Int. J. Appl. Math. Comput. Sci., 2012, Vol. 22, No. 1, 55–65
DOI: 10.2478/v10006-012-0004-y

TEST SIGNAL GENERATION FOR SERVICE DIAGNOSIS BASED ON LOCAL STRUCTURAL PROPERTIES

MICHAEL UNGERMANN [*,**], JAN LUNZE [**], DIETER SCHWARZMANN [*]

[*] Robert Bosch GmbH
Postfach 30 02 40, 70442 Stuttgart, Germany
e-mail: michael.ungermann@de.bosch.com

[**] Institute of Automation and Computer Control
Ruhr-Universität Bochum, Universitätsstrasse 150, 44780 Bochum, Germany
e-mail: lunze@atp.rub.de

The paper presents a new approach to the generation of test signals used in service diagnosis. The tests make it possible to isolate faults, which are isolable only if the system is brought into specific operating points. The basis for the test signal selection is a structure graph that represents the couplings among the external and internal signals of the system and the fault signals. Graph-theoretic methods are used to identify edges that disappear under certain operating conditions and prevent a fault from changing the system behavior at this operating point. These operating conditions are identified by validuals, which are indicators obtained during the graph-theoretic analysis. The test generation method is illustrated by a process engineering example.

Keywords: structural analysis, service diagnosis, active diagnosis, test generation.

1. Introduction

Service diagnosis is carried out in repair shops in order to identify a faulty component to be replaced. It differs from process diagnosis in the sense that the primary system objectives do not have to be met during the diagnostic process. Thus, specific signals can be applied to facilitate the isolation of specific faults. Increasing system complexity entails severe problems for this task. Strong interactions among components, nonlinear dynamics and a large number of possible fault scenarios impede the selection of dedicated tests for fault detection and isolation. Different faults may cause similar behavior of the faulty plant and, therefore, are difficult to be distinguished by a model-based diagnostic method.

This paper proposes a method for test signal generation that is based on determining operating points, in which the influence of specific faults on the input-output (I/O) behavior of the system under consideration is reduced. The method uses a structural model of the system that represents the relation among the external and internal signals and the faults. The structural representation is used to find particular operating regions, in which some of the interactions among these signals are not active and, hence, prevent a specific fault from influencing the I/O-behavior, whereas other faults remain in the cause effect chains from the inputs towards the outputs. If the system is brought into such an operating region, specific fault candidates can be excluded or validated.

In the literature, there are only a few publications that deal with the selection of dedicated test signals for service diagnosis of dynamical systems (e.g., Clever and Isermann, 2008). In a broader view, service diagnosis can be interpreted as a kind of *active diagnosis* in which the diagnostic unit may influence the input signal $\boldsymbol{u}(t)$ via the input generator by some signal $\boldsymbol{v}(t)$ (Fig. 1).

Active diagnosis has been investigated, for example, by Niemann and Poulsen (2005), and with the goal of maintaining system operability by Zhang (1989) as well as Campbell and Nikoukhah (2004). In contrast to these publications, the present paper is based on a structural representation of faulty dynamical systems, which results from a generalization of the structural modeling method described by Blanke *et al.* (2006). Like in the work of Krysander *et al.* (2008), it combines structural analysis with the goal of fault isolation in order to find permanent

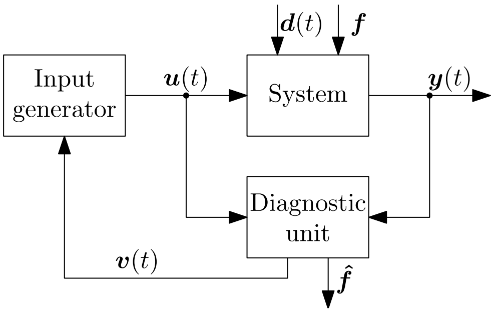

Fig. 1. Active diagnosis.

single faults.

The combination of active diagnosis and structural methods was introduced by Laursen *et al.* (2008) by investigating the isolability properties of hybrid systems in all their discrete operation modes. This method utilized the individual system structures that correspond to the different discrete modes of the hybrid system. Bayoudh *et al.* (2008; 2009) propose to abstract the continuous dynamics of hybrid systems to obtain purely discrete-event models and to apply discrete-event techniques for diagnosis. Finding inputs that allow refining the diagnostic result is reached by formulating the active diagnosis problem as a conditional planning problem.

The present paper deals with nonlinear continuous systems, which do not have *a priori* defined operation modes. A method for identifying a set of operating points, in which the continuous system has a particular system structure, is developed together with a method to steer the potentially faulty system into these operating points. The specific dynamical properties of the system that are associated with these operating points are then used to distinguish otherwise indistinguishable faults.

In an earlier paper of the authors (Ungermann *et al.*, 2010a), the more specific idea of excluding the influence of the fault variables from the structural model was used for the diagnosis of automotive systems. The present paper details the approach presented by Ungermann *et al.* (2010b), which is much more general in the sense that the focus is not on edges adjacent to fault variables, but on arbitrary edges in the structure graph. This corresponds to finding operating regions, in which the whole incidence matrix $\boldsymbol{M} = [\boldsymbol{M}_{\mathcal{X}}\ \boldsymbol{M}_{\mathcal{K}}\ \boldsymbol{M}_{\mathcal{F}}]$ of the graph defined by Ungermann *et al.* (2010a) changes.

The structure of the paper is as follows. Section 2 explains the analytical and structural models of dynamical systems used. Section 3 briefly reviews the idea of consistency-based diagnosis and structural analysis for fault diagnosis. Section 4 describes the main idea of searching for structural changes of the system and identifying the corresponding operating points. The method is summarized in Section 5 in two algorithms and applied in Section 6 to a process engineering example.

2. Modeling framework

2.1. System behavior. The system behavior $\mathcal{B}$ is the union of all I/O pairs allowed by a system. For static SISO systems, the behavior $\mathcal{B}$ may be represented by curves in the I/O space. For MIMO systems, the behavior is a map defined on the union of vector pairs $\{\boldsymbol{u}, \boldsymbol{y}\}$. As such a representation is not possible for dynamical systems, a set representation is chosen in the union of the vector function pairs $\{\boldsymbol{u}(t), \boldsymbol{y}(t)\}$ (Fig. 2). Obviously, a system's behavior $\mathcal{B}_{f_i}$ in the case of fault f_i is generally different from the behavior $\mathcal{B}_0$ in the nominal case.

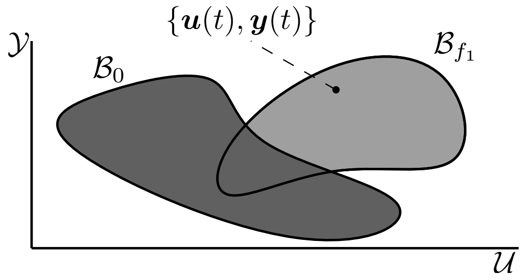

Fig. 2. System behavior $\mathcal{B}$.

2.2. Analytical model. When deriving an *analytical model* of a system, one describes the system behavior by a set of variables $z_j \in \mathcal{Z}$ and a set of constraints $c_i \in \mathcal{C}$ relating these variables to each other. For dynamical systems, constraints c_i include the relation between variables z and their time derivative $\dot{z}$.

Four types of variables are used to describe a faulty system: unknowns $x \in \mathcal{X}$, inputs $u \in \mathcal{U}$, outputs $y \in \mathcal{Y}$ and faults $f \in \mathcal{F}$:

$$\mathcal{Z} = \mathcal{X} \cup \mathcal{U} \cup \mathcal{Y} \cup \mathcal{F}. \tag{1}$$

One writes

$$\mathcal{K} = \mathcal{U} \cup \mathcal{Y} \tag{2}$$

for the known variables. Constraints are represented in the form

$$c_i:\quad 0 = h_i(x, u, y, f), \tag{3}$$

where h_i is a scalar function. The mapping

$$\text{var}:\ 2^{\mathcal{C}} \longrightarrow 2^{\mathcal{Z}} \tag{4}$$

associates a set of constraints with the set of variables occurring in these constraints.

