

Automated Design of an FDI-System for the Wind Turbine Benchmark

Carl Svård* Mattias Nyberg**

* *Department of Electrical Engineering, Linköping University, Sweden.
Scania CV AB, Södertälje, Sweden. (carl@isy.liu.se)*

** *Scania CV AB, Södertälje, Sweden (mattias.nyberg@scania.com)*

Abstract: Present paper proposes an FDI-system for the wind turbine benchmark designed by application of a generic automated design method, in which the number of required human decisions and assumptions are minimized. No specific adaptation of the method for the wind turbine benchmark is needed, and the number of parameter choices is small. The method contains in essence three steps: generation of potential residual generators; residual generator selection; and diagnostic test construction. The second and third step are based on novel ideas developed in this paper; a greedy selection algorithm for the second step, and a methodology based on the Kullback-Leibler divergence for the third step. The proposed FDI-system performs well in spite of no specific adaptation or tuning to the benchmark. All faults in the pre-defined test sequence can be detected and all faults, except a double fault, can also be isolated shortly thereafter. In addition, there are no false or missed detections.

Keywords: fault diagnosis, fault detection, diagnosis, wind turbine benchmark

1. INTRODUCTION

Wind turbines stand for a growing part of power production. The demands for reliability are high, since wind turbines are expensive and their off-time should be minimized. One potential way to meet the reliability demands is to adopt fault tolerant control (FTC), i.e., prevent faults from developing into failures by taking appropriate actions. A typical action is reconfiguration of the control system. An essential part of an FTC-system is fault detection and isolation (FDI). To obtain good detection and isolation of faults, model-based diagnosis is often necessary.

Design of a complete model-based FDI-system is a complex task and involves by necessity several decisions and assumptions. For example the choice of design method, tuning of parameters, and assumptions regarding, e.g., noise distributions and the nature of the faults to be diagnosed. All these activities are time-consuming and in general, an optimal solution requires detailed knowledge of the behavior of the considered system, something that is rarely available for real applications.

Consequently, inspired by work with real industrial applications, we have made a strong effort to develop an automated design process that minimizes the number of required human decisions and assumptions. The paper investigates the potential of designing an FDI-system for the wind turbine benchmark, see Fogh Odgaard et al. [2009], using this philosophy. The methodology used in the design is general, and no specific adaptation to the wind turbine benchmark is necessary.

The design method is composed of three main steps, of which the second and third are based on novel ideas

developed in this paper. In the first step, a large set of potential residual generators are generated. In the second step, the residual generators most suitable to be included in the final FDI-system are selected and then constructed by use of the algorithms presented in Svård and Nyberg [2010]. The selection is done by means of a greedy selection algorithm. In the third and final step, diagnostic tests for the selected set of residual generators are designed. The diagnostic tests are based on a comparison between the estimated probability distributions of residuals, evaluated with current and no-fault data.

As it turns out, the proposed FDI-system performs well when evaluated on the test sequence described in Fogh Odgaard et al. [2009]. All faults in the test sequence can be detected within feasible time, and there are no false or missed detections. Further, all faults, except a double fault, can also be isolated. A tailor-made FDI-system perfectly tuned for the wind turbine benchmark would probably perform better than the one we propose. However, in relation to the required effort for application of the automated design method, and in spite of no extra tuning or specific adaptation to the benchmark, the performance of the FDI-system is good.

The wind turbine benchmark model and the strategy used for modeling of faults, are described in Section 2. Section 3 presents an overview of the proposed solution and subsequent sections describes in detail the different steps of the applied design method: the residual generation in Section 4, and the residual generator selection in Section 5. The method for design of diagnostic tests, and the fault isolation scheme is considered in Section 6. The performance of the designed FDI-system is evaluated and discussed in Section 7, and Section 8 concludes the paper.

2. THE WIND TURBINE BENCHMARK MODEL

The wind turbine system is described and modeled in Fogh Odgaard et al. [2009]. The system contains six sub-systems: blade system, pitch system, drive train, generator, converter, and controller, see Figure 1.

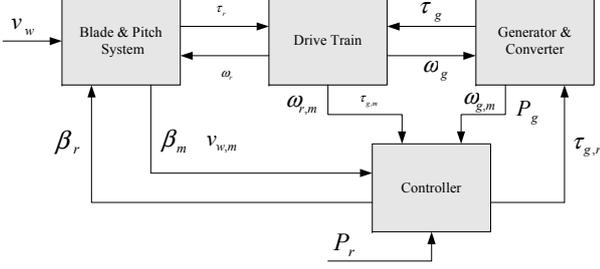


Fig. 1. Overview of the wind turbine System.

2.1 State-Space Realization of Transfer Functions

The pitch system and converter are modeled as frequency domain transfer functions. The residual generation algorithm we intend to apply, assume a model of differential and algebraic equations. To obtain a model in this form, the transfer functions are realized as time-domain state-space systems.

The relation between pitch angle reference $\beta_{i,r}$ and pitch angle output β_i can be realized in state-space form, in fact observable canonical form see e.g. Rugh [1996], as follows

$$\dot{x}_{\beta_{i1}}(t) = -2\zeta\omega_n x_{\beta_{i1}}(t) + x_{\beta_{i2}}(t) \quad (1)$$

$$\dot{x}_{\beta_{i2}}(t) = -\omega_n^2 x_{\beta_{i1}}(t) + \omega_n^2 \beta_{i,r}(t) \quad (2)$$

$$\beta_i(t) = x_{\beta_{i1}}(t), \quad (3)$$

for $i = 1, 2, 3$. Using the same approach, the relation between converter reference $\tau_{g,r}$ and output τ_g can be written as

$$\dot{x}_{\tau_g}(t) = -\alpha_{gc} x_{\tau_g}(t) + \alpha_{gc} \tau_{g,r}(t) \quad (4)$$

$$\tau_g(t) = x_{\tau_g}(t). \quad (5)$$

2.2 Fault Modeling

The set of faults to consider for the wind turbine is specified in Fogh Odgaard et al. [2009] and given by

$$F = \{\Delta\beta_1, \Delta\beta_2, \Delta\beta_3, \Delta\tau_g, \Delta\omega_r, \Delta\omega_g, \Delta\beta_{1,m1}, \Delta\beta_{1,m2}, \Delta\beta_{2,m1}, \Delta\beta_{2,m2}, \Delta\beta_{3,m1}, \Delta\beta_{3,m2}, \Delta\omega_{r,m1}, \Delta\omega_{r,m2}, \Delta\omega_{g,m1}, \Delta\omega_{g,m2}\},$$

where $\Delta\beta_1, \Delta\beta_2, \Delta\beta_3$, and $\Delta\tau_g$ are actuator faults, $\Delta\omega_r$ and $\Delta\omega_g$ system faults, and $\Delta\beta_{1,m1}, \Delta\beta_{1,m2}, \Delta\beta_{2,m1}, \Delta\beta_{2,m2}, \Delta\beta_{3,m1}, \Delta\beta_{3,m2}, \Delta\omega_{r,m1}, \Delta\omega_{r,m2}, \Delta\omega_{g,m1}$, and $\Delta\omega_{g,m2}$, sensor faults.

