

Robust Fault Diagnosis of Non-linear Systems using Interval Constraint Satisfaction and Analytical Redundancy Relations

Sebastian Tornil-Sin, Carlos Ocampo-Martinez, Vicenç Puig and Teresa Escobet

Abstract—In this paper, the robust fault diagnosis problem for non-linear systems considering both bounded parametric modelling errors and noises is addressed using parity equation based Analytical Redundancy Relations and Interval Constraint Satisfaction techniques. Fault detection, isolation and estimation tasks are considered. Moreover, the paper addresses the problem of determining the uncertainty in the parameters of the used uncertain ARR. To illustrate the usefulness of the proposed approach, a case study based on the well known wind turbine benchmark is used.

Index Terms—Fault detection, robustness, intervals, set-membership estimation, constraint satisfaction.

I. INTRODUCTION

MODEL-BASED fault detection and isolation (FDI) of dynamic systems relies on the use of the *analytic redundancy* provided by the model. Model-based FDI systems check the consistency between known variables, inputs and measured outputs, according to the relations provided by a normal operation model of the system. The consistency is normally expressed in terms of *residuals*, computable expressions that evaluate to zero in absence of faults. One residual is enough for fault detection, several residuals allow fault isolation.

When building a model of a dynamic system to monitor its behaviour, there is always some mismatch between the modelled and the real behaviour, e.g., some effects are neglected, some non-linearities are linearised in order to simplify the model, some parameters have tolerance when are compared between several units of the same component, some errors in parameters or in

the structure of the model are introduced in the model calibration process, among others. These modelling errors introduce some uncertainty in the model. According to the FDI literature (see [1] and [2], among others), the goal of a robust fault detection method is to discriminate between fault and uncertainty effects. In practice, a robust fault detection system has to simultaneously maximize sensitivity to faults while minimizing sensitivity to uncertainties. According to [1], two types of robust approaches have been proposed: *active* and *passive*. Active robust approaches face modelling errors as if they were disturbances, using a disturbance decoupling principle and trying to obtain residuals sensitive to faults but not to these errors. Techniques such as unknown input observers [3], eigenstructure assignment [1] or structured parity equations [2], among others, can be found in the literature. On the other hand, process and measurement noises are usually stochastically modelled (the typical assumption is a zero-mean white noise) and their effect is considered by using statistical decision methods [4] to evaluate the residuals.

However, such approaches show several drawbacks. Regarding modelling errors, the first problem consists in the difficulty of representing them as disturbances and solving the associated decoupling problem (the distribution matrix is normally unknown and time varying). Moreover, the number of decoupled disturbances/modelling errors is limited by the degree of freedom in the residual generation procedure [2]. Finally, if the fault detection system is insensitive to modelling errors then it will be also insensitive to parametric faults. As an alternative strategy, disturbances/modelling errors may be assumed to be bounded and their effects propagated to the residual using, for example, interval methods [5]. Regarding measurement noises, in some practical situations it is not realistic to assume the availability of a known statistical distribution law. For instance, when the volume of collected data is small or when the measurement uncertainty is corrupted by some deterministic systematic errors, due to not considered model errors and/or disturbances. If noise bounds are available, several

S.Tornil-Sin, C.Ocampo-Martinez, V.Puig and T.Escobet are with the Advanced Control Systems Research Group, Technical University of Catalonia, Rambla de Sant Nebridi 10, 08222 Terrassa, Spain. (email: {sebastian.tornil,teresa.escobet}@upc.edu)

C.Ocampo-Martinez and V.Puig are also with Institut de Robotica i Informatica Industrial (CSIC-UPC), Llorens i Artigas 4-6, 08028 Barcelona, Spain. (email: {cocampo,vpuig}@iri.upc.edu)

This work has been supported by WATMAN (Ref. DPI-2009-13744) and SHERECS Projects (DPI-2011-26243) of the Spanish Science and Innovation Ministry and the DGR of Generalitat de Catalunya (SAC group Ref. 2009/SGR/1491).

types of mathematical tools can be applied, e.g. uniform distributions, Monte Carlo methods or interval analysis. Hence, the so called *set-membership approach* [6] can be used in the context of fault detection as suggested by [7]. The advantage of the bounded description of uncertainty is that it does not require restrictive assumptions (e.g., a small number of unknown disturbances/parameters, known statistical distribution law). However, a limitation is that faults that produce a residual deviation smaller than the residual uncertainty due to model uncertainty will remain undetected (missed detection).

In this paper, the robust fault diagnosis problem for non-linear systems considering both bounded parametric modelling errors and noises is formulated using the mathematical framework of *Interval Constraint Satisfaction Problems* (ICSPs). A constraint satisfaction problem is defined by a set of constraints imposing relations to be satisfied by the related variables and by predefined domains for these variables. This general setup is here applied to the formulation of the fault detection, isolation and estimation tasks in presence of bounded uncertainty, being this the main contribution of the paper. The proposed fault detection procedure checks the consistency between the observed and the normal system behaviour using a set of *Analytical Redundancy Relations* (ARRs), which relates the values for measured variables according to a model of normal operation (fault-free) for the monitored system. When some inconsistency is detected, the fault isolation mechanism is activated in order to identify the possible fault. The fault isolation is based on identifying the inconsistent ARRs and identifying the fault that corresponds with such signature. Finally, a fault estimation procedure also formulated as a CSP is used to estimate the fault magnitude.

The use of interval constraint satisfaction for fault detection in presence of uncertainty has been already proposed by [8] and [9] using observers and state estimators, respectively. These methods improve the approach for uncertain systems proposed by [10], which only considers system trajectories obtained from the uncertain parameter interval vertices assuming that the monotonicity property holds. On the other hand, this paper proposes to address the fault detection problem by using constraint satisfaction and ARRs in parity equation form. The advantage of using parity equations instead of state estimation/observers is that the state of the system is not required to be estimated since only measured inputs/outputs are used. This paper can be considered an adaptation of the non-linear ARR approach proposed in [11] in the case that noise and model parameters are modelled in the set-membership context and included in the ARRs. In case of linear systems, [12] has suggested

the evaluation of uncertain ARRs generated using the parity space approach using set computations. However, to the best of the authors' knowledge, the fault detection problem in presence of bounded uncertainty has not been considered before by using parity equations and ICSPs within the non-linear context. The problem of how to derive the ARRs is not considered in the paper. It is assumed that ARRs have been already been obtained from the model equations using any of the available ARR generation algorithms [13], [14]. Compared to previous works (see [8] and [9]), other contributions of this paper are highlighted as follows. First, not only the fault detection problem but also the fault isolation and fault estimation problems are considered (by using the same ICSP framework). Second, a method to quantify the uncertainty in the model (uncertain parameter estimation) using data collected from the system in non-faulty scenarios is provided. Finally, the methodology is applied to a realistic example that is used as a benchmark by the research community in fault diagnosis and fault-tolerant control: the wind turbine benchmark.