An operating point is a tuple $(x \in \mathcal{X}, u \in \mathcal{U}, y \in \mathcal{Y})$ that does not contradict the set of constraints $\mathcal{C}$. A set of operating points is called the operating region.

The set $\mathcal{C}$ of constraints contains a qualitative fault model with f_i signifying a fault value. By convention, $f_i = 0\ \forall i$ holds in the nominal case.

2.3. Structural model. The *structure graph* is a qualitative representation of the physical couplings among the inputs, the internal variables (e.g., state variables) and the outputs of a dynamical system. In the following, the idea of this representation is briefly reviewed (Blanke *et al.*, 2006).

The structure graph is a bipartite graph

$$G = (\mathcal{Z} \cup \mathcal{C}, \mathcal{E}) \tag{5}$$

with two kinds of vertices that represent the variables ($\mathcal{Z}$) or the constraints ($\mathcal{C}$), respectively. If the variable z_j appears in the constraint c_i, there is an undirected edge $e \in \mathcal{E}$ between the vertex c_i and the vertex z_j in the graph. The graph can be represented by its incidence matrix $\boldsymbol{M}$ with the *ij*-th element m_{ij} equal to "1" if there is an edge between c_i and z_j:

$$m_{ij} = \begin{cases} 1, & z_j \in \mathrm{var}(\{c_i\}) \\ 0, & z_j \notin \mathrm{var}(\{c_i\}). \end{cases} \tag{6}$$

By analogy to the partition (1) and (2) of the variables, the incidence matrix can be partitioned as $\boldsymbol{M} = [\boldsymbol{M}_{\mathcal{X}}\ \boldsymbol{M}_{\mathcal{K}}\ \boldsymbol{M}_{\mathcal{F}}]$.

An edge between a variable z_j and a constraint c_i does not mean that z_j influences the manifold defined by c_i at all operating points. On the contrary, the exceptional cases where such influences break down are the main basis for the construction of tests in Section 4.

3. Diagnosis

3.1. Behavior and consistency. The idea of consistency-based diagnosis is to check whether or not a measured I/O pair $\{\boldsymbol{u}(t), \boldsymbol{y}(t)\}$ is contained in the system's nominal behavior $\mathcal{B}_0$. In model-based diagnosis, the behavior $\mathcal{B}_0$ is described by a set $\mathcal{C}$ of constraints which is the union of the equations of the system's physical model. If the measured I/O pair $\{\boldsymbol{u}(t), \boldsymbol{y}(t)\}$ does not belong to the behavior $\mathcal{B}_0$, it contradicts the model $\mathcal{C}$ and the system is known to be faulty (fault detection). Similarly, if the I/O pair contradicts the model $\mathcal{C}_{f_i}$, which has been set-up for the fault f_i, and the single fault assumption holds, the system is known not to be subject to the fault f_i (fault identification).

3.2. Structural analysis. The I/O pair can be inconsistent with the set $\mathcal{C}$ of constraints only if the model is over-constrained. The analysis of the structure graph is a tool to find over-constrained subsets $\mathcal{C}^+ \subseteq \mathcal{C}$ of the constraint set $\mathcal{C}$ according to Dulmage and Mendelsohn (1958), as it will be reviewed in the following.

A *matching* is a subset of edges $\mathcal{M} \subseteq \mathcal{E}$ of the structure graph with the property that no two edges in $\mathcal{M}$ have a common vertex. A matching $\mathcal{M}$ on the structure graph is called complete with respect to the unknown variables if there are as many edges in the set $\mathcal{M}$ as unknown variables in $\mathcal{C}$: $|\mathcal{M}| = |\mathcal{X} \cap \mathrm{var}(\mathcal{C})|$. A matching is said to be complete with respect to the constraints if there are as many edges in $\mathcal{M}$ as constraints in $\mathcal{C}$: $|\mathcal{M}| = |\mathcal{C}|$.

According to Blanke *et al.* (2006), a set $\mathcal{C}_\mathrm{D} \subseteq \mathcal{C}$ of constraints is called *structurally just-constrained* if there is a matching on its graph that is complete with respect to the unknown variables and complete with respect to the constraints. In that case, the set $\mathcal{C}_\mathrm{D}$ is marked by a superscript 0 and denoted by $\mathcal{C}_\mathrm{D}^0$.

If there is a matching on the graph that is complete with respect to the unknown variables, but no matching exists that is complete with respect to the constraints, $\mathcal{C}_\mathrm{D}$ is said to be *structurally over-constrained*. In that case, it is marked by a superscript + and denoted by $\mathcal{C}_\mathrm{D}^+$.

According to Krysander *et al.* (2008), an over-constrained set $\mathcal{C}_\mathrm{D}^+ \subseteq \mathcal{C}$ is called *minimal structurally over-constrained* if no proper subset of $\mathcal{C}_\mathrm{D}^+$ is structurally over-constrained. In the following, $\mathcal{C}_\mathrm{MSO}^+$ is used to denote such a set of constraints.

A fault f_i is said to be *structurally detectable* if there is a set $\mathcal{C}_\mathrm{MSO}^+$ that is influenced by the variable f_i:

$$\exists \mathcal{C}_\mathrm{MSO}^+ : f_i \in \mathrm{var}(\mathcal{C}_\mathrm{MSO}^+). \tag{7}$$

Two faults f_i and f_j are called *structurally isolable* if there is a set $\mathcal{C}_\mathrm{MSO}^+$ that is influenced by f_i but not by f_j:

$$\exists \mathcal{C}_\mathrm{MSO}^+ : f_i \in \mathrm{var}(\mathcal{C}_\mathrm{MSO}^+) \wedge f_j \notin \mathrm{var}(\mathcal{C}_\mathrm{MSO}^+). \tag{8}$$

The properties of structural detectability and structural isolability can be tested by means of conditions on the incidence matrix $\boldsymbol{M}$.

4. Local structures

An important observation is the fact that, for particular values of some of the variables, other variables may not have any impact on a constraint, although they appear in this constraint. This fact can be seen in the simple example

$$\begin{aligned} c_1 :&\quad h_1 = x_1(t) - y_1(t) + f_1, \\ c_2 :&\quad h_2 = x_2(t) - u_2(t) + f_2, \\ c_3 :&\quad h_3 = x_1(t)x_2(t) - u_1(t) + f_3, \end{aligned} \tag{9}$$

which will be used for illustration purposes in the following. For $x_1 = 0$, the constraint c_3 does not depend upon the variable x_2. Therefore, in the structural representation of the constraint set $\mathcal{C} = \{c_1, c_2, c_3\}$ under the assumption that $x_1 = 0$ holds, there is no edge between the constraint-vertex c_3 and the variable-vertex x_2.

More generally, for a constraint c_i in which the elements of the set

$$\mathcal{Z}_i = \mathrm{var}(\{c_i\}) \tag{10}$$

appear as variables, it is possible to obtain a condition $c_{\mathrm{Elim},k}$ on the variables in the set $\mathcal{Z}_i$ under which the constraint c_i can be simplified such that only a subset $\mathcal{Z}_p \subset \mathcal{Z}_i$ of variables actually appears in the simplified version of c_i. Such conditions $c_{\mathrm{Elim},k}$ define operating regions on the set $\mathcal{Z}_i$ of variables.

