(S)
    x := var_X(S)
    for all e ∈ S do
        S' := S \ {e}
        C := FINDCOMPUTATIONSEQUENCE(S', x)
        if C ≠ ∅ then
            R := {C ∪ e}
            return R
        end if
    end for
    return ∅
end function
```

The realization relies heavily on the function FINDCOMPUTATIONSEQUENCE, which finds a *minimal* and *irreducible* computation sequence $\mathcal{C}$ for the variables x, and is described in detail in (Svärd and Nyberg, 2010). Whether it is possible or not to find a computation sequence for a set of variables depends naturally on the properties of the equations. 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.

The following result shows that the residual generation method given by the algorithm REALIZESEQRESGEN satisfies Assumption 1, and thereby is suitable to use in the framework described in Section 2.1.

Proposition 4. *Let S be an equation set. If S is used as input to the algorithm* REALIZESEQRESGEN *and the output is a non-empty R, then R is sensitive to fault f if $e_f \in S$. Moreover, if $e_f \notin S$ then R is not sensitive to fault f.*

Proof. If the output R from the algorithm REALIZESEQRESGEN is non-empty, then $R = \{\mathcal{C} \cup e\}$ where $e \in S$ and $\mathcal{C}$ is a *minimal* and *irreducible* computation sequence for the variables $x = \mathrm{var}_X(S)$, see Theorem 4 in (Svärd and Nyberg, 2010). Under the assumption that $e_f \in S$ the residual generator $R = \{\mathcal{C} \cup e\}$ is sensitive to fault f if e_f is used in the computation sequence $\mathcal{C}$ or if $e = e_f$. Since the latter case is trivial due to the fact that $e \in S$, consider the former and assume that e_f is *not* used in $\mathcal{C}$. This implies that there exists a computation sequence $\mathcal{C}'$ for x such that $\mathcal{C}' \subset \mathcal{C}$. This contradicts the minimality of $\mathcal{C}$ and hence it follows that e_f must be used in $\mathcal{C}$ and consequently R is sensitive to f. The second part of the claim, that is, if $e_f \notin S$ then R is not sensitive to fault f trivially follows from the facts that $R = \{\mathcal{C} \cup e\}$, $e \in S$, and that $\mathcal{C}$ is a computation sequence for the variables $\mathrm{var}_X(S)$. □

5.2 Necessary Realizability Criterion

In Theorem 2 in (Svärd and Nyberg, 2010), 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 (Krysander *et al.*, 2008). It is also shown (as was exploited in the proof of Proposition 4) that a non-empty computation sequence returned by FINDCOMPUTATIONSEQUENCE in the algorithm REALIZESEQRESGEN indeed is minimal and irreducible. Thus, if an equation set S is realizable with the sequential residual generation method then S is an MSO set. Consequently, a necessary realizability criterion for the method 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. Fortunately, there are efficient algorithms for finding all MSO sets in a large set of equations, see e.g. (Krysander *et al.*, 2008).

As a side remark, note that the maximal 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 Section 2.1. Nevertheless, their actual fault sensitivity may differ due for example different sensitivity for noise, etc. To make the final selection of which of the residual generators created from an MSO set that should be included in the final diagnosis system, evaluation by means on execution using real measurements from different fault cases might be needed. For this purpose, algorithm REALIZESEQRESGEN can be trivially modified to return all residual generators that can be created from the MSO set used input, and not only one.

5.3 The Truck Diesel Engine System

In this study, a 13-L six-cylinder Scania diesel engine equipped with Exhaust Gas Recirculation (EGR), Variable Geometry Turbine (VGT), and intake throttle, is considered. Stricter emission legislation requirements for heavy-duty truck diesel engines implies stricter on-board diagnosis (OBD) legislation requirements. The OBD-legislation states that all manufactured vehicles must be equipped with an OBD-system capable of detecting faults in all components that, if broken, result in emissions above predefined OBD-thresholds during a specified test cycle. Consequently, the diagnosis system must be able to detect and isolate faults in all emission critical components.

Isolability Requirement

For the considered truck diesel engine, emission critical components include all actuators and sensors. It is therefore required that, at least, single faults 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 considered. It is desirable that these leakages can be detected and isolated, from each other, but also from all sensor and actuator faults. All considered faults for the truck diesel engine system along with their description can be found in Table 1. Since it is required that all 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 1. That is, $\mathcal{F} = \left\{(f_{W_\text{ic}}, f_{W_\text{im}}), (f_{W_\text{ic}}, f_{W_\text{em}}), \ldots, \left(f_{y_{T_\text{im}}}, f_{y_{p_\text{em}}}\right)\right\}$, and $|\mathcal{F}| = 12 \times 11 = 132$.