The paper is structured as follows. Section II outlines a basic background on constraint satisfaction. In Section III, robust fault detection using ARRs is formulated as a ICSP. In Section IV, fault isolation and estimation task using ARRs are also formulated as ICSPs. Section V shows how to estimate the uncertainty in the system model parameters using data collected in a non-faulty scenario. The description of the whole integrated diagnosis methodology is addressed in Section VI. In Section VII, to illustrate the usefulness of the proposed approach, it is applied to the wind turbine benchmark. Finally, in Section VIII, the conclusions are drawn.

II. CONSTRAINT SATISFACTION BACKGROUND

A. Constraint Satisfaction Problems

A CSP on sets can be formulated as a 3-tuple $\mathcal{H} = (\mathcal{Z}, \mathcal{D}, \mathcal{C})$ [15], where

- $\mathcal{Z} = \{z_1, \dots, z_n\}$ is a finite set of variables,
- $\mathcal{D} = \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$ is the set of variables domains represented by closed sets, and
- $\mathcal{C} = \{c_1, \dots, c_m\}$ is a finite set of constraints, where each constraint c_i is specified by a pair $(\mathcal{Z}_i, \mathcal{R}_i)$ such that \mathcal{Z}_i is a subset of \mathcal{Z} , called the constraint scope, and \mathcal{R}_i , called constraint relation, is a relation specifying the allowed combination of values for the variables in \mathcal{Z}_i .

Solving a CSP consists in finding all variable value assignments such that all constraints are satisfied. The variable value assignment $(\tilde{z}_1, \dots, \tilde{z}_n) \in \mathcal{D}$ is a solution of \mathcal{H} if all constraints in \mathcal{C} are satisfied. The set of all

solution points of \mathcal{H} is called the *global solution set* and denoted by $\mathcal{S}(\mathcal{H})$. The variable $z_i \in \mathcal{Z}$ is *consistent* in \mathcal{H} if and only if $\forall \check{z}_i \in \mathcal{D}_i$

$$\exists (\check{z}_1 \in \mathcal{D}_1, \dots, \check{z}_{i-1} \in \mathcal{D}_{i-1}, \check{z}_{i+1} \in \mathcal{D}_{i+1}, \dots, \check{z}_n \in \mathcal{D}_n)$$

such as $(\check{z}_1, \dots, \check{z}_n) \in \mathcal{S}(\mathcal{H})$.

The solution of a CSP is said to be *globally consistent* if and only if every variable is consistent. A variable is *locally consistent* if and only if it is consistent with respect to all directly connected constraints. Thus, the solution of the CSP is said to be locally consistent if all variables are locally consistent. An algorithm for finding an approximation of the solution set of a CSP can be found in [15].

B. Implementation using Intervals

It is well known that the solution of CSPs involving sets has a high computational complexity [15]. In order to reduce complexity, the CSPs derived from the fault diagnosis problem are relaxed to Interval Constraint Satisfaction Problems (ICSPs) [16].

Available interval constraint satisfaction solvers allow to combine techniques with local and global consistency. Local consistency can be obtained by using *domain contraction* and *propagation*, global consistency requires *bisection*.

Contraction: First of all, consider the following definition.

Definition 2.1 (Contractor): A contractor is an operator that reduces domains. Applied to the solution \mathcal{H} of a CSP, an operator $\mathcal{C}_{\mathcal{H}} : \mathbb{IR}^n \rightarrow \mathbb{IR}^n$ is a contractor if it satisfies

$$\forall [z] \in \mathcal{D} : \begin{cases} \mathcal{C}_{\mathcal{H}}([z]) \subset [z] \\ \mathcal{C}_{\mathcal{H}}([z]) \cap \mathcal{S}(\mathcal{H}) = [z] \cap \mathcal{S}(\mathcal{H}). \end{cases} \quad (1)$$

Contractors are algorithms that reduce the interval domains of variables which comply with a set of constraints. The purpose of a contractor is to reduce any box $[z]$ without losing any solution point in $\mathcal{S}(\mathcal{H})$. In [15], a number of contractors for a variety of sets are given. The application of the contractor operator is therefore known as *contraction*.

Propagation: Consider now the following definition.

Definition 2.2 (Propagator): A propagator is an operator that sequentially performs contractors until no more significant contraction can be observed when several constraints are involved.

The procedure of applying propagators is known as *propagation*. The interval propagation method converges to a box which contains all solution vectors of the constraint set. If this box is empty, it means that there is no solution. It can be shown that the box to which the

method converges does not depend on the order to which the contractors are applied [15], but the computation time is highly sensitive to this order. There is no optimal order in general, but in practice, one of the most efficient is called *forward-backward propagation*.

Bisection: The combined use of contraction and propagation leads only to obtain a *local consistent solution*. The locality problem is due to the strategy for reducing domains processes every constraint projection independently. To escape from local consistency, every resultant box from the application of contraction and propagation should be bisected in two sub-boxes, sharing all variable domains of the original box, except the one with largest width, which is split by its mid point. Then, contractors and propagation are applied on the new resultant sub-boxes. This process, called *bisection*, is iterated until no refinement of those sub-boxes below to a pre-established amount is achieved.

C. Computational and implementation issues

Contractors are algorithms that present polynomial complexity in the number of uncertain variables. Unfortunately, since they only assure local consistency, their use may lead to over-bounded solutions depending on the considered problem. To avoid this, contractions are combined with bisections. However, the use of bisections leads to a branch and bound algorithmic structure with exponential complexity.

To obtain the results presented in the case of study, the *Real Paver* solver [17] has been used. *Real Paver* allows to use or to limit bisections, which according to the previous discussion has a direct impact in the obtained solutions and in the computation time. This allows to adapt the solver to the particular properties of the considered problem.

III. ROBUST FAULT DETECTION AS ICSP

A. System Modelling

Consider that the behaviour of system to be monitored may be described by the following discrete-time non-linear model:

$$\begin{aligned} x_{k+1} &= g(x_k, u_k, \theta_k) + w_k, \\ y_k &= h(x_k, u_k, \theta_k) + v_k, \end{aligned} \quad (2)$$

where $x \in \mathbb{R}^{n_x}$ is the vector of system states, $u \in \mathbb{R}^{n_u}$ is the vector of system inputs and $y \in \mathbb{R}^{n_y}$ is the vector of system outputs; $\theta_k \in \mathbb{R}^{n_\theta}$ is a vector of uncertain parameters; $w_k \in \mathbb{R}^{n_w}$ and $v_k \in \mathbb{R}^{n_v}$ are process and measurement noises; $g : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_y}$ are the state-space and measurement non-linear functions, respectively.