For algebraic constraints, a constraint $c_{\mathrm{Elim},k}$ defining an operating region can be found by derivation of c_i with respect to a variable $z_j \in \mathcal{Z}_i$:

$$c_{\mathrm{Elim},k}: \quad h_{\mathrm{Elim},k} = \frac{\mathrm{d}}{\mathrm{d}z_j} h_i = 0. \tag{11}$$

If the system is restricted to the operating region defined by (11), the variable z_j does not influence the constraint c_i. Then, the edge e_k between c_i and z_j can be removed from the structure graph, which is the same as setting $m_{ij} = 0$ in the incidence matrix. For the example (9), one obtains

$$c_{\mathrm{Elim},1}: \quad h_{\mathrm{Elim},1} = \frac{\mathrm{d}}{\mathrm{d}x_2} h_3 = x_1(t) = 0, \tag{12}$$

and one can remove the edge e_1 between c_3 and x_2 in the structure graph.

If not only a single edge e_k but a set of edges $\mathcal{E}_p$ can be eliminated from the structure graph, one finds a set of constraints $\mathcal{C}_{\mathrm{Elim},p}$ defining an operating region under which this is the case. The resulting structure graph is called the *local structure* because it does not describe the interaction of the variables in all the operating points but merely in a particular operating region. The local structure graph is denoted by

$$G_p = (\mathcal{Z} \cup \mathcal{C}, \mathcal{E} \setminus \mathcal{E}_p). \tag{13}$$

The graph G_p has the incidence matrix $\boldsymbol{M}_p$. It is valid only in the operating region defined by (11). As the diagnosability properties of a dynamical system depend upon the system structure, they may change with the structure (Laursen *et al.*, 2008). In the following section, an approach to construct tests using local system structures is presented.

5. Test generation

5.1. Concept. The tests to be selected should isolate faults from each other, which under normal operating conditions are not distinguishable. The main idea is to bring the system into a particular operating point where one fault, say the fault f_i, has an influence on a residual generator whereas another fault f_j does not have any effect. Then, the evaluation of the residual shows whether or not the fault f_i has occurred.

This particular operating region can be associated with a local structure. This local structure is used to determine residual generators, which are then called *local residual generators*. In the best case, only a single fault f_i remains in the set of fault variables that may have an impact on a local residual generator. Then, an analysis of the system behavior at this operating point by checking whether or not the local residual is zero shows whether the fault f_i is present.

The test selection method consists of two steps. First, edges of the structure graph that may vanish are determined together with the corresponding condition (11). Second, the test that brings the system at this operating point is selected. Both steps are described in the preceding paragraphs.

5.2. Determining potential structures. The following algorithm computes all edges that may vanish in particular operating regions. The main idea is to check for each edge in the structure graph whether or not it may vanish. Then, a condition on the other known and unknown variables, which are adjacent to the concerned constraint, is checked for contradiction with the nominal system and stored. Conditions on fault variables are not stored, because these variables cannot be influenced by changing the operating point.

Algorithm 1. [Find operating regions]

```
Given:    The analytical model $\mathcal{C}$ which describes a system's nominal behavior and its behavior
          in presence of the faults $f_i \in \mathcal{F}$.
Init:     $\mathcal{E}_{\mathrm{Elim}} = \emptyset$, $\mathcal{C}_{\mathrm{Reg}} = \emptyset$.
Step 1:   Compute the structure graph $G = (\mathcal{Z} \cup \mathcal{C}, \mathcal{E})$ of the system.
Step 2:   For each edge $e_k \in \mathcal{E}$ in the structure graph represented by a nonzero $m_{ij}$, apply
          Eqn. (11) to compute a constraint $c_{\mathrm{Elim},k}$ under which edge $e_k$ disappears from the structure graph.
Step 2a:  If $c_{\mathrm{Elim},k}$ contradicts $\mathcal{C}$, do not add $e_k$ to $\mathcal{E}_{\mathrm{Elim}}$.
Step 2b:  Otherwise, add $e_k$ to $\mathcal{E}_{\mathrm{Elim}}$ and $c_{\mathrm{Elim},k}$ to $\mathcal{C}_{\mathrm{Reg}}$.
Result:   A set of edges $\mathcal{E}_{\mathrm{Elim}}$ that can be eliminated and a set of constraints $\mathcal{C}_{\mathrm{Reg}}$
          defining the corresponding operating region.
```

If only edges adjacent to fault variables are considered, the presented approach is similar to the fault hiding method delivered by Ungermann *et al.* (2010a). However, the algorithm above yields more general operating conditions because it is not restricted to such edges.

5.3. Check for the operating region. If one wants to use local residuals for diagnosis, one needs to check that the system is in the operating region in which a particular local structure holds, and thus a local residual can be applied. This test boils down to the decision whether the system's variables satisfy a constraint $c_{\mathrm{Elim},k} \in \mathcal{C}_{\mathrm{Reg}}$ under which an edge e_k disappears. One, therefore, defines the validual as follows.

Definition 1. The signal $v_k(t)$ is called a *validual* if the relation $v_k(t) = 0$ indicates that the fault free system is in the operating region where the edge e_k disappears from the structure graph G.

The name *validual* was chosen for two reasons: firstly, because $v_k(t) = 0$ validates that the system is in a particular operating region, and secondly, because the procedure to compute it is similar to the one used for determining a residual. Like residuals, a validual is computed from the system input and output signals. It can be obtained by structural analysis as explained in the following.

The operating region defined by $c_{\mathrm{Elim},k} \in \mathcal{C}_{\mathrm{Reg}}$ under which the edge e_k disappears from the structure graph is known from Algorithm 1. It generally contains unknown variables:

$$\mathcal{X} \cap \mathrm{var}(\{c_{\mathrm{Elim},k}\}) \neq \emptyset. \tag{14}$$

In order to decide whether a fault free system is in an operating region described by $c_{\mathrm{Elim},k}$, one needs to conclude from known variables on the unknowns on the left-hand side of Eqn. (14). It is therefore necessary to find a set of constraints that is just-constrained with respect to the unknowns belonging to the set (14), because such a set of constraints allows computing the unknown variables. This is the case if a subset $\mathcal{C}^0_{\mathrm{Ste},k} \subseteq \mathcal{C}$ with the property

$$\mathcal{X} \cap \mathrm{var}(\{c_{\mathrm{Elim},k}\}) \subseteq \mathcal{X} \cap \mathrm{var}(\mathcal{C}^0_{\mathrm{Ste},k}) \tag{15}$$

can be found. Eliminating these unknowns from the constraint set

$$\mathcal{C}^+_{\mathrm{val},k} = \mathcal{C}^0_{\mathrm{Ste},k} \cup \{c_{\mathrm{Elim},k}\}, \tag{16}$$

one obtains a condition on the known variables occurring in the set

$$\mathcal{K} \cap \mathrm{var}(\mathcal{C}^+_{\mathrm{val},k}). \tag{17}$$

This condition is unique if there is a complete matching that is causal with respect to the unknowns (for the causality definition, cf. the work of Blanke *et al.* (2006)) on the structure graph of $\mathcal{C}^0_{\mathrm{Ste},k}$. Then, by assuming all fault variables to be zero, one can eliminate all unknowns from $\mathcal{C}^+_{\mathrm{val},k}$ by computing them with $\mathcal{C}^0_{\mathrm{Ste},k}$ and injecting the result in $c_{\mathrm{Elim},k}$. Then the resulting expression becomes a validual $v_k(t)$.

For the edge e_1 in the example in (9), one finds the constraint set $\mathcal{C}^+_{\mathrm{val},1} = \{c_1, c_{\mathrm{Elim},1}\}$ and the corresponding validual $v_1(t) = y_1(t)$. In Section 6.3, the above procedure is exemplified with the Eqns. (30)–(34).

The constraint $c_{\mathrm{Elim},k}$ is satisfied in the nominal case if $\mathcal{C}^+_{\mathrm{val},k}$ is satisfied, which is the case if $v_k(t) = 0$ holds. In analogy to the residual generator this procedure is called the *validual generator* in the following.