To incorporate fault information in the nominal model, we have chosen to model all faults as additive signals in corresponding equations. Thus, we are not taking into account all information regarding the nature of faults given in Fogh Odgaard et al. [2009]. Consider for example fault $\Delta\beta_1$ which represents an actuator fault in pitch system 1, see (1)-(3), resulting in changed dynamics of β_1 due to dropped main line pressure or high air content in the oil. One possible way to model this fault would be as a deviation in parameters ω_n and ζ in (1) and (2). With

the chosen signal approach, the fault is instead modeled as an additive signal in (3) for $i = 1$, i.e., $\beta_1 = x_{\beta_{11}} + \Delta\beta_1$.

Note that the adopted fault modeling approach is general and no assumptions are made regarding for example the time-behavior of faults. Thus, the approach is able to handle for example multiplicative faults even though the fault signal is assumed to be additive. Consider for example a multiplicative fault in β_1 given by $\beta_1 = \delta \cdot x_{\beta_{11}}$, $\delta \neq 1$, which can be equivalently described by $\Delta\beta_1 = x_{\beta_{11}}(\delta - 1)$.

The main argument for using this, more general, approach is that we consider it hard, or even impossible, to know exactly how a faulty component behaves in reality. Furthermore, data from all fault-cases for evaluation and validation of a more detailed model are seldom available. Modeling faults in this way also results in a minimum of fault modes. This is beneficial since it gives a smaller model which simplifies several steps in model-based diagnosis, e.g. residual generation and isolation. In addition, regarding how diagnosis information is utilized, e.g., for Fault Tolerant Control, it is unnecessary to distinguish between different fault modes if they are associated with the same action or consequence. Indeed, this applies to all sensor faults in the wind turbine, since the system should be reconfigured regardless of the type of sensor fault, i.e. *fixed value* or *gain factor*, see Table 2 in Fogh Odgaard et al. [2009]. Last, but not least, an additional important motivator is simplicity, since extending the nominal model with additive fault signals in this way is straightforward and easy.

2.3 Model Extensions

According to Fogh Odgaard et al. [2009], the same pitch angle reference signal β_r is fed to all three pitch systems (1)-(3), i.e., $\beta_{i,r} = \beta_r$ for $i = 1, 2, 3$. However, according to the provided SIMULINK[®] model, the individual reference signals are instead calculated in a control loop outside the pitch system as

$$\beta_{i,r} = \beta_r + \beta_i - \left(\frac{\beta_{i,m1} + \beta_{i,m2}}{2} \right), \quad i = 1, 2, 3 \quad (6)$$

where β_i is given by (1)-(3), and $\beta_{i,m1}$ and $\beta_{i,m2}$ are sensor measurements. To incorporate this information in the design of the FDI system, the original wind turbine model is extended with the relations between $\beta_{i,r}$ and β_r given by (6).

2.4 The Model with Faults

The complete model of the wind turbine model, with fault signals denoted by Δ , used in this work for design of an FDI-system is given below.

$$\begin{aligned} e_1 : \quad \tau_r &= \sum_{i=1}^3 \frac{\rho\pi R^3 C_q(\lambda, \beta_i) v_w^2}{6} \\ e_2 : \quad \lambda &= \frac{\omega_r R}{v_w} \\ e_3, e_5, e_7 : \quad \dot{x}_{\beta_{i1}} &= -2\zeta\omega_n x_{\beta_{i1}} + x_{\beta_{i2}}, \quad i = 1, 2, 3 \\ e_4, e_6, e_8 : \quad \dot{x}_{\beta_{i2}} &= -\omega_n^2 x_{\beta_{i1}} + \omega_n^2 \beta_{i,r}, \quad i = 1, 2, 3 \\ e_9, e_{10}, e_{11} : \quad \beta_i &= x_{\beta_{i1}} + \Delta\beta_i, \quad i = 1, 2, 3 \\ e_{12} : \quad \dot{\omega}_g &= \left(\frac{\eta_{dt} B_{dt}}{N_g J_g} \right) \omega_r + \left(\frac{-\frac{\eta_{dt} B_{dt}}{N_g^2} - B_g}{J_g} \right) \omega_g + \end{aligned}$$

$$\begin{aligned}
& \left(\frac{\eta_{dt} K_{dt}}{N_g J_g} \right) \theta_{\Delta} - \left(\frac{1}{J_g} \right) \tau_g + \Delta \omega_g \\
e_{13} : \quad \dot{\omega}_r &= - \left(\frac{B_{dt} - B_r}{J_r} \right) \omega_r + \left(\frac{B_{dt}}{N_g J_r} \right) \omega_g - \\
& \left(\frac{K_{dt}}{J_r} \right) \theta_{\Delta} + \left(\frac{1}{J_r} \right) \tau_r + \Delta \omega_r \\
e_{14} : \quad \dot{\theta}_{\Delta} &= \omega_r - \left(\frac{1}{N_g} \right) \omega_g \\
e_{15} : \quad \dot{x}_{\tau_g} &= -\alpha_{gc} x_{\tau_g} + \alpha_{gc} \tau_{g,r} \\
e_{16} : \quad \tau_g &= x_{\tau_g} + \Delta \tau_g \\
e_{17} : \quad P_g &= \eta_{gc} \omega_g \tau_g \\
e_{18}, e_{20}, e_{22} : \quad \beta_{i,m1} &= \beta_i + \Delta \beta_{i,m1}, \quad i = 1, 2, 3 \\
e_{19}, e_{21}, e_{23} : \quad \beta_{i,m2} &= \beta_i + \Delta \beta_{i,m2}, \quad i = 1, 2, 3 \\
e_{24}, e_{25} : \quad \omega_{r,mj} &= \omega_r + \Delta \omega_{r,mj}, \quad j = 1, 2 \\
e_{26}, e_{27} : \quad \omega_{g,mj} &= \omega_g + \Delta \omega_{g,mj}, \quad j = 1, 2 \\
e_{28} : \quad v_{w,m} &= v_w \\
e_{29} : \quad \tau_{g,m} &= \tau_g \\
e_{30} : \quad P_{g,m} &= P_g \\
e_{31}, e_{32}, e_{33} : \quad \beta_{i,r} &= \beta_r + \beta_i - \left(\frac{\beta_{i,m1} + \beta_{i,m2}}{2} \right), \quad i = 1, 2, 3
\end{aligned}$$