Table 1: Considered Faults

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

Truck Diesel Engine Model

The model of the truck diesel engine used in this work is described in (Wahlström and Eriksson, 2011) 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 $f_{y_{p_\text{im}}}$, representing a fault in the intake manifold pressure sensor y_{p_im}, is modeled by simply adding $f_{y_{p_\text{im}}}$ to the equation describing the relation between the sensor value y_{p_im} and the actual intake manifold pressure p_im according to $y_{p_\text{im}} = p_\text{im} + f_{y_{p_\text{im}}}$.

The model contains in total 46 equations, 43 unknown variables, 11 known variables, of which 3 are actuators, 6 sensors, and 2 control inputs, and the 12 faults in Table 1. Of the 46 equations, 5 are differential equations and the rest algebraic equations. The model contains several non-linear functions.

5.4 Results and Discussion

To find all candidate equation sets, i.e., MSO sets, the algorithm in (Krysander *et al.*, 2008) was employed and implemented in the function FINDCES in the selection algorithms. In total, 270 MSO sets, were found in the truck diesel engine model, i.e., $|\mathcal{S}_G| = 270$. Given the isolability requirement defined above, the 270 candidate equation sets were ordered into 132 isolation classes according to (1) and (2) so that $|\mathcal{I}| = 132$. Due to the complexity of the selection problem, especially in terms of the cardinalities of the sets $\mathcal{S}_G$ and $\mathcal{I}$, it was impossible to find the collection of all minimal hitting sets for $\mathcal{I}$, see Section 3.2. Thus, usage of selection algorithm FINDRESGENMHS in Section 3.1 was unfeasible and consequently the greedy selection algorithm FINDRESGENGREEDY from Section 4 was used instead.

The algorithm SELECTRESGENGREEDY was implemented in MATLAB. The function REALIZESEQRESGEN was implemented according to the algorithm in Section 5.1, and the function FINDCOMPUTATIONSEQUENCE, for finding computation sequences, was implemented according to the corresponding algorithm in (Svärd and Nyberg, 2010).

Table 2: Isolability Matrix

	f_{W_ic}	f_{W_im}	f_{W_em}	$f_{y_{p_\text{amb}}}$	$f_{y_{T_\text{amb}}}$	$f_{y_{p_\text{ic}}}$	$f_{y_{p_\text{im}}}$	$f_{y_{T_\text{im}}}$	$f_{y_{p_\text{em}}}$	$f_{u_{x_\text{th}}}$	$f_{u_{x_\text{egr}}}$	$f_{u_{x_\text{vgt}}}$
f_{W_ic}	x			x								x
f_{W_im}		x		x								x
f_{W_em}			x	x								x
$f_{y_{p_\text{amb}}}$				x								x
$f_{y_{T_\text{amb}}}$				x	x							x
$f_{y_{p_\text{ic}}}$				x		x						x
$f_{y_{p_\text{im}}}$				x			x					x
$f_{y_{T_\text{im}}}$				x				x				x
$f_{y_{p_\text{em}}}$				x					x			x
$f_{u_{x_\text{th}}}$				x						x		x
$f_{u_{x_\text{egr}}}$				x							x	x
$f_{u_{x_\text{vgt}}}$				x								x

Table 3: Fault Signature Matrix

	f_{W_ic}	f_{W_im}	f_{W_em}	$f_{y_{p_\text{amb}}}$	$f_{y_{p_\text{amb}}}$	$f_{y_{p_\text{ic}}}$	$f_{y_{p_\text{im}}}$	$f_{y_{T_\text{im}}}$	$f_{y_{p_\text{em}}}$	$f_{u_{x_\text{th}}}$	$f_{u_{x_\text{egr}}}$	$f_{u_{x_\text{vgt}}}$
R_1			x	x	x	x	x	x	x		x	x
R_2	x	x	x	x	x		x	x	x	x		x
R_3	x	x		x	x	x	x		x	x	x	x
R_4	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			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	x
R_{11}	x		x	x		x	x	x	x	x	x	x

The output from the selection algorithm was a set containing 11 residual generators. All of the 11 selected 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 2 shows the resulting isolability matrix for the set of selected residual generators. Table 3 shows the fault signature matrix for the 11 selected residual generators with respect to the faults in Table 1. The fault signature for a residual generator R contains a "x" in the column corresponding to fault f, if R is sensitive to f.