If the validual is zero and there is no fault, the system is in the operating region that corresponds to the structure

$$G_k = \{\mathcal{Z} \cup \mathcal{C}, \mathcal{E} \setminus e_k\}. \tag{18}$$

For reasons of fault isolability (see Section 5.7), one wants the cardinality $|\mathcal{C}^0_{\mathrm{Ste},k}|$ to be as small as possible. This problem is solved by determining the smallest over-constrained subset $\mathcal{C}^+_{\mathrm{val},k}$ of constraints from

$$\mathcal{C} \cup \{c_{\mathrm{Elim},k}\} \tag{19}$$

that contains $c_{\mathrm{Elim},k}$. In the work of Krysander *et al.* (2008), a subset of equations no proper subset of which is structurally over-constrained is called *Minimal Structurally Over-constrained* (MSO). If a set of constraints is MSO and contains $c_{\mathrm{Elim},k}$, it is the smallest possible set of constraints satisfying the above properties. Hence, the problem of identifying conditions on known variables that allow us to guarantee a particular structure is transformed into the well known problem of determining MSOs. The following algorithm sums up this method.

Algorithm 2. [Find constraints for a validual]

```
Given:   $\mathcal{C}$, $c_{\mathrm{Elim},k} \in \mathcal{C}_{\mathrm{Reg}}$.
Step 1:  Compute all MSOs $\mathcal{C}^+_{\mathrm{MSO},l}$ of $\mathcal{C} \cup \{c_{\mathrm{Elim},k}\}$.
Step 2:  Eliminate all MSOs that do not contain $c_{\mathrm{Elim},k}$ .
Step 3:  Choose the smallest MSO: $\mathcal{C}^+_{\mathrm{val},k} = \mathcal{C}^+_{\mathrm{MSO},l}$
         for which there is a complete causal matching of the unknowns on the structure graph
         of $\mathcal{C}^0_{\mathrm{Ste},k} = \mathcal{C}^+_{\mathrm{val},k} \setminus c_{\mathrm{Elim},k}$.
Result:  $\mathcal{C}^+_{\mathrm{val},k}$.
```

By repeating this algorithm for all $c_{\mathrm{Elim},k} \in \mathcal{C}_{\mathrm{Reg}}$, one can compute a total of $|\mathcal{C}_{\mathrm{Reg}}|$ different validuals, each corresponding to an eliminated edge. With different combinations of validuals being zero, a total of $2^{|\mathcal{E}_{\mathrm{Elim}}|}$ local structures G_p can be generated. Some of them might not be realizable, because not all combinations of validuals can be simultaneously zero due to contradicting constraints in the set $\mathcal{C}_{\mathrm{Reg}}$.

5.4. Computing local residual generators. A local residual $r(t)$ is a signal that is computed from the system's inputs and outputs. It is always zero in the fault free case and the system is restricted to a specific operating region. If the system is in this operating region and the signal $r(t)$ is nonzero, one can infer on the presence of a fault. Such an operating region is given by the one that is associated with a particular local structure. Because structural fault isolability properties are determined with the structure graph, they may be different for different local structures, and therefore some local structures may be better for fault isolation than others. If an advantageous local structure G_p out of the local structures that can be reached by eliminating the edges $\mathcal{E}_p \subseteq \mathcal{E}_{\mathrm{Elim}}$ is selected, structural analysis can be applied to G_p in order to compute all MSOs $\mathcal{C}^+_{\mathrm{MSO,res},s}$ for this local system structure. A local residual $r_s(t)$ can then be computed by eliminating all unknown variables from the set $\mathcal{C}^+_{\mathrm{MSO,res},s}$ by using the description $\mathcal{C}_{\mathrm{Elim},p}$ of the corresponding operating region if necessary.

For the example (9) and the operating region (12), the above procedure yields $\mathcal{C}^+_{\mathrm{MSO,res},1} = \{c_1, c_3\}$, which leads to the local residual $r_1(t) = -u_1(t)$.

5.5. Test construction. The local residuals $r_s(t)$ determined with $\mathcal{C}^+_{\mathrm{MSO,res},s}$ can only be used for diagnosis if the system is in an operating region where the local structure graph $G_p = (\mathcal{Z} \cup \mathcal{C}, \mathcal{E} \setminus \mathcal{E}_p)$ holds. This is the case if all edges $e_k \in \mathcal{E}_p$ disappear from G due to the system's inputs and outputs satisfying the corresponding constraint sets $\mathcal{C}^+_{\mathrm{val},k}$. These constraint sets are satisfied if all the validuals $v_k(t), k : e_k \in \mathcal{E}_p$ are zero.

A block diagram of the test principle, for an operating region in which the system's local structure G_p is valid, is shown in Fig. 3. Assume that two edges were eliminated from G in order to get the reduced structure G_p. For each edge, a validual $v_k(t)$ exists. If both validuals are zero, the local structure holds and the local residuals that were obtained from this local structure become valid for fault diagnosis. Then, the current values $r_1(t)$ and $r_2(t)$ of the local residuals are used in a decision logic to conclude which fault is present.

5.6. Active diagnosis. The vector $\boldsymbol{v}(t)$ in Fig. 1 is formed by the validuals $v_k(t)$ being the influence of the diagnostic unit on the input generator. The input generator has to compute an input signal $\boldsymbol{u}(t)$ that makes the relation $\boldsymbol{v}(t) = 0$ hold. Three cases of input generators can be distinguished:

- If the known variables in $\mathcal{K} \cap \mathrm{var}(\mathcal{C}^+_{\mathrm{val},k})$ are inputs, one can force the signals $v_k(t)$ to become zero by applying inputs satisfying $\mathcal{C}^+_{\mathrm{val},k}$. The input generator is a feed-forward controller.

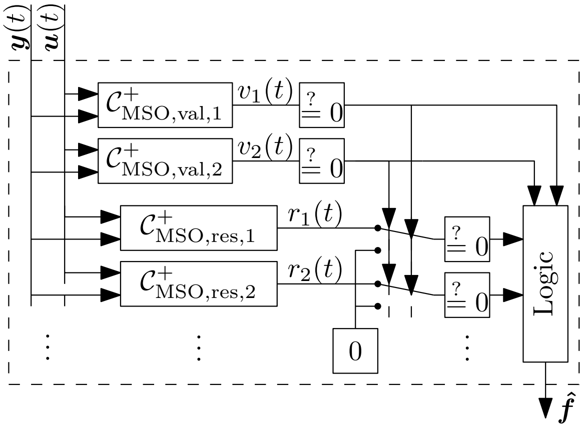

Fig. 3. Testing principle.

- If the set $\mathcal{K} \cap \mathrm{var}(\mathcal{C}^+_{\mathrm{val},k})$ contains only outputs, $v_k(t) = 0$ may be reached by applying a controller that makes the signal $v_k(t)$ vanish. The input generator is a feedback controller.

- If inputs and outputs appear in $\mathcal{C}^+_{\mathrm{val},k}$, one can solve the set of constraints for the inputs, thus obtaining a part of a control law steering the system in the required operating region. For the remaining input variables, signals have to be chosen that steer the system's output in a way satisfying $\mathcal{C}^+_{\mathrm{val},k}$.

5.7. Meaning of a nonzero local residual. Every local residual generator is based on an MSO $\mathcal{C}^+_{\mathrm{MSO,res},s}$ that is calculated with the local system structure, which is valid in the selected operating region. Assuming a single fault scenario and the validual being zero, the corresponding local residual may be nonzero for two reasons:

1. One of the faults in $\mathcal{F} \cap \mathrm{var}(\mathcal{C}^+_{\mathrm{MSO,res},s})$ is present (the system is in the particular operating region).

2. One of the faults in $\mathcal{F} \cap \mathrm{var}(\mathcal{C}^+_{\mathrm{val},k})$ is present (the system is not in the particular operating region).

Therefore, one can conclude that if the validual $v_k(t)$ is zero and the local residual $r_s(t)$ is nonzero, one of the faults in the set

$$(\mathcal{F} \cap \mathrm{var}(\mathcal{C}^+_{\mathrm{val},k})) \cup (\mathcal{F} \cap \mathrm{var}(\mathcal{C}^+_{\mathrm{MSO,res},s})) \qquad (20)$$

is present. Similar reasoning holds for structures in which more than one edge is eliminated.