3. OVERVIEW OF PROPOSED SOLUTION

The proposed FDI-system for the wind turbine is comprised of three sub-systems: residual generation, fault detection and fault isolation, see Figure 2.

Measurements, i.e., sensor readings, from the wind turbine are fed to a bank of residual generators whose output is a set of residuals. The residuals are used as input to the fault detection block. This block contains diagnostic tests based on the residuals, and its output indicates if any test has alarmed, that is, if a fault has been detected in the part of the system monitored by the corresponding residual. The result from the fault detection is fed to the fault isolation block in which the detected fault(s) are isolated.

In the subsequent sections, we describe in detail the different steps of the design method used to create the proposed FDI-system.

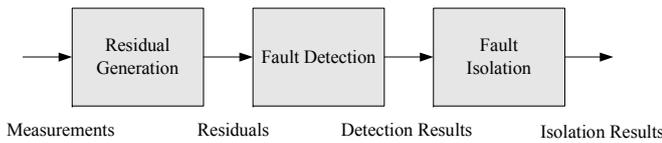


Fig. 2. Schematic Overview of the FDI-System.

4. RESIDUAL GENERATION

The set of residual generators used in the FDI-system 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], Staroswiecki [2002], Pulido and Alonso-González [2004], Ploix et al. [2005], Travé-Massuyès et al. [2006], Blanke et al. [2006], Svärd and Nyberg [2010]. This class

of residual generation methods, referred to as *sequential residual generation*, has shown to be successful for real applications and also has the potential to be automated to a high extent.

4.1 Sequential Residual Generation

We will now very briefly recapitulate some concepts and results of sequential residual generation given in Svärd and Nyberg [2010], to which we also refer for technical details.

We consider a model (E, X, D, Z) to be a set of differential and algebraic equations $E = \{e_1, \dots, e_m\}$ containing unknown variables $X = \{x_1, \dots, x_n\}$, differential variables $D = \{\dot{x}_1, \dots, \dot{x}_n\}$, and known variables $Y = \{y_1, \dots, y_r\}$. The equations in E are, without loss of generality, assumed to be on the form

$$e_i : f_i(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{y}) = 0, \quad i = 1, \dots, m, \quad (7)$$

where $\dot{\mathbf{x}}$, \mathbf{x} and \mathbf{y} are vectors of the variables in D , X , and Y respectively. Note that the model of the wind turbine presented in Section 2.4 can trivially be cast into this form.

Computation Sequence The order and from which equations variables are computed is described by a *computation sequence*, defined as an ordered set of variable and equation pairs

$$\mathcal{C} = ((V_1, E_1), (V_2, E_2), \dots, (V_k, E_k)), \quad (8)$$

where $V_i \subseteq X \cup D$ and $E_i \subseteq E$. The computation sequence \mathcal{C} implies that we first compute the variables in V_1 from equations E_1 , then the variables in V_2 from equations E_2 and so forth. Whether or not it is possible to compute the specified variables from the corresponding equations, depends naturally on the analytical properties of the equations. Equally important are however the *causality assumption*, i.e., if integral and/or derivative causality should be used to handle differential equations, and the properties of the *computational tools* that are available for use. Computational tools, as for example analytical or numerical equation solvers, are essential for handling so called algebraic loops. For a detailed discussion regarding algebraic loops, computational tools, and causality see Svärd and Nyberg [2010].

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 (8), 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 $\text{var}_X(\cdot)$ tells which unknown variables 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*.

Finding Sequential Residual Generators Regarding implementation aspects, e.g. complexity and numerical issues, it is unnecessary to compute variables that are not contained in the residual equation, or not used to compute any of the variables contained in the residual equation. Furthermore, it is also desirable that computation of variables in each step is performed from as small equation sets as possible. It can be shown, see Svärd and Nyberg [2010], that the equations in a computation sequence fulfilling the above properties, together with a redundant residual equation, in fact correspond to an Minimal Structurally

Overdetermined (MSO) set, see Krysander et al. [2008]. In other words, a necessary condition for the existence of a sequential residual generator for a model is that the model, or sub-model, is an MSO set.

4.2 Structural Analysis of Wind Turbine Model

Consider now the model of the wind turbine described in Section 2.4, with equations $E = \{e_1, e_2, \dots, e_{33}\}$, unknown variables

$$X = \{\tau_r, \beta_1, \lambda, v_w, \beta_2, \beta_3, \omega_r, x_{\beta_{11}}, x_{\beta_{12}}, \beta_{1,r}, x_{\beta_{21}}, x_{\beta_{22}}, \beta_{2,r}, x_{\beta_{31}}, x_{\beta_{32}}, \beta_{3,r}, \omega_g, \theta_\Delta, \tau_g, x_{\tau_g}, P_g\},$$

and known, i.e., measured, variables

$$Z = \{\beta_r, \tau_{g,r}, \beta_{1,m1}, \beta_{1,m2}, \beta_{2,m1}, \beta_{2,m2}, \beta_{3,m1}, \beta_{3,m2}, \omega_{r,m1}, \omega_{r,m2}, \omega_{g,m1}, \omega_{g,m2}, v_{w,m}, \tau_{g,m}, P_{g,m}\}.$$

In summary, the model contains 33 equations, 21 unknown variables, and 15 known variables.