The proposed model is able to represent different types of uncertainties through θ_k , w_k and v_k . Unmodelled dynamics, disturbances and errors due to the discretization process may be captured in the process noise w_k . Errors in sensors are represented in the measurement noise v_k . Parameter uncertainty is represented through θ_k . It is assumed that all uncertain parameters, process and measurements noises are unknown (i.e., their instantaneous values) but bounded in (known) intervals. All this latter may be expressed in compact form as

$$\theta_k \in \Theta = \{\theta \in \mathbb{R}^{n_\theta} \mid \underline{\theta} \leq \theta \leq \bar{\theta}\}, \quad (3)$$

$$w_k \in \mathcal{W} = \{w \in \mathbb{R}^{n_w} \mid \underline{w} \leq w \leq \bar{w}\}, \quad (4)$$

$$v_k \in \mathcal{V} = \{v \in \mathbb{R}^{n_v} \mid \underline{v} \leq v \leq \bar{v}\}, \quad (5)$$

where Θ , \mathcal{W} and \mathcal{V} are axis-aligned n -dimensional boxes (Cartesian products of intervals).

Different cases may be considered with respect to the time variance of the parameter vector θ_k (uncertainty in v_k and w_k is naturally assumed to be time-variant). If no additional conditions are stated, according to (3) the system is considered time variant with parameters bounded but freely varying inside Θ . But sometimes the inter-sample variance is known to be bounded. This can be represented by adding to the system description (2) the following equations

$$\theta_{k+1} = \theta_k + p_k, \quad (6)$$

$$p_k \in \mathcal{P} = \{p \in \mathbb{R}^{n_p} \mid \underline{p} \leq p \leq \bar{p}\} \quad (7)$$

Finally, it can be assumed that the monitored system is time invariant. This can be easily represented as particular case with $p_k = 0$.

A key aspect about the use of uncertain models is how the uncertainty bounds are obtained. Regarding process and measurement noises, it is assumed in this work that a priori theoretical or practical considerations allow to obtain useful intervals that define the boxes \mathcal{W} and \mathcal{V} . This is the case, for instance, when the precision or the maximum absolute errors of the used sensors are known. Or when starting from a given continuous-time model a particular discretization method is used. Regarding the parameter uncertainty, it is assumed that a priori initial intervals can also be obtained, leading to an initial box Θ_0 . For instance, this is the case when tolerances for the values of components or limits for physical coefficients are known. A problem appears when these intervals are too wide and their direct use limit seriously the detectability of faults. To avoid this problem, set-membership parameter estimation [6] can be applied. An algorithm implementing such type of parameter estimation is proposed in Section V. Applied to data collected in a fault-free scenario, the algorithm

will obtain the reduced parameter box Θ that will be used for fault diagnosis purposes.

B. ARR Generation using Structural Analysis

The design of model-based diagnosis systems may be based on utilizing the system model in the construction of diagnosis tests based on ARRs. These latter are static or dynamic constraints deduced from the system model that link the time evolution of the known variables (inputs and measured outputs) when the system operates in a non-faulty condition, and commonly represented as equalities of the form $r_i = 0$. To obtain ARRs for state-space representations such as (2), it is necessary to manipulate the model to eliminate the unknown state x . As a consequence of this model manipulation, the obtained ARRs relate the values of the known variables not only in the current time instant but along a time horizon of length L . For linear systems, the unknown state may be eliminated by using the parity-space approach [18]. For particular types of non-linear systems, elimination methods (e.g., elimination theory, Gröbner bases or characteristic sets) may be used [11]. For complex and non-linear systems in general, it is possible to obtain ARRs, by using, among others (e.g., state observers), the structural methods presented in [19].

Let define the sequences of values for the variables involved in (2) as

$$\tilde{q}_k = \{q_j\}_{j=k-L}^k, \quad q \in \{u, y, \theta, w, v, p\}. \quad (8)$$

Any equation obtained from manipulation or combination of the equations in (2) can be represented in the following way:

$$r_i = \Psi_i(\tilde{u}_k, \tilde{y}_k, \tilde{\theta}_k, \tilde{w}_k, \tilde{v}_k, \tilde{p}_k) = 0, \quad (9)$$

where Ψ_i is called the *ARR expression*. Strictly speaking, it can not be said that (9) is an ARR because it depends on several unknown variables, but this will be taken into account later. For a more compact notation, if all the unknown but bounded variables are represented together

$$\tilde{\delta}_k = \left(\tilde{\theta}_k, \tilde{v}_k, \tilde{w}_k, \tilde{p}_k \right), \quad (10)$$

then the general expression for an uncertain ARR is

$$r_i = \Psi_i(\tilde{u}_k, \tilde{y}_k, \tilde{\delta}_k) = 0. \quad (11)$$

The structural analysis of the system will provide a set of ARRs that agree with the previous general form. This set can be represented as

$$\mathcal{R} = \{r_i = \Psi_i(\tilde{u}_k, \tilde{y}_k, \tilde{\delta}_k) = 0, i = 1, \dots, n_r\}, \quad (12)$$

where n_r is the number of obtained ARRs.

Remark 3.1: According to (11), the use of ARR is preferred since it allows to reduce the ICSP computational burden. However, the proposed methodology also allows to consider directly the initial non-linear equations when their combination is not possible.

C. Fault Detection using ARRs and ICSP

Using the set of generated ARRs, the fault detection procedure must check at each time instant whether or not they are consistent with the observations. If all the variables in the ARRs expressions were known, this would mean to check if all ARR expressions evaluate to zero or not. However, only some of the variables are known and a different type of consistency checking procedure is needed to deal with the uncertain variables. If the unknown variables are simply not taken into account in the evaluation of the ARR expressions (evaluating simplified expressions commonly called computational forms), then non-zero values will be obtained in the non-faulty case. If the model parameters were precisely known and statistical distributions for the noises were available, an statistical framework might be adopted to determine whether these deviations from zero had to be attributed to the uncertainty or to the presence of faults. See the book [4] for further details about such type of statistical FDI techniques. Here, instead, considering the unknown but bounded description of the noise, disturbances and parametric uncertainty in (2), the consistency checking procedure is performed as follows.