For a test for the example system (9), which consists of the validual $v_1(t)$ and the local residual $r_1(t)$, the set (20) is $\{f_1, f_3\}$. Note that, for this example, standard approaches yield only one residual which is sensitive to the faults $\{f_1, f_2, f_3\}$. The new test therefore exhibits better isolability properties than the standard approach.

A particularly interesting case for fault isolation is given if the condition

$$\mathcal{F} \cap \mathrm{var}(\mathcal{C}^+_{\mathrm{val},k}) = \emptyset \tag{21}$$

holds. In that case, the validual $v_k(t)$ obtained from $\mathcal{C}^+_{\mathrm{val},k}$ is not influenced by any fault. Then, a fault may not lead to $v_k(t) = 0$ although the system is not in the operating region defined by $c_{\mathrm{Elim},k}$. Therefore, if the property (21) is satisfied, and if the relation $v_k(t) = 0$ holds, one can be sure that the system is actually in the corresponding operating region. Similarly, in the work of Laursen *et al.* (2008) it is assumed that the property (21) holds when steering a hybrid system in its discrete modes for fault isolation purposes.

5.8. Special case: Fault hiding. Some input signals may lead to particular validuals being zero thus signifying that all edges between a fault variable and the remaining graph disappear. In that case, the present I/O pair is independent of the presence of the fault not only contained in the faulty behavior but also in the nominal behavior. The I/O pair is not changed by the presence of the fault, and the fault is said to be hidden. This approach was originally introduced by Riggins and Ribbens (1997) for linear systems on the basis of state-space models, and for nonlinear systems by Ungermann *et al.* (2010a) based on structural models.

Figure 4 illustrates this situation. The nominal system behavior and the system behavior for two different faults f_1 and f_2 are shown. If the inputs $\boldsymbol{u}_{\mathrm{A}}(t)$ and $\boldsymbol{u}_{\mathrm{B}}(t)$ are applied, either f_1 or f_2 is hidden. Consecutively applying $\boldsymbol{u}_{\mathrm{A}}(t)$ and $\boldsymbol{u}_{\mathrm{B}}(t)$ allows us to distinguish between the faults f_1 and f_2.

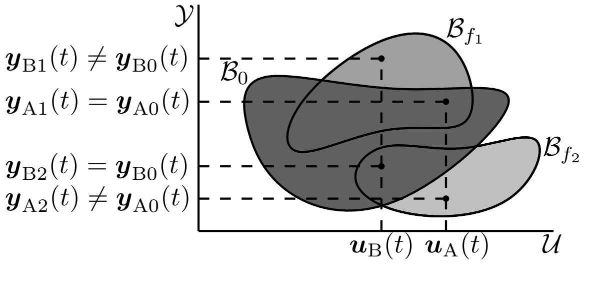
$\mathcal{Y}$
$\boldsymbol{y}_{\mathrm{B}1}(t) \neq \boldsymbol{y}_{\mathrm{B}0}(t)$
$\boldsymbol{y}_{\mathrm{A}1}(t) = \boldsymbol{y}_{\mathrm{A}0}(t)$
$\boldsymbol{y}_{\mathrm{B}2}(t) = \boldsymbol{y}_{\mathrm{B}0}(t)$
$\boldsymbol{y}_{\mathrm{A}2}(t) \neq \boldsymbol{y}_{\mathrm{A}0}(t)$
$\mathcal{B}_0$
$\mathcal{B}_{f_1}$
$\mathcal{B}_{f_2}$
$\boldsymbol{u}_{\mathrm{B}}(t)$
$\boldsymbol{u}_{\mathrm{A}}(t)$
$\mathcal{U}$

Fig. 4. Fault hiding.

6. Example

In this section, the new approach for test generation is applied to a process engineering example.

6.1. System description and model. The system is shown in Fig. 5. Two inflows $q_{\mathrm{hot}}(t)$ and $q_{\mathrm{cold}}(t)$ of water of different temperatures T_{hot} and $T_{\mathrm{cold}}(t)$ are controlled by two valves $u_1(t)$ and $u_2(t)$. The water mixes in the tank, where the fluid level $L(t)$ and the fluid temperature $T(t)$ are measured as outputs $y_1(t)$ or $y_2(t)$, respectively. Two pipes at the bottom and in the middle of the tank allow water to flow out of the tank ($q_{\mathrm{mid}}(t)$ and $q_{\mathrm{bot}}(t)$) according to Torriceli's law. The temperature T_{hot}, the area A of the cylindric tank and the effective area of the pipes a are parameters. The temperature $T_{\mathrm{cold}}(t)$ is unknown and may vary in time. Faults that may occur in the system are listed in Tab. 1.

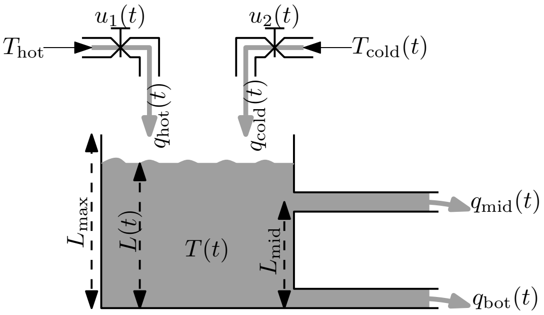

Fig. 5. Tank.

Table 1. Faults in the system.

f_i	Fault
f_1, f_2	Leaking valve 1 / valve 2
f_3	Heating broken (wrong T_{hot})
f_4	Middle-pipe clogged
f_5	Bottom-pipe clogged
f_6	Offset of level sensor
f_7	Offset of temperature sensor

The equations governing the system's behavior are given below. Note that only a *qualitative model* of the faults is used—no values are introduced for the corresponding fault variables. This means that only the *way* a fault influences the system behavior is described by the model, but the severity of the faults is not represented.

$$c_1 : h_1 = q_{\mathrm{hot}}(t) - u_1(t) - f_1,$$

$$c_2 : h_2 = q_{\mathrm{cold}}(t) - u_2(t) - f_2,$$

$$c_3 : h_3 = A\dot{L}(t) - q_{\mathrm{hot}}(t) - q_{\mathrm{cold}}(t) + q_{\mathrm{mid}}(t) + q_{\mathrm{bot}}(t),$$

$$c_4 : h_4 = q_{\mathrm{bot}}(t) - (1 - f_5)a\sqrt{L(t)},$$

$$c_5 : h_5 = q_{\mathrm{mid}}(t) - \begin{cases} (1 - f_4)a\sqrt{L(t) - L_{\mathrm{mid}}}, & L(t) > L_{\mathrm{mid}}, \\ 0, & \text{otherwise}, \end{cases}$$

$$\begin{aligned} c_6 : h_6 = {} & A(\dot{T}(t)L(t) + T(t)\dot{L}(t)) + T(t)q_{\mathrm{bot}}(t) \\ & + T(t)q_{\mathrm{mid}}(t) - T_{\mathrm{cold}}(t)q_{\mathrm{cold}}(t) \\ & - (1 + f_3)T_{\mathrm{hot}}q_{\mathrm{hot}}(t), \end{aligned}$$

$$c_7 : h_7 = L(t) - y_1(t) + f_6,$$
$$c_8 : h_8 = T(t) - y_2(t) + f_7,$$
$$d_1 : h_{\mathrm{d1}} = \dot{L}(t) - \frac{\mathrm{d}}{\mathrm{d}t}L(t),$$
$$d_2 : h_{\mathrm{d2}} = \dot{T}(t) - \frac{\mathrm{d}}{\mathrm{d}t}T(t).$$

The analytical model of the system constitutes the constraint set

$$\mathcal{C} = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, d_1, d_2\}. \tag{22}$$

The set of variables $\mathcal{Z} = \mathcal{K} \cup \mathcal{X} \cup \mathcal{F}$ includes the known variables

$$\mathcal{K} = \{u_1, u_2, y_1, y_2\}, \tag{23}$$

the unknown variables

$$\mathcal{X} = \left\{q_{\mathrm{hot}}, q_{\mathrm{cold}}, q_{\mathrm{mid}}, q_{\mathrm{bot}}, L, \dot{L}, T, \dot{T}, T_{\mathrm{cold}}\right\} \tag{24}$$

and the faults

$$\mathcal{F} = \{f_1, f_2, f_3, f_4, f_5, f_6, f_7\}. \tag{25}$$

The above model is valid for $273\mathrm{K} < T(t) < 373\mathrm{K}$ (no freezing or boiling of the water) and $0 \le L(t) \le L_{\max}$.