As indicated in Section 4, a first step when searching for a sequential residual generator for a model may be to find an MSO set in the model. There are efficient algorithms for finding all MSO sets in large equation sets, see e.g. Krysander et al. [2008]. By using a MATLAB[®] implementation of the algorithm presented in Krysander et al. [2008], 1058 MSO sets was found in total. Given an MSO set we may construct a sequential residual generator by removing one equation from the set and then find a computation sequence in the remaining, just-determined, set of equations, see e.g. Svård and Nyberg [2010]. Hence, the number of potential sequential residual generators that can be constructed from an MSO set equals the number of equations in the set, and the total number of potential sequential residual generators in a model is the total sum of the equations in each of all MSO sets for the model. For the wind turbine model, there are in total 15248 potential sequential residual generators.

5. SELECTION OF RESIDUAL GENERATORS

It is not feasible to implement and use all 15248 potential residual generators in the final FDI-system. A more attractive approach is instead to pick, from the set of all potential residual generators, a smaller set of residual generators with desired properties.

5.1 Requirements on Residual Generators

The desired properties of the sought set of residual generators are: (i) the set of residual generators should enable us to isolate all single faults from each other; (ii) a residual generator set of smaller cardinality is preferred before a larger one, given that the two sets have equal isolability properties; (iii) a residual generator based on an MSO set of smaller cardinality is preferred before a residual generator based on an MSO set of larger cardinality, given that the two sets have equal detectability and isolability properties. Properties (ii) and (iii) are mainly motivated by implementation aspects such as complexity and numerical issues.

We will base the selection of residual generators on quantitative, structural, properties of the MSO sets instead

of more qualitative or analytical properties on the actual residual generators. The latter is intractable since it require that residual generators are implemented, executed and evaluated, and also access to representative measurement data for all fault cases.

5.2 Formulation of the Selection Problem

We will now formulate the selection problem in terms of properties on a set of MSO sets. To do this, the notions of detectability and isolability are needed. Assuming that each fault occurs in only one equation, let e_{f_i} denote the equation in an equation set E containing fault f_i , for example $e_{\Delta\beta_{1,m1}} = e_{18}$, see Section 2. Note that if a fault f_j occurs in more than one equation, the fault f_j can be replaced with a new variable x_{f_j} in these equations, and the equation $x_{f_j} = f_j$ added to the equation set. This added equation will then be the only equation where f_j occurs. To proceed, let $(\cdot)^+$ denote an operator extracting the overdetermined part of a set of equations. According to Krysander and Frisk [2008], a fault f_i is *structurally detectable* in the equation set E if $e_{f_i} \in (E)^+$ and *structurally isolable* from fault f_j in the equation set E if $e_{f_i} \in (E \setminus e_{f_j})^+$.

By utilizing the *structure*, i.e. which unknown variables are contained in which equation, see e.g. Blanke et al. [2006], of the wind turbine model, the structural isolability properties of the model was calculated. All considered faults, see Section 2.2, can be (structurally) isolated from each other in the wind turbine model.

To this end, let \mathcal{M} denote the set of all MSO sets in the model, and F the set of considered faults. Let $f_i, f_j \in F$ and define the *isolation class* for (f_i, f_j) as

$$I_{f_i, f_j} = \left\{ m \in \mathcal{M} : e_{f_i} \in (m)^+ \wedge e_{f_j} \notin (m)^+ \right\}, \quad (9)$$

that is, the subset of \mathcal{M} in which fault f_i is structurally detectable and fault f_j is not. Further, let

$$\mathcal{I} = \{I_{f_i, f_j} : \forall (f_i, f_j) \in F \times F, f_i \neq f_j\} \quad (10)$$

denote the set of all isolation classes for full isolation of all faults in F .

To be able to satisfy the isolability property (i) stated above, we want to find a set $M \subseteq \mathcal{M}$ with a non-empty intersection with all isolation classes, that is $M \cap I_{f_i, f_j} \neq \emptyset$ for all $I_{f_i, f_j} \in \mathcal{I}$. To also satisfy the property (ii) we want to find an M so that $|M|$ is minimized, where the operator $|\cdot|$ returns the cardinality of a set. There are several possibilities for a metric that helps us find an M that satisfies property (iii). We opt for simplicity and have therefore chosen to minimize $\sum_{m \in M} |m|$. As an additional requirement, on top of (i), (ii), and (iii) above, we require that at least one residual generator can be constructed from every $m \in M$.

5.3 Solving the Selection Problem

In fact, the requested property on M implies that we should find a hitting set for \mathcal{I} . To also satisfy the requirement concerning the cardinality of M , a minimal hitting set (MHS) would be preferred. The MHS problem is known to be NP-hard, see e.g. Garey and Johnson [1979], but

there are several algorithms that gives an approximative solution, see for example Abreu and van Gemund [2009] and references therein.

We have chosen to compute an approximate solution to the problem with a greedy selection approach, see e.g. Black [2005]. To accomplish this, we need to specify a *utility function*, i.e., a function that evaluates the usefulness of a given MSO set, and also state the properties of a complete solution to the selection problem. Following the greedy selection approach, we add to the solution the MSO set with the largest utility until the solution is complete. Furthermore, we only add MSO sets from which at least one residual generator can be constructed.

Characterization of a Solution We will now characterize a complete solution to the selection problem for use in the selection algorithm. First, we define the *isolation class coverage* of a set of MSO sets $M \subseteq \mathcal{M}$ as

$$\Sigma_M = \{I_{f_i, f_j} \in \mathcal{I} : \exists m \in M, m \in I_{f_i, f_j}\}, \quad (11)$$

which states which of the isolation classes in \mathcal{I} that are covered by the MSO sets in M . The property (i) stated above, i.e., the isolation or hitting set property, can with the isolation class coverage notion be formulated as $\Sigma_M = \mathcal{I}$. This characterizes a complete solution of the selection problem.

Utility Function To evaluate a specific MSO set, we want to take into account the properties (ii) and (iii) above. We will use the utility function

$$\eta(m, \mathcal{I}) = \frac{|\Sigma_{\{m\}}|}{|\mathcal{I}|}, \quad (12)$$

reflecting how many of the isolation classes in \mathcal{I} that are covered by the MSO set m . Since we aim at covering all isolation classes with a minimum of MSO sets, property (ii), we want to pick an MSO set that maximizes this term. If there are several MSO sets with equal utility, the MSO set among these with smallest cardinality will be picked. In this way, property (iii) is taken into account.