Definition 3.1 (Consistency for sets of ARRs): Given a set of ARRs expressed according to (12) and a sequence of measured system inputs \tilde{u}_k and outputs \tilde{y}_k at time k , the set of ARRs is consistent with those measurements and the known bounds of uncertain parameters and noises if there exist valid sequences (with values inside their corresponding intervals) for these uncertain variables that simultaneously satisfy the ARRs, i.e.

$$\exists \tilde{\delta}_k \in \tilde{\mathcal{D}}_\delta \mid \left(\Psi_i(\tilde{u}_k, \tilde{y}_k, \tilde{\delta}_k) = 0 \right)_{i=1}^{n_r}, \quad (13)$$

where $\tilde{\delta}_k \in \tilde{\mathcal{D}}_\delta$ means that every element in the sequence $\tilde{\delta}_k$ is inside the domain given by (the \times operator represent the Cartesian product)

$$\mathcal{D}_\delta = \Theta \times \mathcal{W} \times \mathcal{V} \times \mathcal{P}. \quad (14)$$

The previous definition directly provides a way to implement fault detection. If the set of ARRs (12) is proven to be inconsistent (not consistent according to the previous definition) at a given time instant k , then the system behaviour can not be explained by the model (2) and a fault is detected. Inconsistency can be verified by

Algorithm 1 Fault detection using ICSP

```

1: for  $k = 0$  to  $N$  do
2:   Get current measured input/output data  $(u_k, y_k)$ 
3:   Update measured input/output sequences  $(\tilde{u}_k, \tilde{y}_k)$ 
4:    $\mathcal{Z} \leftarrow \{\tilde{\delta}_k\}$ 
5:    $\mathcal{D} \leftarrow \{\tilde{\mathcal{D}}_\delta\}$ 
6:    $\mathcal{C} \leftarrow \{\Psi_i(\tilde{u}_k, \tilde{y}_k, \tilde{\delta}_k) = 0, i = 1..n_r\}$ 
7:    $\mathcal{H}_k = (\mathcal{Z}, \mathcal{D}, \mathcal{C})$ 
8:    $\mathcal{S}_k = \text{solve}(\mathcal{H}_k)$ 
9:   if  $\mathcal{S}_k = \emptyset$  then
10:     Exit (Fault detected)
11:   end if
12: end for

```

solving an ICSP whose variables are the ones involved in (10), their domains are given by (14) and the constraints correspond to the ARRs in (12), and checking whether the obtained result is the empty set. This procedure is summarized in Algorithm 1. Notice that the sequences \tilde{u}_k and \tilde{y}_k are considered as parameters (not variables) in the ICSP formulation and that the total number of variables in the problem is $(n_\theta + n_w + n_v + n_p)L$.

Remark 3.2: The requirement about the existence of sequences for the uncertain variables that simultaneously satisfy all the ARRs allows to take into account that for a variable that appear in several ARRs the associated unknown value (for a given time instant) must be the same in all these ARRs. If this fact would not be considered then some sensitivity to faults would be lost.

Remark 3.3: Since outer approximations are computed by ICSP solvers, if the obtained result is the empty set then it can be assured that the exact solution is in fact the empty set. It is therefore preserved in practice the expected property associated to a bounded description of uncertainty by which false alarms are avoided.

IV. FAULT ISOLATION AND ESTIMATION AS ICSPS

A. Fault Isolation as ICSPs

Once a fault is detected by using Algorithm 1, fault isolation is considered. Fault isolation aims at identifying the fault acting on the system in a set of possible faults or *fault hypothesis set* $\mathcal{F} = \{f^1, f^2, \dots, f^{n_f}\}$. It is assumed that just one fault may be acting on the system at a given time instant.

Fault isolation is based on identifying at time instant k those ARRs that are consistent and those that are not and implementing a diagnostic reasoning that leads to the identification of the fault present in the system, assuming that different faults affect different ARRs. This fact means that, unlike for fault detection, consistency has to

be evaluated independently for each ARR. This can be done according to the following definition.

Definition 4.1 (Consistency for a single ARR):

Given an ARR expressed according to (11) and a sequence of measured system inputs \tilde{u}_k and outputs \tilde{y}_k at time k , the ARR is consistent with those measurements and the known bounds of uncertain parameters and noises if there exist valid sequences for these uncertain variables that satisfy the ARR, i.e.

$$\exists \tilde{\delta}_k \in \tilde{\mathcal{D}}_\delta \mid \Psi_i(\tilde{u}_k, \tilde{y}_k, \tilde{\delta}_k) = 0. \quad (15)$$

Using the previous definition, fault isolation starts by obtaining the *observed fault signature* $\{\phi_{1k}, \phi_{2k}, \dots, \phi_{n_r k}\}$, where each single fault signal indicator ϕ_{ik} is defined as

$$\phi_{ik} = \begin{cases} 0, & \text{if } r_{ik} \text{ is consistent,} \\ 1, & \text{if } r_{ik} \text{ is inconsistent.} \end{cases} \quad (16)$$

Standard fault isolation reasoning exploits the knowledge about the binary relation between the set of fault hypothesis and the set of ARRs that is stored in the so called *Fault Signature Matrix* (FSM), denoted as M . An element m_{ij} (i indicates rows, j indicates columns) of M is equal to 1 if the fault f^j affects the computation of the ARR r_i ; otherwise, the element m_{ij} is zero-valued. A column of M is known as a *theoretical fault signature* and indicates which ARRs are affected by a given fault. A set of faults is *isolable* if all the columns in M are different (two columns that are equal indicate two faults that can not be distinguished).

Based on the use of FSMs, different reasoning procedures have been proposed in the literature, see for instance [20]. The accepted as standard procedure by the FDI community involves finding a matching between the observed fault signature and one of the theoretical fault signatures. However, this reasoning is not appropriate in an unknown but bounded context. Due to the uncertainty, when a fault is present in the system, an undefined number of the ARRs affected by the fault can be found inconsistent, mainly depending on the sensitivity of each ARR with respect to the fault and on the fault magnitude. In other words, the observed fault signature will not exactly match the theoretical signature of the present fault. In this case, if the column-matching procedure is used, then the particular fault will not be identified. An appropriate reasoning should only consider the ARRs that are inconsistent when searching for the fault (inconsistency is relevant, consistency is not). An ARR that is found inconsistent indicates that one of the faults that affect the ARR is acting on the system. But the contrary is not true, if an ARR is satisfied this do not assures that none of the associated faults is present. According

to the established terminology [20], the used algorithm must avoid *single-fault exoneration* (which is implicit in the column matching reasoning).

Under single-fault assumption, this can be easily achieved by taking into account that the fault that is actually present in the system has to affect all the ARRs that have been found inconsistent according to the observed fault signature (if not the single fault hypothesis can not explain the observed behaviour). Algorithm 2 summarizes an isolation procedure based on this idea. Notice that the procedure requires the solution of n_r ICSP problems with the same number of variables that the one needed to solve the fault detection problem. This fact justifies the use of Algorithm 1 for real-time fault detection (Algorithm 2 may be directly applied for simultaneous fault detection and isolation, but at expenses of a higher computational cost).

Remark 4.1: Since consistency is checked independently for ARRs that share uncertain variables, some sensitivity to faults is lost. On one hand, this is a second argument that justifies the use of Algorithm 1 for fault detection. On the other hand, this means that some faults may be correctly detected but not isolated.

Remark 4.2: Due to the uncertainty, it is possible that the observed fault signature may be attributed to more than one fault and hence more than one fault candidate is provided by Algorithm 2. On other hand, it can always be assured that the real fault present in the system is one of the proposed fault candidates.

B. Fault Estimation

Once a fault f^j ($j \in \{1, \dots, n_f\}$) has been isolated at time k , fault estimation is considered. The goal of the fault estimation task is to estimate the fault magnitude f_k^j . This is useful, for instance, for implementing fault-tolerant control strategies. Due to the uncertainty, the estimation of the magnitude of a given fault at a given time instant will be obtained as an interval. The fault estimation procedure described here is based on modifying the consistency checking procedure introduced in Definition 3.1.