6.2. Structural analysis. The structure graph of the system is depicted in Fig. 6. By applying the method of Krysander *et al.* (2008) to the complete structure graph, one obtains only one MSO:

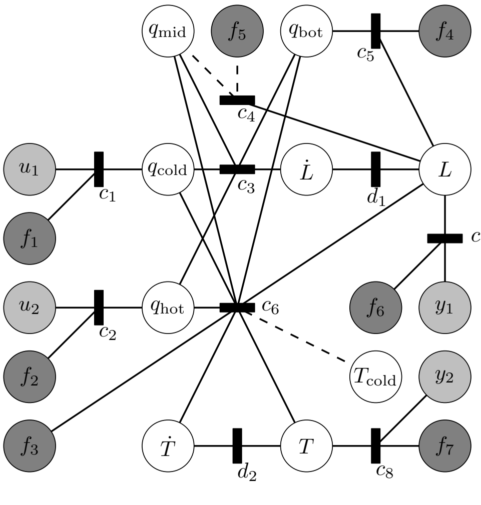

Fig. 6. Structure graph of the process engineering system.

$$\mathcal{C}^{+}_{\mathrm{MSO}} = \{c_1, c_2, c_3, c_4, c_5, c_7, d_1\}, \tag{26}$$

which is sensitive to the faults

$$\mathcal{F} \cap \mathrm{var}(\mathcal{C}^{+}_{\mathrm{MSO}}) = \{f_1, f_2, f_4, f_5, f_6\}. \tag{27}$$

Therefore, the faults f_3 and f_7 are structurally non-detectable, while the other faults are not structurally isolable.

6.3. Determining tests with the proposed method. In the first step, Algorithm 1 is used to determine all edges that may vanish in particular operating regions along with constraints describing these operating regions. The set of edges e_k obtained in Step 2 of the algorithm is given in Tab. 2 together with their adjacent variable-vertex z_j and constraint-vertex c_i and the condition $c_{\mathrm{Elim},k}$ under which the edge disappears.

Table 2. Edges in the graph that can be eliminated.

Edge	z_j	c_i	Constraint $c_{\mathrm{Elim},k}$
e_1	f_5	c_4	$L(t) = 0$
e_2	L	c_4	$L(t) \to \infty$
e_3	f_4	c_5	$L(t) \le L_{\mathrm{mid}} \ \vee \ L(t) \to \infty$
e_4	L	c_5	$L(t) \le L_{\mathrm{mid}} \ \vee \ L(t) \to \infty$
e_5	T	c_6	$A\dot{L}(t) + q_{\mathrm{bot}}(t) + q_{\mathrm{mid}}(t) = 0$
e_6	q_{bot}	c_6	$T(t) = 0$
e_7	q_{mid}	c_6	$T(t) = 0$
e_8	T_{cold}	c_6	$q_{\mathrm{cold}}(t) = 0$
e_9	q_{cold}	c_6	$T_{\mathrm{cold}}(t) = 0$
e_{10}	f_3	c_6	$T_{\mathrm{hot}} q_{\mathrm{hot}}(t) = 0$
e_{11}	$\dot{T}$	c_6	$L(t) = 0$
e_{12}	L	c_6	$\dot{T}(t) = 0$
e_{13}	$\dot{L}$	c_6	$T(t) = 0$
e_{14}	T	c_6	$\dot{L}(t) = 0$

The edges e_2, e_6, e_7, e_9 and e_{13} actually cannot be eliminated because the corresponding condition requires the temperatures to be zero or the level to be greater than $L_{\max}$, which contradicts the region of the model validity. Therefore, the set of edges that can actually be eliminated is

$$\mathcal{E}_{\mathrm{Elim}} = \{e_1, e_3, e_4, e_5, e_8, e_{10}, e_{11}, e_{12}, e_{14}\}. \tag{28}$$

Note that some of the edges cannot be eliminated independently of each other because the corresponding constraints $c_{\mathrm{Elim},k}$ are the same. Therefore, not all of the $2^{|\mathcal{E}_{\mathrm{Elim}}|} = 2^9 = 512$ theoretically possible structures are realizable. For example, it is not possible to eliminate the edge e_3 without eliminating the edge e_4, because the conditions $c_{\mathrm{Elim},3}$ and $c_{\mathrm{Elim},4}$ describing the corresponding operating regions are identical.

In the following, one particular local structure is investigated, where the edges e_3, e_4 and e_8 are eliminated.

In Fig. 6 these edges are dashed. The local structure $G_p = \{\mathcal{Z} \cup \mathcal{C}, \mathcal{E} \setminus \{e_3, e_4, e_8\}\}$ is valid if and only if the system is in the operating region defined by

$$\mathcal{C}_{\text{Reg}} = \{c_{\text{Elim},3}, c_{\text{Elim},4}, c_{\text{Elim},8}\}. \tag{29}$$

Algorithm 2 yields the constraint sets

$$\mathcal{C}^+_{\text{val},3} = \{c_7, c_{\text{Elim},3}\}, \tag{30}$$

$$\mathcal{C}^+_{\text{val},4} = \{c_7, c_{\text{Elim},4}\}, \tag{31}$$

$$\mathcal{C}^+_{\text{val},8} = \{c_2, c_{\text{Elim},8}\}. \tag{32}$$

From these constraint sets, one can determine the validuals

$$v_3(t) = v_4(t) = \begin{cases} 0, & y_1(t) \le L_{\text{mid}}, \\ y_1(t) - L_{\text{mid}}, & y_1(t) > L_{\text{mid}}, \end{cases} \tag{33}$$

$$v_8(t) = u_2(t). \tag{34}$$

The validuals $v_3(t)$ and $v_4(t)$ are identical, because the corresponding constraints $c_{\text{Elim},3}$ and $c_{\text{Elim},4}$ are the same. In the local system structure G_p one finds eight MSOs:

$$\begin{aligned}
\mathcal{C}^+_{\text{MSO,res},1} &= \{c_2, c_3, c_4, c_5, c_6, c_7, c_8, d_1, d_2\}, \\
\mathcal{C}^+_{\text{MSO,res},2} &= \{c_1, c_3, c_4, c_5, c_6, c_7, c_8, d_1, d_2\}, \\
\mathcal{C}^+_{\text{MSO,res},3} &= \{c_1, c_2, c_4, c_5, c_6, c_7, c_8, d_1, d_2\}, \\
\mathcal{C}^+_{\text{MSO,res},4} &= \{c_1, c_2, c_3, c_4, c_6, c_7, c_8, d_1, d_2\}, \\
\mathcal{C}^+_{\text{MSO,res},5} &= \{c_1, c_2, c_3, c_5, c_6, c_7, c_8, d_1, d_2\}, \\
\mathcal{C}^+_{\text{MSO,res},6} &= \{c_1, c_2, c_3, c_4, c_5, c_7, d_1\}, \\
\mathcal{C}^+_{\text{MSO,res},7} &= \{c_1, c_2, c_3, c_4, c_5, c_6, c_8, d_1, d_2\}, \\
\mathcal{C}^+_{\text{MSO,res},8} &= \{c_1, c_2, c_3, c_4, c_5, c_6, c_8, d_2\}.
\end{aligned}$$

Note that the restriction of the system to the operating region results in the fact that significantly more MSOs than in the general case can be found. Equation (20) allows us to determine which faults may be present when a local residual is nonzero. These are the fault variables that occur in the constraint set $\mathcal{C}^+_{\text{MSO,res},s}$, which is used to determine a local residual $r_s(t)$, and the fault variables that occur in the constraint sets $\mathcal{C}^+_{\text{val},3}$, $\mathcal{C}^+_{\text{val},4}$ and $\mathcal{C}^+_{\text{val},8}$, which are used to determine the validual generators. In Table 3, these fault variables are given for each local residual generator.