5.4 The Selection Algorithm

The algorithm SELECTRESIDUALGENERATORS used for selecting residual generators by means of greedy selection is given below. Input to the algorithm is a set of MSO sets \mathcal{M} , and a set of isolation classes \mathcal{I} . The output is a set of MSO sets $M \subseteq \mathcal{M}$ and set of residual generators R . The function FINDCOMPUTATIONSEQUENCE, described in Svård and Nyberg [2010], is used to find a computation sequence given a just-determined set of equations.

```

function SELECTRESIDUALGENERATORS( $\mathcal{M}, \mathcal{I}$ )
   $M := \emptyset$ 
   $R := \emptyset$ 
  while  $\mathcal{I} \neq \emptyset$  do
     $H = \{\hat{m} \in \mathcal{M} : \hat{m} = \arg \max_{m \in \mathcal{M}} \eta(m, \mathcal{M})\}$ 
     $m^* := \arg \min_{m \in H} |m|$ 
     $x := \text{var}_X(m^*)$ 
     $r := \emptyset$ 
    for all  $e \in m^*$  do
       $m' := m^* \setminus \{e\}$ 
       $C := \text{FINDCOMPUTATIONSEQUENCE}(m', x)$ 
      if  $C \neq \emptyset$  then
         $r := r \cup \{(\mathcal{T}(C), e)\}$ 
      end if
    end for
  end while

```

```

if  $r \neq \emptyset$  then
   $M := M \cup \{m^*\}$ 
   $R := R \cup \{r\}$ 
end if
 $\mathcal{M} := \mathcal{M} \setminus \{m^*\}$ 
 $\mathcal{I} := \mathcal{I} \setminus \Sigma_{\{m^*\}}$ 
end while
end function

```

5.5 Resulting Set of Residual Generators

Both functions SELECTRESIDUALGENERATORS and FINDCOMPUTATIONSEQUENCE were implemented in MATLAB[®]. As computational tool, see Svård and Nyberg [2010], the algebraic equation solver MAPLE[®] was utilized, which allows symbolic solving of algebraic loops. To avoid implementation issues related to numerical differentiation and thereby make the final deployment of the FDI-system simpler, FINDCOMPUTATIONSEQUENCE was configured to return only computation sequences using integral causality. Moreover, also due to implementation issues, FINDCOMPUTATIONSEQUENCE was configured to prefer algebraic equations as residuals before differential equations, if possible.

The algorithm selected 16 MSO sets, i.e. $|M| = 16$, and $\sum_{m \in M} |m| = 61$. Of the 16 selected MSO sets, 6 contain algebraic equations only. The other 10 MSO sets contain both algebraic and differential equations. The total number of found residual generators is 34, that is, $|R| = 34$. Of these 34 residual generators, 18 are static and the remaining 16 are dynamic and consequently, due to the configuration of the algorithm, uses integral causality.

5.6 Fault Signature Matrix

Given an MSO set m its *fault signature* F_m , with respect to the faults in F , is defined as

$$F_m = \{f_i \in F : e_{f_i} \in m\}.$$

For instance, the fault signature of the MSO set $m_1 = \{e_{26}, e_{27}\} \subseteq \mathcal{M}$ is $F_{m_1} = \{\Delta\omega_{g, m1}, \Delta\omega_{g, m2}\}$. A convenient representation of the fault signature of a set of MSO sets $M = \{m_1, m_2, \dots, m_k\}$ with respect to F is the *fault signature matrix* (FSM) S with elements defined by

$$S_{ij} = \begin{cases} x, & \text{if } f_j \in F_{m_i}, m_i \in M \\ 0, & \text{else.} \end{cases}$$

The FSM for the 16 MSO sets on which the selected residual generators are based, is given in Table 1.

6. FAULT DETECTION AND ISOLATION

For fault detection and isolation, we construct diagnostic tests based on each of the 16 residuals. Since no assumptions are made regarding the nature of the faults that should be detected, nothing is known about faults temporal properties, size, rate of occurrence, etc. Hence, we may not be able to fully exploit the potential of some general method for change detection as for example the CUSUM-test, see e.g. Gustafsson [2000]. However, for most systems there are no-fault data available, either as real or simulated measurements. To take advantage of this fact, we base our diagnostic tests on a comparison between the estimated probability distributions of residuals evaluated with current and no-fault data.

Table 1. Fault Signature Matrix

	$\Delta\beta_1$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\omega_g$	$\Delta\tau_g$	$\Delta\beta_{1,m1}$	$\Delta\beta_{1,m2}$	$\Delta\beta_{2,m1}$	$\Delta\beta_{2,m2}$	$\Delta\beta_{3,m1}$	$\Delta\beta_{3,m2}$	$\Delta\omega_r,m1$	$\Delta\omega_r,m2$	$\Delta\omega_g,m1$	$\Delta\omega_g,m2$
$r_1 (m_1)$														x	x
$r_2 (m_2)$												x	x		
$r_3 (m_3)$										x	x				
$r_4 (m_4)$								x	x						
$r_5 (m_5)$						x	x								
$r_6 (m_8)$					x										
$r_7 (m_{11})$				x									x		x
$r_8 (m_{27})$			x								x				
$r_9 (m_{29})$		x							x						
$r_{10} (m_{31})$	x						x								
$r_{11} (m_7)$														x	
$r_{12} (m_6)$															x
$r_{13} (m_{14})$				x								x		x	
$r_{14} (m_{28})$			x							x					
$r_{15} (m_{30})$		x						x							
$r_{16} (m_{32})$	x					x									

6.1 Diagnostic Test Design

Let P^{NF} be an estimate of the distribution of a residual from no-fault data, and P an estimate of the distribution of the same residual from current data, both discretized in n bins. Then the Kullback-Leibler (K-L) divergence, Kullback and Leibler [1951], between P and P^{NF} is given by

$$D(P\|P^{NF}) = \sum_{j=1}^n P(j) \log \frac{P(j)}{P^{NF}(j)}. \quad (13)$$

To apply the K-L divergence for construction of a diagnostic test, we proceed as follows. Given a representative batch of no-fault data \mathcal{Z}^{NF} , i.e. in our case measurements of the known variables in Z , we run the set of residual generators and obtain a set of residuals. For each residual r_i , we then estimate its probability distribution and obtain P_i^{NF} , i.e. actually $P_i^{NF} \approx P(R_i|\mathcal{Z}^{NF})$ where R_i is a stochastic variable, discretized in n bins, representing residual r_i . This procedure can be done off-line.