Unlike for fault isolation, where it is not necessary, the estimation of the fault magnitude requires the modelling of the way the fault affects the system. In general, if the effect of the isolated fault f_k^j is included in the system model (2) as

$$\begin{aligned} x_{k+1} &= g(x_k, u_k, \theta_k, f_k^j) + w_k, \\ y_k &= h(x_k, u_k, \theta_k, f_k^j) + v_k, \end{aligned} \quad (17)$$

then the application of the structural analysis approach leads to a set of ARRs that include the fault effect. This

Algorithm 2 Fault isolation using ICSP

```
1:  $k \leftarrow$  fault detection time
2:  $\mathcal{Z} \leftarrow \{\tilde{\delta}_k\}$ 
3:  $\mathcal{D} \leftarrow \{\mathcal{D}_\delta\}$ 
4: for  $i = 1$  to  $n_r$  do
5:    $\mathcal{C}_i \leftarrow \{\Psi_i(\tilde{u}_k, \tilde{y}_k, \tilde{\delta}_k) = 0\}$ 
6:    $\mathcal{H}_i = (\mathcal{Z}, \mathcal{D}, \mathcal{C}_i)$ 
7:    $\mathcal{S}_i = \text{solve}(\mathcal{H}_i)$ 
8:   if  $\mathcal{S}_i = \emptyset$  then
9:      $\phi_{i_k} = 1$ 
10:  else
11:     $\phi_{i_k} = 0$ 
12:  end if
13: end for
14:  $\mathcal{FC} \leftarrow \{f^1, f^2, \dots, f^{n_f}\}$ 
15: for  $i = 1$  to  $n_r$  do
16:   if  $\phi_{i_k} = 1$  then
17:     for  $j = 1$  to  $n_f$  do
18:       if  $m_{ij} = 0$  then
19:          $\mathcal{FC} \leftarrow \mathcal{FC} - f^j$ 
20:       end if
21:     end for
22:   end if
23: end for
24: Fault candidate set  $\mathcal{FC}$ 
```

set of fault-dependent ARR_s can be represented as

$$\mathcal{R}^j = \{r_i \mid r_i = \Psi_i^{f^j}(\tilde{u}_k, \tilde{y}_k, \tilde{\delta}_k, \tilde{f}_k^j) = 0, i = 1, \dots, n_{r_j}\}, \quad (18)$$

where \tilde{f}_k^j is the sequence of fault magnitudes along a horizon of length L and n_{r_j} is the number of ARR_s that are affected by the fault f^j . It must be noticed that the previous nomenclature is valid for both additive and multiplicative (parametric) faults.

The fault estimation procedure requires an interval bounding the possible fault magnitudes. Sometimes this interval can be obtained from theoretical or practical considerations about the nature of the fault. But if this is not possible then the initial interval can be chosen arbitrary large. For parametric faults, the initial interval used by the fault estimation algorithm can be the same used as initial interval by the parameter estimation algorithm (see Section V).

In general, assume that

$$f_k^j \in \mathcal{F}^j = \left\{ f_k^j \mid \underline{f}^j \leq f_k^j \leq \bar{f}^j \right\}. \quad (19)$$

Definition 4.2 (Fault magnitude estimation): Given an isolated fault f^j and the sequence of measured system inputs \tilde{u}_k and outputs \tilde{y}_k at time k , the estimated

magnitude for the fault at time k is given by

$$\mathcal{F}_k^j = \left\{ f_k^j \in \mathcal{F}^j \mid \begin{aligned} &\exists (\delta_l \in \mathcal{D}_\delta)_{l=k-L}^k, \\ &\exists \left(f_l^j \in \mathcal{F}_l^j \right)_{l=k-L}^{k-1}, \\ &\left(\Psi_i^{f^j}(\tilde{u}_k, \tilde{y}_k, \tilde{\delta}_k, \tilde{f}_k^j) = 0 \right)_{i=1}^{n_{r_j}} \end{aligned} \right\}, \quad (20)$$

with $\mathcal{F}_k^j \subset \mathbb{R}$.

Notice that the previous definition is recursive in time, i.e., the computation of \mathcal{F}_k^j uses the previously estimated values $\mathcal{F}_l^j, l = k-L..k-1$. Once again, the previous set can be approximated by the solution of an ICSP. This is reflected in Algorithm 3, where k_i indicates the isolation time and where the domains for the values in the fault magnitude sequence, represented by $\tilde{\mathcal{F}}_k^j$, agree with (20), i.e., $f_k^j \in \mathcal{F}^j, f_l^j \in \mathcal{F}_l^j, l = k-L..k-1$.

Remark 4.3: When several fault candidates are indicated by the fault isolation algorithm, the fault estimation algorithm is applied independently to all of them. For a given fault candidate, if the associated ICSP is found inconsistent (empty solution set) then the fault can be excluded as candidate. On the other hand, if the ICSP is consistent then an interval of possible fault magnitudes will be obtained.

Remark 4.4: The application of the ICSP framework assumes that the estimated sets \mathcal{F}_k^j are intervals (the result obtained at a given time instant is used as interval domain in the ICSP for the next time instant), but according to Definition 4.2 this fact can not be assured for any type of ARR expressions (for discontinuous expressions, for instance). Hence, it is assumed that interval enclosures are obtained at each time from the solution provided by the ICSP solver (which in general will be a union of connected or disconnected intervals).

Remark 4.5: If f^j is known to be an abrupt fault (its magnitude remains constant once it has appeared), then the domain for f_k^j can be fixed as the previously estimated interval \mathcal{F}_{k-1}^j (instead of the a priori known interval \mathcal{F}^j).

V. ARR UNCERTAIN PARAMETER ESTIMATION AS ICSP

One of the key points in passive robust model based fault detection is how models and their uncertainty bounds are obtained. Classical system identification methods [21] are formulated under a statistical framework. Assuming that the measured variables are corrupted by additive noises with known statistical distributions and that the model structure is known, a parameter estimation algorithm will provide nominal values for the