In structural analysis, the *signature matrix* is used to express which fault may be the reason for a nonzero residual. Similarly, from Table 3 one can obtain a signature matrix for a test based on validuals and local residuals (see Table 4). From this matrix one can conclude that faults that were not detectable or isolable using standard structural analysis become so by steering the system in a particular operating region.

Table 3. Sensitivity to the faults.

MSO	Faults in $\boldsymbol{\mathcal{C}^+_{\text{MSO,res},s}}$	Faults in $\boldsymbol{\mathcal{C}^+_{\text{val},k}}$
$\mathcal{C}^+_{\text{MSO,res},1}$	f_2, f_3, f_5, f_6, f_7	f_2, f_4, f_6
$\mathcal{C}^+_{\text{MSO,res},2}$	f_1, f_3, f_5, f_6, f_7	f_2, f_4, f_6
$\mathcal{C}^+_{\text{MSO,res},3}$	$f_1, f_2, f_3, f_5, f_6, f_7$	f_2, f_4, f_6
$\mathcal{C}^+_{\text{MSO,res},4}$	$f_1, f_2, f_3, f_5, f_6, f_7$	f_2, f_4, f_6
$\mathcal{C}^+_{\text{MSO,res},5}$	f_1, f_2, f_3, f_6, f_7	f_2, f_4, f_6
$\mathcal{C}^+_{\text{MSO,res},6}$	f_1, f_2, f_5, f_6	f_2, f_4, f_6
$\mathcal{C}^+_{\text{MSO,res},7}$	f_1, f_2, f_3, f_5, f_7	f_2, f_4, f_6
$\mathcal{C}^+_{\text{MSO,res},8}$	$f_1, f_2, f_3, f_5, f_6, f_7$	f_2, f_4, f_6

Table 4. Signature matrix if $v_3(t) = v_4(t) = v_8(t) = 0$ holds.

r_s	f_1	f_2	f_3	f_4	f_5	f_6	f_7
r_1		1	1	1	1	1	1
r_2	1	1	1	1	1	1	1
r_3	1	1	1	1	1	1	1
r_4	1	1	1	1	1	1	1
r_5	1	1	1	1		1	1
r_6	1	1		1	1	1	
r_7	1	1	1	1	1	1	1
r_8	1	1	1	1	1	1	1

6.4. Simulation results. In Fig. 7, the local residuals $r_1(t)$, $r_2(t)$, $r_5(t)$ and $r_6(t)$ along with the validuals $v_3(t)$, $v_4(t)$ and $v_8(t)$ are shown for the fault free system and for the system subject to the fault f_2. The local residuals are obtained by elimination of the unknowns from the corresponding sets of constraints $\mathcal{C}^+_{\text{MSO,res},1}$, $\mathcal{C}^+_{\text{MSO,res},2}$, $\mathcal{C}^+_{\text{MSO,res},5}$ and $\mathcal{C}^+_{\text{MSO,res},6}$:

$$\begin{aligned} r_1(t) =& (A\dot{y}_2 y_1(t) + \dot{y}_1(t)(y_2(t) - T_{\text{hot}})) \\ &+ a\sqrt{y_1(t)}(y_2(t) - T_{\text{hot}}), \end{aligned} \tag{35}$$

$$\begin{aligned} r_2(t) =& A(\dot{y}_1(t)y_2(t) + y_1(t)\dot{y}_2(t)) \\ &+ y_2(t)a\sqrt{y_1(t)} - T_{\text{hot}}u_1(t), \end{aligned} \tag{36}$$

$$r_5(t) = T_{\text{hot}}u_1(t) - A\dot{y}_2(t)y_1(t) - y_2(t)u_1(t), \tag{37}$$

$$r_6(t) = u_1(t) - A\dot{y}_1(t) - a\sqrt{y_1(t)}. \tag{38}$$

As long as not all three validuals are zero, the local residuals (35)–(38) cannot be used for fault isolation, because the system is not in the operating region corresponding to the structure that was used to determine these local residuals. In the fault free case, all the three validuals are zero for $t > 175$ s. This is the property that allows us to use the local residuals for fault diagnosis. All four residuals vanish as expected. In the case of the fault f_2, all validuals are zero for $t > 230$ s. Then the fact that $r_1(t) \neq 0$ holds allows us to infer on the presence of one of the faults in $\{f_2, f_3, f_4, f_5, f_6, f_7\}$.

Figure 8 shows the effect of the faults f_1, f_3 and f_5 on the local residuals. In the case of the fault f_3, the local

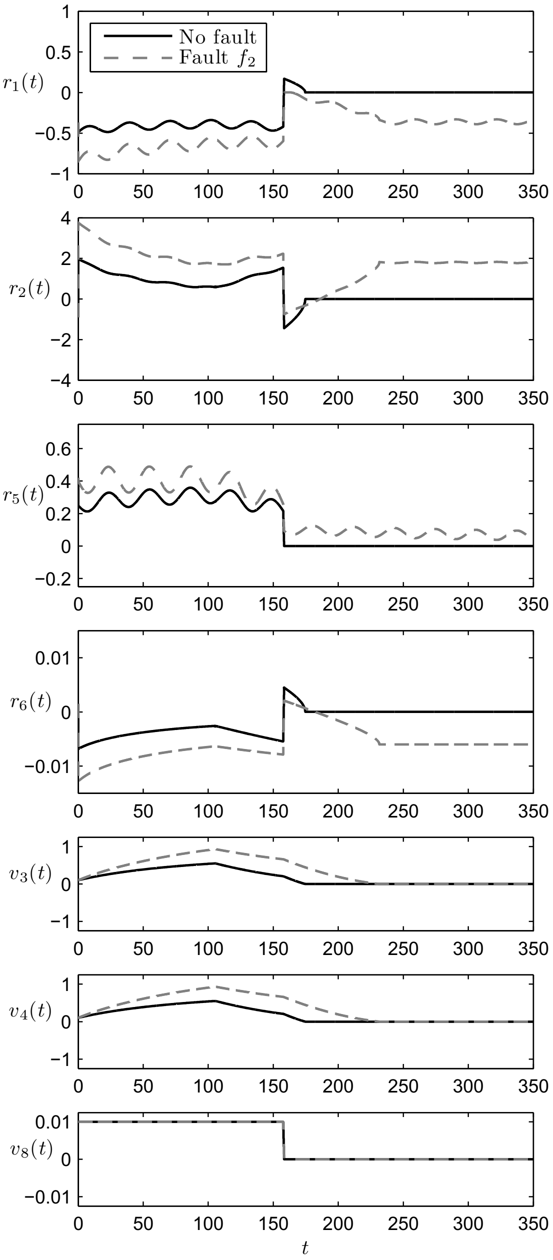

Fig. 7. Test based on the elimination of the edges e_3, e_4 and e_8: nominal behavior and behavior in the case of fault f_2.