On-line, we continuously estimate the distribution of r_i using a sliding window containing N samples of r_i . If we by P_i^t denote the estimated distribution of r_i calculated at time t , i.e. $P_i^t \approx P(R_i|\mathcal{Z}^t)$ where \mathcal{Z}^t denotes the batch of data in the sliding window at time t , the diagnostic test is designed as

$$T_i(t) = \begin{cases} 1, & \text{if } D(P_i^t\|P_i^{NF}) \geq J_i, \\ 0, & \text{else,} \end{cases} \quad (14)$$

where J_i is the threshold for alarm. The K-L divergence $D(P_i^t\|P_i^{NF})$ is referred to as the test quantity of the diagnostic test T_i .

To estimate a probability distribution, we create a normalized histogram with n bins for the data set from which the distribution should be estimated.

6.2 Fault Isolation Strategy

To obtain the total diagnosis statement from a set of alarming diagnostic tests, we simply match their fault signatures with the FSM given in Table 1. For example, if only T_{10} alarms, we look at the row corresponding to r_{10} and conclude that either fault $\Delta\beta_1$ or $\Delta\beta_{1,m2}$ are present.

If then also T_{16} alarms, we combine the row corresponding to r_{16} with the row corresponding to r_{10} and conclude that fault $\Delta\beta_1$ must be present.

To handle also multiple faults, we use the fault signatures in the original FSM in Table 1 to create an extended FSM with fault signatures also for multiple faults. This is done by column-wise OR-operations in the original FSM. For instance, the column in the FSM for the double fault $\Delta\omega_{g,m1} \wedge \Delta\omega_{g,m2}$ will get “x” in rows corresponding to r_1 , r_7 , r_{11} , r_{12} , and r_{13} and zeros elsewhere. In the fault isolation scheme, we first attempt to isolate all single faults using the original FSM in Table 1. If this does not succeed, we try to isolate double faults, and so forth.

Due to uncertainties as for example modeling errors and measurement noise, the power of diagnostic tests are not ideal for all faults. That is, the probability of detection given a certain fault is not always 1. To take this into account, the isolation scheme interprets an “x” in a certain row in Table 1 as if the test *may* respond if the corresponding fault occurs and consequently no conclusions are drawn if a test does not respond.

6.3 Implementation Details

The final FDI-system was implemented in SIMULINK[®] according to the structure in Figure 2.

The 16 residual generators were implemented as Embedded Matlab Functions (EMF) in which the code was automatically generated from the structures obtained from the algorithms FINDCOMPUTATIONSEQUENCE and FINDRESIDUALGENERATORS. The initial conditions for the states in the dynamic residual generators were derived from the corresponding sensor measurements, if available, otherwise set to zero. For instance, $\theta_\Delta(t_0) = 0$, $x_{\beta_{i1}}(t_0) = \frac{\beta_{i,m1}(t_0) + \beta_{i,m2}(t_0)}{2}$, and $\omega_g(t_0) = \frac{\omega_{g,m1}(t_0) + \omega_{g,m2}(t_0)}{2}$. This may cause transients in the residuals, but this is not considered a problem. The output from the residual generation block is a vector \mathbf{r} containing the 16 residuals.

The diagnostic tests were implemented as EMF’s in the fault detection block in Figure 2. The output from this block consist of the two signals `detectionResults` and `detectionDone`. The signal `detectionResults` is a binary vector where elements correspond to the result of the diagnostic tests T_i , $i = 1, 2, \dots, 16$. The signal `detectionDone` is simply set to one if any test has alarmed, and zero else.

The fault isolation scheme and the logic for performing first single fault isolation, then double fault isolation and so on, was also implemented as an EMF. The output from the fault isolation block consist of the signals `isolationDone` and `isolationResult`. The signal `isolationResult` contains the diagnosis statement and consists of a binary vector where elements correspond to the considered faults. The signal `isolationDone` is a binary signal set to one if fault isolation could be performed, and zero else. The signal `isolationDone` is only set to one if the diagnosis statement has been equal for t_i^{val} samples in a row.

6.4 Parameters

The number of bins n in the histograms used as distribution estimates, is a trade-off between detection time, noise sensitivity, and complexity, in terms of computational power and memory. A large n results in fast detection, but on the other hand also in increased sensitivity for noise. Also, a large n requires more memory and involves more computations, in comparison with a smaller n .

The size N of the sliding window used to batch data for creation of the histograms is a trade-off between detection performance, noise sensitivity, and complexity. A large N will give the K-L test quantity low-pass characteristics, resulting in a smoothed K-L test quantity. This makes it possible to detect small changes in the estimated distributions. On the other hand, a large N requires more memory. The choice of N is also related to the number of bins n in the histograms and vice versa, since a small N together with a large n , will result in a sparse histogram. Hence, the choices of N and n must match.

Preliminary investigations indicate however that the method is quite insensitive to the values of n and N if $15 \leq n \leq 50$ and $2000 \leq N \leq 6000$. A decent trade-off, taking this into account, but also the complexity issues discussed above, is $n = 20$ and $N = 3000$, which are the values used in the final FDI-system.

The choice of alarm thresholds J_i , $i = 1, 2, \dots, 16$, is a trade-off between detection time and the number of false detections. The higher the thresholds, the longer the detection time and the lower the rate of false alarms. The choice of alarm thresholds is related to the choices of n and N since both affect how sensitive a K-L test quantity is to noise, which in turn affects the rate of false detections. We aim at choosing the alarm thresholds so that the number of false detections is minimized, implying that the choice of J_i must match the choices of n and N .

The only parameter involved in the fault isolation is the isolation validation time t_I^{val} . It can be used to compensate for the fact that different diagnostic tests have different detection times. By choosing a large t_I^{val} , we decrease the probability of false isolation, but on the other hand increase the isolation time.

7. EVALUATION AND RESULTS

To evaluate the performance of the proposed FDI-system, we use the test definition described in Fogh Odgaard et al. [2009]. The test definition is based on measured wind data and a sequence of injected faults. The set of injected faults, their time of occurrence and description, is specified in Table 2. The sequence contains 5 sensor faults and 3 actuator faults. Note that the fault injected at 1000-1100 s is the double fault $\Delta\omega_{r,m2} \wedge \Delta\omega_{g,m2}$.