Algorithm 3 Fault estimation using ICSP for a given f^j

- 1: $\tilde{\mathcal{F}}_k^j \leftarrow \{\mathcal{F}^j\}$
 - 2: **for** $k = k_i$ to N **do**
 - 3: Get current measured input/output data (u_k, y_k)
 - 4: Update measured input/output sequences $(\tilde{u}_k, \tilde{y}_k)$
 - 5: $\mathcal{Z} \leftarrow \{\tilde{\delta}_k, \tilde{f}_k\}$
 - 6: $\mathcal{D} \leftarrow \{\tilde{\mathcal{D}}_\delta, \tilde{\mathcal{F}}_k^j\}$
 - 7: $\mathcal{C} \leftarrow \{\Psi_i^{f^j}(\tilde{y}_k, \tilde{u}_k, \tilde{\delta}_k, \tilde{f}_k^j) = 0, i = 1..n_{r_j}\}$
 - 8: $\mathcal{H}_k = (\mathcal{Z}, \mathcal{D}, \mathcal{C})$
 - 9: $\mathcal{S}_k = \text{solve}(\mathcal{H}_k)$
 - 10: Estimated fault size \mathcal{F}_k^j
 - 11: **end for**
-

parameters together with descriptions of the associated uncertainty in terms of the covariance matrix or confidence regions for a given probability level [22], [23]. However, this type of approaches can not be applied when measurement errors are described as unknown but bounded values and/or modelling errors exist. The problem of bounding the model uncertainty has been mainly stated in many references coming from robust control field. Recently, some methodologies that provide a model with its uncertainty have been developed but always thinking on its application to control [24]. One of the methodologies assumes the bounded but unknown description of the noise and parametric uncertainty. This methodology is known as *bounded-error estimation* or *set-membership estimation* [6], which produces a set of parameters consistent with the model structure selected and the pre-specified noise bounds. In [25], it is suggested that bounded error estimation problem can be solved using constraint satisfaction tools. This is the type approach used for estimating parametric uncertainty of the ARR in (12).

Regarding the uncertain variables that appear in (2), it is assumed that a priori theoretical or practical considerations allow to obtain useful intervals associated to process and measurement noises, leading to known boxes \mathcal{W} and \mathcal{V} . Moreover, if a bounded-rate time varying formulation is used, then it is assumed that the intervals defining \mathcal{P} are also known. The goal of the parameter estimation algorithm is to characterize the parameter box Θ consistent with the data collected in a fault-free scenario. The parameter estimation algorithm is quite similar to the fault estimation algorithm described in the previous subsection, the differences rely on the estimation of parameters and the off-line nature of its implementation. An initial box Θ_0 is required, but it can also be chosen arbitrarily large.

Definition 5.1 (Feasible Parameter Set (FPS)):

Given the set of ARRs (12) and a sequence of inputs \tilde{u}_k and outputs \tilde{y}_k at time k , the set of parameters consistent with the measurements and the noise and parameter variation bounds, \mathcal{W} , \mathcal{V} and \mathcal{P} , is given by

$$\Theta_k = \left\{ \theta_k \in \Theta_0 \mid \begin{aligned} &\exists (\theta_l \in \Theta_l)_{l=k-L}^{k-1}, \\ &\exists (w_l \in \mathcal{W})_{l=k-L}^k, \\ &\exists (v_l \in \mathcal{V})_{l=k-L}^k, \\ &\exists (p_l \in \mathcal{P})_{l=k-L}^k, \\ &\left(\Psi_i(\tilde{u}_k, \tilde{y}_k, \tilde{\theta}_k, \tilde{w}_k, \tilde{v}_k, \tilde{p}_k) = 0 \right)_{i=1}^{n_r} \end{aligned} \right\}, \quad (21)$$

with $\Theta_k \subset \mathbb{R}^{n_\theta}$.

The previous set identifies the set of parameters compatible with the observations and the uncertainty bounds at time instant k (this includes observations in the temporal window $k-L..k$). If measurements along a complete time horizon $k = 0..N_F$ ($N_F \gg L$) are available for a fault-free scenario, the parameter set Θ which is compatible with all the available data is of interest. This set will be computed in a different way depending on the assumed time variance for the parameter set. If a time variant behaviour is assumed then Θ is given by the union of all the individual Θ_k computed along the horizon. If the system is assumed to be time invariant then Θ is given the intersection between the different Θ_k . In the latter case, the intersection can be computed implicitly by using Θ_{k-1} as the search domain Θ_k . This is reflected in the following definition.

Definition 5.2 (Time-Invariant FPS): Given the set of ARRs (12) and sequences of inputs and outputs from $k = 0$, the set of LTI parameters consistent with the measurements and the noise bounds \mathcal{W} and \mathcal{V} is given by

$$\Theta_k = \left\{ \theta \in \Theta_{k-1} \mid \begin{aligned} &\exists (w_l \in \mathcal{W})_{l=k-L}^k, \\ &\exists (v_l \in \mathcal{V})_{l=k-L}^k, \\ &(\Psi_i(\tilde{u}_k, \tilde{y}_k, \theta, \tilde{w}_k, \tilde{v}_k) = 0)_{i=1}^{n_r} \end{aligned} \right\}. \quad (22)$$

The previous definition directly leads to the parameter estimation procedure reflected in Algorithm 4.

Remark 5.1: The application of the ICSP framework assumes that the estimated set Θ_k is a box, but in general it would be an arbitrary shaped set. Hence, it is assumed that the interval hull (smallest box that approximates the set of interest) is obtained at each time instant from the solution provided by the ICSP solver.

Algorithm 4 ARR uncertain parameter estimation using ICSP

- 1: **for** $k = 0$ to N **do**
 - 2: Get current measured input/output data (u_k, y_k)
 - 3: Update measured input/output sequences $(\tilde{y}_k, \tilde{u}_k)$
 - 4: $\mathcal{Z} \leftarrow \{\theta, \tilde{w}_k, \tilde{v}_k\}$
 - 5: $\mathcal{D} \leftarrow \{\Theta_{k-1}, \mathcal{W}, \mathcal{V}\}$
 - 6: $\mathcal{C} \leftarrow \{\Psi_i(\tilde{u}_k, \tilde{y}_k, \theta, \tilde{w}_k, \tilde{v}_k) = 0, i = 1..n_r\}$
 - 7: $\mathcal{H}_k = (\mathcal{Z}, \mathcal{D}, \mathcal{C})$
 - 8: $\mathcal{S}_k = \text{solve}(\mathcal{H}_k)$
 - 9: Estimated parameter set Θ_k
 - 10: **end for**
-

VI. DESCRIPTION OF THE WHOLE INTEGRATED DIAGNOSIS METHODOLOGY

For clarity purposes, the whole diagnosis methodology is summarized in this section. The methodology is developed in two different phases. The *Off-line phase* is completed by the following sequence of steps:

- 1) By using a-priori physical knowledge about the system, define the model structure, i.e., functions g and h and uncertainty bounds $\Theta_0, \mathcal{W}, \mathcal{V}$ and \mathcal{P} .
- 2) Define the set of faults of interest or fault hypothesis set $\mathcal{F} = \{f^1, f^2, \dots, f^{n_f}\}$.
- 3) Using structural analysis [19], derive the set of ARR that will be used for fault detection and isolation $\mathcal{R} = \{\Psi_i(\dots), i = 1..n_r\}$ and the fault signature matrix M .
- 4) Using \mathcal{R} , apply Algorithm 4 to data collected in a fault-free scenario to obtain the parameter box Θ .
- 5) Include the fault effects in the ARR to obtain the sets of fault dependent ARR that will be used for the magnitude estimation of each fault $\mathcal{R}^j = \{\Psi_i^j(\dots), i = 1..n_{r_j}, j = 1..n_f\}$.