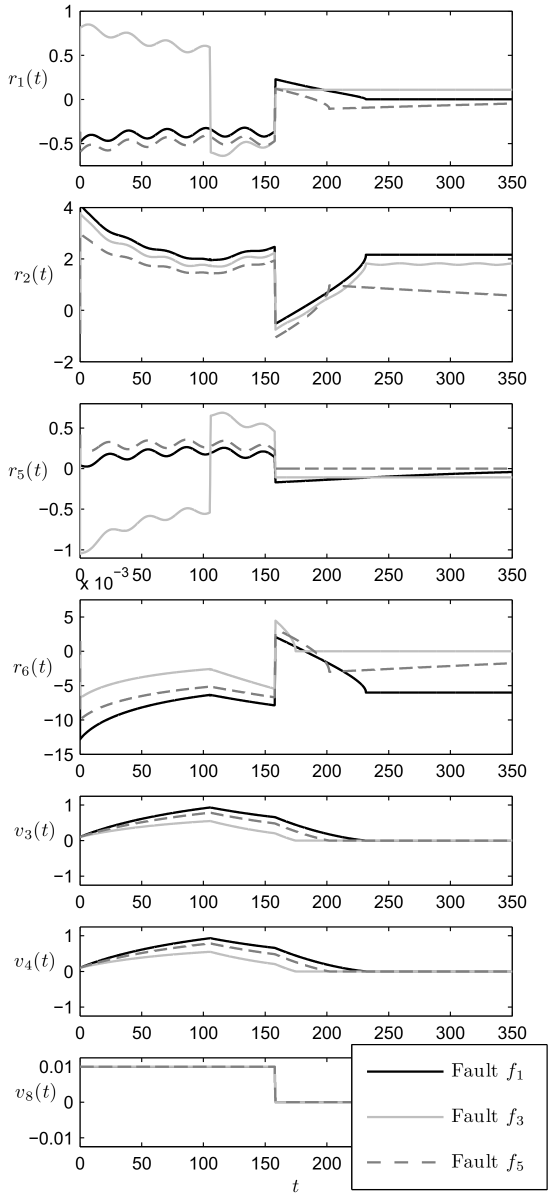

Fig. 8. Test based on the elimination of the edges e_3, e_4 and e_8: fault scenarios f_1, f_3 and f_5.

residuals are nonzero if the validuals are zero. Hence, the local residuals allow detecting fault f_3. Note that with the result of standard structural analysis in Section 6.2, the detection of the fault f_3 is not possible. Again, the specific structure, which holds as long as the validuals $v_3(t)$, $v_4(t)$ and $v_8(t)$ are zero, makes it possible to detect this fault.

Moreover, if the validuals are zero, the local residuals allow distinguishing between some faults. If some local residuals are nonzero and the single fault assumption holds, only a fault acting on all the nonzero local residuals may be present. For example, if $r_2(t)$, $r_5(t)$ and $r_6(t)$ are nonzero (black line in Fig. 8), the signature matrix of Table 4 allows us to conclude that one of the faults f_1, f_2, f_4 or f_6 is present. This conclusion matches with the actually simulated fault scenario f_1.

In Fig. 7, one can also observe that, in the presence of the fault f_2, the local residual $r_2(t)$ is nonzero when the validuals are zero. This illustrates the meaning of Eqn. (20). Although f_2 does not appear in the constraint set $\mathcal{C}^+_{\mathrm{MSO,res},2}$, which was used to determine $r_2(t)$, the local residual is sensitive to f_2. This is because f_2 appears in the constraint set $\mathcal{C}^+_{\mathrm{val},8}$, which was used to determine the validual $v_8(t)$.

7. Conclusion

This paper describes a method for test generation based on local system structures. It is shown that particular input signals may steer a system in an operating region where some of the edges in the structure graph disappear and, hence, faults can be distinguished, which otherwise cannot. In this way structural analysis makes it possible to identify operating regions together with specific local residual generators. The main result of this work is a method for determining signals from known variables guaranteeing the presence of a particular local structure. This was achieved by transforming the original problem into that of finding MSOs and solving it with well known methods (Krysander *et al*., 2008).

The presented approach offers a solution of the problem observed by Laursen *et al*. (2008) that the diagnostic result can be wrong when steering in the corresponding operating region fails.

A dynamical system may have a large number of different local structures, some of which entail different local structural isolability properties. Future research will therefore concentrate on the problem of identifying local system structures that allow the isolation of faults which are not isolable by using an arbitrary local structure.

References

Bayoudh, M., Travé-Massuyès, L. and Olive, X. (2008). Towards active diagnosis of hybrid systems, *19th International Workshop on Principles of Diagnosis, Blue Mountains, Australia*, pp. 231–237.

Bayoudh, M., Travé-Massuyès, L. and Olive, X. (2009). Active diagnosis of hybrid systems guided by diagnosability properties, *IFAC Safeprocess, Barcelona, Spain*, pp. 1498–1503.

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

Campbell, S.L. and Nikoukhah, R. (2004). *Auxiliary Signal Design for Failure Detection*, Princeton University Press, Princeton, NJ.

Clever, S. and Isermann, R. (2008). Model-based fault detection and diagnosis with special input excitation applied to a modern diesel engine, *Proceedings of the 17th IFAC World Congress, Seoul, Korea*, pp. 2031–2036.

Dulmage, A.L. and Mendelsohn, N.S. (1958). Coverings of bipartite graphs, *Canadian Journal of Mathematics* **10**(4): 516–534.

Krysander, M., Åslund, J. and Nyberg, M. (2008). An efficient algorithm for finding minimal overconstrained subsystems for model-based diagnosis, *IEEE Transactions on Systems, Man, and Cybernetics* **38**(1): 197–206, DOI: 10.1109/TSMCA.2007.909555.

Laursen, M., Blanke, M. and Düştegör, D. (2008). Fault diagnosis of a water for injection system using enhanced structural isolation, *International Journal of Applied Mathematics and Computer Science* **18**(4): 593–603, DOI: 10.2478/v10006-008-0052-5.

Niemann, H. and Poulsen, N.K. (2005). Active fault diagnosis in closed-loop systems, *Proceedings of the 16th IFAC World Congress, Prague, Czech Republic*, DOI: 10.3182/20050703-6-CZ-1902.01877.

Riggins, R.N. and Ribbens, W. B. (1997). Designed inputs for detection and isolation of failures in the state transition matrices of dynamic systems, *IEEE Transactions on Control Systems Technology* **5**(2): 149–162.

Ungermann, M., Lunze, J. and Schwarzmann, D. (2010a). Model-based test signal generation for service diagnosis of automotive systems, *Advances in Automotive Control, Munich, Germany*, DOI: 10.3182/20100712-3-DE-2013.00029.

Ungermann, M., Lunze, J. and Schwarzmann, D. (2010b). Service diagnosis utilizing the dependencies between the system structure and the operating points, *Conference on Control and Fault-Tolerant Systems, Nice, France*, pp. 873–878.

Zhang, X.J. (1989). *Auxiliary Signal Design in Fault Detection and Diagnosis*, Springer, Heidelberg.

Michael Ungermann obtained the German diploma in electrical engineering from TU Darmstadt and the French diploma in control of industrial processes from INPG in 2008. Since then, he has been a Ph.D. student at the Institute of Automation and Computer Control at Ruhr-Universität Bochum with Corporate Sector Research and Advanced Engineering of Robert Bosch GmbH. His main interest is service diagnosis based on structural models.

Jan Lunze obtained the Ph.D. and D.Sc. degrees, both from the Technical University of Ilmenau, in 1980 and 1983. Between 1992 and 2001 he was a professor of control engineering at the Technical University of Hamburg–Harburg. Now he is a professor of automatic control and the head of the Institute of Automation and Computer Control at Ruhr-Universität Bochum, where he teaches systems and control theory. His research interests are in fault-tolerant control, hybrid systems and networked control systems.

Dieter Schwarzmann received his Ph.D. in control from Ruhr-Universität Bochum in 2006 in nonlinear internal model control. In 2003 he obtained his M.Sc. in mechanical engineering from the Rose-Hulman Institute of Technology and his German degree from Stuttgart University. He is now employed at Robert Bosch GmbH, working in a research and development unit on control systems. Modeling physical systems and adaptive control are his main areas of interest.

Received: 17 January 2011
Revised: 6 July 2011