The no-fault distributions used in the evaluation were estimated from residual data stemming from 100 Monte Carlo simulations with no-fault data, i.e. inputs corresponding to the measured variables in Z . Each set of no-fault data was generated with the provided wind turbine model with different noise realizations. The alarm thresholds J_i , $i = 1, 2, \dots, 16$, were chosen in order to minimize the number of false detections. They were therefore computed as a

Table 2. Fault Sequence

Fault	Time (s)	Description
$\Delta\omega_{r,m2}$	1000 - 1100	$\omega_{r,m2} = 1.1\omega_{r,m2}$
$\Delta\omega_{g,m2}$	1000 - 1100	$\omega_{g,m2} = 0.9\omega_{g,m2}$
$\Delta\omega_{r,m1}$	1500 - 1600	$\omega_{r,m1} = 1.4 \text{ rad/s}$
$\Delta\beta_{1,m1}$	2000 - 2100	$\beta_{1,m1} = 5^\circ$
$\Delta\beta_{2,m2}$	2300 - 2400	$\beta_{2,m2} = 1.2\beta_{2,m2}$
$\Delta\beta_{3,m1}$	2600 - 2700	$\beta_{3,m1} = 10^\circ$
$\Delta\beta_2$	2900 - 3000	$\omega_n = \omega_{n2}, \zeta = \zeta_2$
$\Delta\beta_3$	3400 - 3500	$\omega_n = \omega_{n3}, \zeta = \zeta_3$
$\Delta\tau_g$	3800 - 3900	$\tau_g = \tau_g + 2000 \text{ Nm}$

Table 3. FDI Results. Time values in seconds.

	$\Delta\omega_{r,m2}$ $\Delta\omega_{g,m2}$	$\Delta\omega_{r,m1}$	$\Delta\beta_{1,m1}$	$\Delta\beta_{2,m2}$	$\Delta\beta_{3,m1}$	$\Delta\beta_2$	$\Delta\beta_3$	$\Delta\tau_g$
Req.	0.1	0.1	0.1	0.1	0.1	0.08	6	0.05
\bar{T}_D	0.040	0.16	0.058	4.30	0.069	51.57	18.1	7.94
T_D^{max}	0.04	0.27	0.07	6.10	0.07	51.88	19.05	7.98
T_D^{min}	0.03	0.06	0.05	0.40	0.06	50.57	16.37	7.90
\bar{T}_I	-	2.53	0.12	88.85	0.13	56.95	31.84	7.99
T_I^{max}	-	3.13	0.12	114.26	0.13	120.73	111.96	8.03
T_I^{min}	-	1.89	0.11	13.17	0.12	51.62	17.91	7.95
MD	0	0	0	0	0	0	0	0
FD	0	0	0	0	0	0	0	0

safety factor $\alpha = 1.1$ times the maximum value of the corresponding K-L test quantities from 100 simulations with no-fault data. The isolation validation time t_I^{val} was set to 4 samples.

7.1 Results and Analysis

By means of Monte Carlo simulations, the FDI-system was simulated 100 times with data from the provided wind turbine model set up according to the above described test sequence.

Based on the results from the 100 runs, the mean time of detection \bar{T}_D , maximum time of detection T_D^{max} , minimum time of detection T_D^{min} , mean time of isolation \bar{T}_I , minimum time of isolation T_I^{min} , the total number of missed detections MD, and the total number of false detections FD, for each of the faults in the test sequence, were computed. The results along with the requirements as specified in Fogh Odgaard et al. [2009] are shown in Table 3, where all time values are given in seconds.

According to the row corresponding to \bar{T}_D in Table 3, all faults in the test sequence could be detected. For faults $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$, $\Delta\beta_{1,m1}$, $\Delta\beta_{3,m1}$ detection requirements are met, by means of both \bar{T}_D and T_D^{max} .

All faults, except the double fault $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$ could also be isolated. However, the mean time of isolation, \bar{T}_I , for some faults, e.g. $\Delta\beta_{2,m2}$, is substantially longer than the corresponding mean time of detection. The main reason for this is that some tests respond slower to faults than other. As said, fault $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$ could not be isolated. In fact, this fault is not uniquely isolable with the isolation strategy described in Section 6.2 since the test response of fault $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m2}$ is a subset of the test response of fault $\Delta\omega_{g,m2} \wedge \Delta\omega_{r,m1}$, see Table 1. Both faults $\Delta\omega_{g,m2}$ and $\Delta\omega_{r,m2}$ are however contained in the diagnosis statement computed after the faults have been detected.

It seems like sensor faults, e.g., $\Delta\beta_{3,m1}$ tend to be easier to detect and isolate than actuator faults as for example $\Delta\tau_g$ and $\Delta\beta_2$. One possible explanation may be that actuator faults in general cause changes in dynamics, whose effects are attenuated by modeling errors, noise, etc.

As can be seen in the last two rows of Table 3, there are no missed or false detections in neither of the 100 test runs.

7.2 Case Study of Fault $\Delta\omega_{r,m1}$

To study in more detail how the FDI-system handles faults, we consider the sensor fault $\Delta\omega_{r,m1}$. The fault corresponds to a fixed value of 1.4 rad/s being measured by sensor $\omega_{r,m1}$ and occurs at time $t = 1500$ s. According to the FSM in Table 1, the residuals sensitive to fault $\Delta\omega_{r,m1}$ are r_2 and r_{13} . These residuals along with the corresponding K-L test quantities are shown in Figure 3. As can be seen, both the residuals and the test quantities respond distinctively to the fault.

To also illustrate the isolation procedure, we show in Figure 4 the entries in `detectionResult` corresponding to the diagnostic tests T_2 and T_{13} (top), the entries in signal `isolationResult` corresponding to faults $\Delta\omega_{r,m1}$ (middle) and $\Delta\omega_{r,m2}$ (bottom), and also the signal `isolationDone` (middle). As can be seen in Figure 4, the first test that reacts to the fault is T_2 . This occurs at $t = 1500.23$ s. Since T_2 is sensitive to both fault $\Delta\omega_{r,m1}$ and $\Delta\omega_{r,m2}$ and no other test has alarmed, the diagnosis statement is that either $\Delta\omega_{r,m1}$ or $\Delta\omega_{r,m2}$ may be present, and no fault can be isolated. At $t = 1502.55$ s, test T_{13} alarms. Test T_{13} is sensitive to faults $\Delta\omega_g$, $\Delta\omega_{r,m1}$, and $\Delta\omega_{r,m2}$, and the updated total diagnosis statement based on that both T_2 and T_{13} have alarmed thus becomes $\Delta\omega_{r,m1}$, see Table 1. This occurs at time $t = 1502.59$ s.