As seen in Table 3, all of the selected residual generators are sensitive to the faults $f_{y_{p_\text{amb}}}$ and $f_{u_{x_\text{vgt}}}$, which is also indicated in Table 2. Thus, faults $f_{y_{p_\text{amb}}}$ and $f_{u_{x_\text{vgt}}}$ are not isolable from the other faults and the isolability requirement $\mathcal{F}$, defined in Section 5.3, is not met. However, according to Proposition 2, Table 2 shows the maximum attainable isolability in the truck diesel engine model with the considered sequential residual generation method.

6 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 which enabled an efficient reduction of the search-space by taking the realizability properties of the residual generation method into account, and in which the isolability requirement was equivalently stated in terms of properties of subsets of the model equations. Both algorithms are general in the sense that they support any computerized residual generation method.

The algorithm FINDRESGENMHS, 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. The iterative algorithm FINDRESGENGREEDY 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 Proposition 3, and that the solution provided by FINDRESGENGREEDY indeed fulfills the isolability requirement was guaranteed by Proposition 2.

In an application example, the greedy algorithm FINDRESGENGREEDY was applied to the problem of finding a set of suitable residual generators for detection and isolation of faults in a complex truck diesel engine system. In the application example, a prior known sequential residual generation method was employed for design of residual generators.

REFERENCES

(Abreu and van Gemund, 2009) R. Abreu and A.J.C 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*, pages 2–9, Lake Arrowhead, California, USA, 2009.

(Aho *et al.*, 1974) A.V. Aho, J.E. Hopcroft, and J.D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

(Black, 2005) P.E. Black. Greedy algorithm. Dictionary of Algorithms and Data Structures (online), U.S. National Institute of Standards and Technology, February 2005. http://tinyurl.com/3x5zzpp, Accessed: 2010-09-13.

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

(Cassar and Staroswiecki, 1997) J.P. Cassar and M. Staroswiecki. A structural approach for the design of failure detection and identification systems. In *Proceedings of IFAC Control Ind. Syst.*, pages 841–846, Belfort, France, 1997.

(Chen and Patton, 1999) J. Chen and R.J. Patton. *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.

(Chvatal, 1979) V Chvatal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, August 1979.

(Garey and Johnson, 1979) M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

(Johnsson, 1974) D.S Johnsson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256–278, 1974.

(Karp, 1972) R.M. Karp. Reducibility among combinatorial problems. In R.E. Miller. and J.W. Thatcher, editors, *Complexity of Computer Computation*, pages 85–103, New York, 1972. Plenum Pres.

(Krysander *et al.*, 2008) 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*, 38(1):197–206, 2008.

(Krysander *et al.*, 2010) M. Krysander, J. Åslund, and E. Frisk. A structural algorithm for finding testable sub-models and multiple fault isolability analysis. In *21st International Workshop on Principles of Diagnosis (DX-10)*, Portland, Oregon, USA, 2010.

(Lovász, 1975) L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete Math*, 1975.

(Nyberg and Krysander, 2008) Mattias Nyberg and Mattias Krysander. Statistical properties and design criterions for AI-based fault isolation. In *Proceedings of the 17th IFAC World Congress*, pages 7356–7362, Seoul, Korea, 2008.

(Nyberg, 1999) Mattias Nyberg. Automatic design of diagnosis systems with application to an automotive engine. *Control Engineering Practice*, 87(8):993–1005, August 1999.

(Ploix *et al.*, 2005) 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.

(Pulido and Alonso-González, 2004) B. Pulido and C. Alonso-González. Possible conflicts: a compilation technique for consistency-based diagnosis. *"IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics", Special Issue on Diagnosis of Complex Systems*, 34(5):2192–2206, 2004.

(Staroswiecki and Declerck, 1989) M. Staroswiecki and P. Declerck. Analytical redundancy in non-linear interconnected systems by means of structural analysis. In *Proceedings of IFAC AIPAC'89*, pages 51–55, Nancy, France, 1989.

(Svärd and Nyberg, 2010) 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*, 40(6):1310–1328, 2010.

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

(Wahlström and Eriksson, 2011) 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*, 225(7), July 2011.