The *On-line phase* is given by the execution, at each time instant k , of the following conditional steps:

- 1) Apply Algorithm 1 for fault detection.
- 2) If a fault is detected then apply Algorithm 2 for fault isolation.
- 3) If exactly one fault candidate f^j is obtained at previous step then use Algorithm 3 for the estimation of the fault magnitude f_k^j .

VII. CASE OF STUDY

This section illustrates the effectiveness of the proposed fault diagnosis approach by applying it to the Wind Turbine based benchmark proposed in [26].

A. System Description

Wind turbines generate electrical energy from the wind kinetic energy. The wind turbine described in the

Fault Tolerant Control benchmark proposed in [26] is a three blade horizontal axis variable speed wind turbine with a full converter coupling. The basic operation principle is that the wind energy is captured by the blades and transformed into mechanical rotational energy through the rotor and the shaft. This energy conversion can be optimized by changing the aerodynamics of the turbine by pitching the blades or by controlling the relative rotational speed of the turbine against the wind speed. The mechanical energy is in turn converted into electrical energy by a generator fully coupled to a converter. Between the rotor and the generator, a drive train is used to increase the rotational speed from the rotor to the generator. The converter can be used to set the generator torque, which consequently can be used to control the rotational speed of the generator as well as the rotor. The objective of the overall control system is to follow a power reference.

A system block diagram is presented in Figure 1, showing the relations between the different subsystems: Blade & Pitch, Drive Train, Generator & Converter and the Controller. The controlled inputs are the pitch position reference for the blades β_r and the converter reference $\tau_{g,r}$. The pitch position of each blade is measured using two sensors to ensure physical redundancy: $\beta_{1,m1}, \beta_{1,m2}, \beta_{2,m1}, \beta_{2,m2}, \beta_{3,m1}, \beta_{3,m2}$. The generator and rotor speeds are also measured with two sensors each: $\omega_{r,m1}, \omega_{r,m2}, \omega_{g,m1}, \omega_{g,m2}$.

The model that details the operation of each subsystem can be found in the original reference [26].

B. Fault Scenarios

The benchmark specifies the use of a given wind speed sequence and a set of faults that includes eight faults with different locations and types. The input sequence corresponds to real measured wind data from a wind park, sampled with $f_s=100\text{Hz}$ along a 4400s time period. The set of selected faults is the following:

- 1) Fault 1: $\beta_{1,m1}^f = 5^\circ$ (sensor fault, fixed value) in the time period 2000s-2100s.

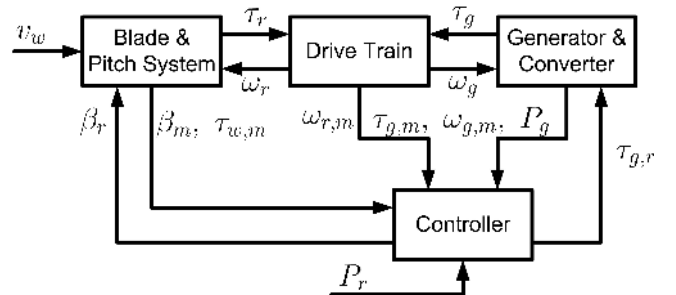


Figure 1. Block diagram of the wind turbine.

- 2) Fault 2: $\beta_{2,m2}^f = 1.2 * \beta_{2,m2}$ (sensor fault, gain factor) in the time period 2300s-2400s.
- 3) Fault 3: $\beta_{3,m1}^f = 10^\circ$ (sensor fault, fixed value) in the time period 2600s-2700s.
- 4) Fault 4: $\omega_{r,m1}^f = 1.4$ m/s (sensor fault, fixed value) in the time period 1500s-1600s.
- 5) Fault 5: $\omega_{r,m2}^f = 1.1 * \omega_{r,m2}$ (sensor fault, gain factor) and $\omega_{g,m1}^f = 0.9 * \omega_{g,m1}$ (sensor fault, gain factor) in the time period 1000s-1100s.
- 6) Fault 6: parameters in pitch actuator 2 abruptly change from $\{\omega_n, \xi\}$ (pitch actuators are modelled as second order systems) to $\{\omega_{n2}, \xi_2\}$ (actuator fault, changed dynamics), due to a hydraulic pressure drop, at time $t=2900$ s; the initial value is recovered in $t=3000$ s.
- 7) Fault 7: parameters in pitch actuator 3 change from $\{\omega_n, \xi\}$ to $\{\omega_{n3}, \xi_3\}$ (actuator fault, changed dynamics), due to the presence of air un the oil, following a given temporal evolution: change from the initial to the final value linearly over 30s, then maintained during 40s, and finally slowly decreasing up to the initial value during 30s; the fault begins at 3500s and ends at 3600s.
- 8) Fault 8: $\tau_g^f = \tau_g + 2000$ (actuator fault, offset) from 3800s to 3900s.

C. Analytical Redundancy Relations

According to [27], after applying structural analysis [19] with the aid of the SaTool [28] to the set of equations provided in [26], the set of twelve ARR_s given in (24)-(35) can be obtained, where $a_{..}$, $b_{..}$ and $c_{..}$ are model parameters that have to be estimated (η_g is a known coefficient). This is the set of ARR_s that will be used for fault detection and isolation. It must be noticed that a non-linearity is hidden in some of the ARR_s due to the use of the variable τ_r , which is estimated from the wind speed v_w by using the relation

$$\tau_r(k) = \rho \pi R^3 C_q(\lambda(k), \beta(k)) v_w(k)^2 / 2. \quad (23)$$

Moreover, SaTool provides the FSM represented in Table I, which captures the relation between residuals and faults and where a cross 'x' indicates that a given ARR is affected by a given fault (according to the notation used in Section IV-A, $m_{ij} = 1$ where there is a cross, $m_{ij} = 0$ elsewhere).

Notice that this paper is not focused on how to obtain the ARR_s. Structural methods have been actually used to obtain the ARR_s for the example, but other methods could be considered and the proposed ICSP formulation could also be applied to the resulting set of ARR_s. For instance, it would be possible to apply the methodology

Table I
FAULT SIGNATURE MATRIX

ARR	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8
r_1				x	x			
r_2	x	x	x		x	x	x	x
r_3					x			
r_4	x	x	x		x	x	x	x
r_5	x							
r_6	x							
r_7		x						
r_8		x				x		
r_9			x					
r_{10}			x				x	
r_{11}								x
r_{12}						x		

to the complete set of ARR_s obtained for the wind turbine benchmark in [29].

D. Uncertainty and Parameter Estimation

The parameter estimation procedure described in Algorithm 4 has been applied to the fault-free scenario specified in the benchmark in order to obtain the intervals for the parameters of the ARR_s that will be used for fault detection and isolation purposes.