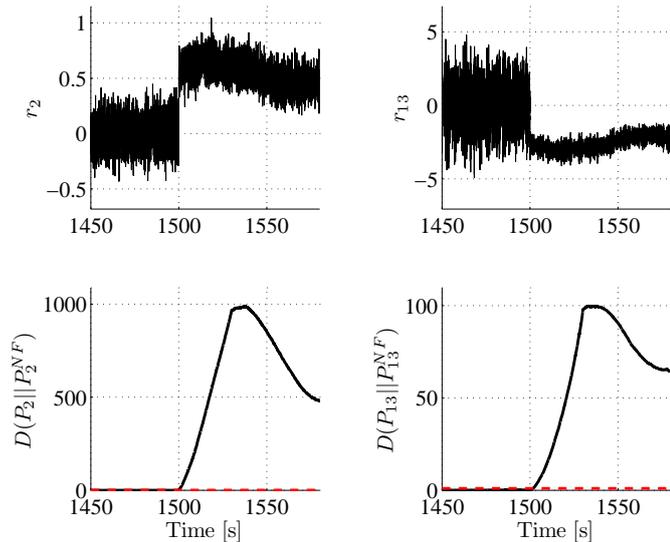


Fig. 3. Affected residuals r_2 (top-left) and r_{13} (top-right), and the corresponding K-L test quantities $D(P_2^t || P_2^{NF})$ (bottom-left) and $D(P_{13}^t || P_{13}^{NF})$ (bottom-right) at the time of occurrence of fault $\Delta\omega_{r,m1}$.

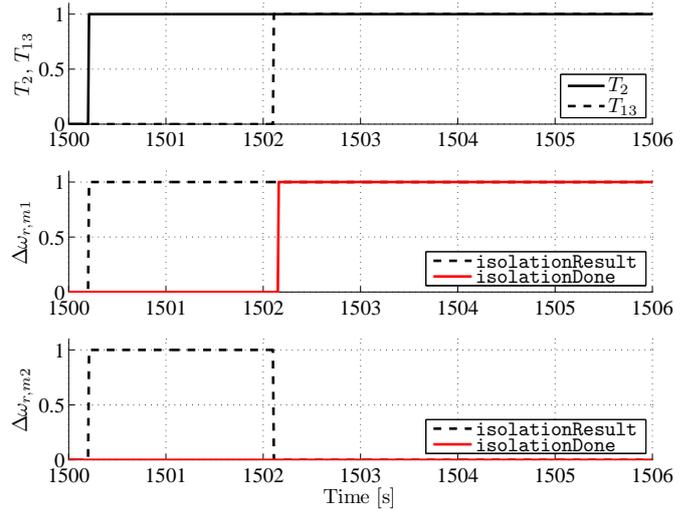


Fig. 4. Isolation procedure for fault $\Delta\omega_{r,m1}$. Top figure shows diagnostic tests T_2 and T_{13} . Middle and bottom figures show entries in `isolationResult` corresponding to faults $\Delta\omega_{r,m1}$ and $\Delta\omega_{r,m2}$, respectively.

8. CONCLUSIONS

We have proposed an FDI-system for the wind turbine benchmark designed by application of a generic automated design method, in which the number of required human decisions and assumptions are minimized. No specific adaptation of the method for the wind turbine benchmark was needed. The method contains in essence three steps: generation of potential residual generators; residual generator selection; and diagnostic test construction. The second and third step are based on novel ideas developed in this paper: a greedy selection algorithm for the second step, and a method based on the K-L divergence for the third step.

The performance of the proposed FDI-system has been evaluated using the pre-defined test sequence for the wind turbine benchmark. The FDI-system performs well; all faults in the test sequence were detected within feasible time and all faults, except a double fault, could be isolated shortly thereafter. In addition, there are no false or missed detections. A tailor-made, finely tuned, FDI-system for the benchmark would probably perform better. However, in relation to the required design effort, and that no specific adaptation or tuning of the method to the benchmark was done, the performance is good.

ACKNOWLEDGEMENTS

This work was supported by Scania CV AB, Södertälje, Sweden.

REFERENCES

- R. Abreu and A.J.C van Gemund. A low-cost approximate minimal hitting set algorithm and its application to model-based diagnosis. In V. Bilitko and J.C. Beck, editors, *Proceedings of the Eighth Symposium on Abstraction, Reformulation, and Approximation*, pages 2–9, Lake Arrowhead, California, USA, September 2009.
- P.E. Black. Greedy algorithm. *Dictionary of Algorithms and Data Structures* (online), U.S. National In-

- stitute of Standards and Technology, February 2005. <http://tinyurl.com/3x5zzpp>, Accessed: 2010-09-13.
- M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki. *Diagnosis and Fault-Tolerant Control*. Springer, second edition, 2006.
- 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.
- P. Fogh Odgaard, J. Stoustrup, and M. Kinnaert. Fault tolerant control of wind turbines – a benchmark model. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pages 155–160, Barcelona, Spain, 2009.
- M.R. Garey and D.S. Johnson. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.
- Fredrik Gustafsson. *Adaptive Filtering and Change Detection*. Wiley, July 2000.
- M. Krysander and E. Frisk. Sensor placement for fault diagnosis. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(6): 1398–1410, Nov. 2008.
- M. Krysander, J. Åslund, and M. Nyberg. An efficient algorithm for finding minimal over-constrained subsystems for model-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, 38(1):197–206, 2008.
- S. Kullback and R.A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.
- 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.
- 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.
- W.J. Rugh. *Linear System Theory*, chapter 13. Prentice Hall Information and System Sciences, 1996.
- M. Staroswiecki. *Fault Diagnosis and Fault Tolerant Control*, chapter Structural Analysis for Fault Detection and Isolation and for Fault Tolerant Control. Encyclopedia of Life Support Systems, Eolss Publishers, Oxford, UK, 2002.
- 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.
- 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. ISSN 1083-4427. doi: 10.1109/TSMCA.2010.2049993.
- 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.