The parameter estimation algorithm requires the a priori knowledge of bounds for the process and measurement noises. For each measured variable, the noise bound has been chosen as the maximum difference, along the fault-free scenario data, between the values provided by the two sensors that measure the variable. On other hand, model errors and the uncertainty in the measurement of the wind speed, which act as a model input, might be represented as process errors. Alternatively, process errors are assumed to be null and the effect of these sources of uncertainty will be captured as parameter uncertainty after the estimation.

Besides the bounds for the noises, the parameter estimation algorithm requires initial intervals for the parameters (initial parameter box Θ_0) that are going to be estimated. These initial intervals could be simply chosen as $[-\infty, +\infty]$, but a different option has been used, based on the results of a nominal parameter estimation. Hence, classical parameter estimation has been applied to estimate some nominal parameters for each ARR (except for the ARR_s that just compare two measurements of the same variable) and then each initial interval has been chosen as $[0, 2 * \theta_{i,nom}]$ if the nominal value for the parameter is positive, or as $[2 * \theta_{i,nom}, 0]$ if the nominal value is negative.

Results of this uncertain parameter estimation are summarized in Table II. For each parameter appearing

$$r_1(k) = \omega_{r,m1}(k) - \omega_{r,m2}(k) \quad (24)$$

$$r_2(k) = \omega_{r,m2}(k) - a_{21}\omega_{r,m2}(k-1) - b_{21}\tau_r(k-1) - c_{21}\tau_{g,m}(k-1) \quad (25)$$

$$r_3(k) = \omega_{g,m1}(k) - \omega_{g,m2}(k) \quad (26)$$

$$r_4(k) = \omega_{g,m2}(k) - a_{41}\omega_{g,m2}(k-1) - b_{41}\tau_r(k-1) - c_{41}\tau_{g,m}(k-1) \quad (27)$$

$$r_5(k) = \beta_{1,m1}(k) - \beta_{1,m2}(k) \quad (28)$$

$$r_6(k) = \beta_{1,m2}(k) - a_{61}\beta_{1,m2}(k-1) - a_{62}\beta_{1,m2}(k-2) - b_{61}\beta_r(k-1) - b_{62}\beta_r(k-2) \quad (29)$$

$$r_7(k) = \beta_{2,m1}(k) - \beta_{2,m2}(k) \quad (30)$$

$$r_8(k) = \beta_{2,m2}(k) - a_{81}\beta_{2,m2}(k-1) - a_{82}\beta_{2,m2}(k-2) - b_{81}\beta_r(k-1) - b_{82}\beta_r(k-2) \quad (31)$$

$$r_9(k) = \beta_{3,m1}(k) - \beta_{3,m2}(k) \quad (32)$$

$$r_{10}(k) = \beta_{3,m2}(k) - a_{101}\beta_{3,m2}(k-1) - a_{102}\beta_{3,m2}(k-2) - b_{101}\beta_r(k-1) - b_{102}\beta_r(k-2) \quad (33)$$

$$r_{11}(k) = \tau_{g,m}(k) - a_{111}\tau_{g,m}(k-1) - b_{111}\tau_{g,r}(k-1) \quad (34)$$

$$r_{12}(k) = P_{g,m}(k) - \eta_g\omega_{g,m2}\tau_{g,m} \quad (35)$$

Table II
UNCERTAIN PARAMETER ESTIMATION

Parameter	Initial interval	Final interval
a_{21}	[0,1.9837]	[0.7107,1.4523]
b_{21}	[0,1.7345e-009]	[0,1.7345e-009]
c_{21}	[0,7.4906e-007]	[0,7.4906e-007]
a_{41}	[0,1.9998]	[0.9785,1.9677]
b_{41}	[0,3.3113e-008]	[2.0303e-008,3.1808e-008]
c_{41}	[-3.2095e-006,0]	[-2.9952e-006,-2.8157e-008]
a_{61}	[0,3.6371]	[0.0834,3.6371]
a_{62}	[-1.6715,0]	[-1.6715,0]
b_{61}	[-2.7357e-004,0]	[-2.7357e-004,0]
b_{62}	[0,0.0345]	[0,0.0345]
a_{81}	[0,3.6395]	[0.0743,3.6395]
a_{82}	[-1.6738,0]	[-1.6738,0]
b_{81}	[0,0.0017]	[0,0.0017]
b_{82}	[0,0.0325]	[0,0.0325]
a_{101}	[0,3.6382]	[0,3.6382]
a_{102}	[-1.6728,0]	[-1.6728,0]
b_{101}	[-2.0893e-004,0]	[-2.0893e-004,0]
b_{102}	[0,0.0347]	[0,0.0347]
a_{111}	[0,1.2131]	[0.5443,0.6727]
b_{111}	[0,0.7869]	[0.3272,0.4556]

in the set of ARRs, the first column indicates the initial interval for this parameter, selected by using the previously described procedure, while the second column indicates the final interval obtained after applying the uncertain parameter estimation algorithm to the fault-free scenario data. As example of the refinement, Figure 2 details the evolution of the estimated intervals for a_{21} .

Finally, it must be noticed that, in order to validate the completeness of the obtained intervals (in fact, the completeness of the initial intervals), the fault detection Algorithm 1 has been applied to the fault-free scenario and it has been verified that no false alarms are reported.

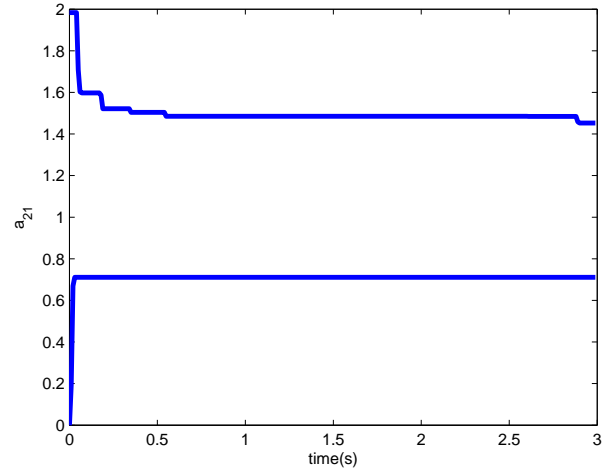


Figure 2. Evolution of the estimation interval for a_{21} provided by the uncertain parameter estimation.

E. Fault Detection and Isolation

The fault detection and isolation procedures summarized in Algorithm 1 and Algorithm 2 have been applied to the eight fault scenarios specified in subsection VII-B.

Satisfactory results (fault detected and isolated) have been obtained in all scenarios except for fault scenario 6, in which the fault effects are small enough to be undetectable given the uncertainty. Table III shows the isolation times obtained in the eight fault scenarios. The results are quite similar (including that Fault 6 is undetected) to the ones reported in [27], obtained using a zonotope set-membership approach. For comparison with other approaches, the reader is referred to papers presented in the two invited sessions about the wind turbine benchmark competition at IFAC 2011.

Table III
FDI RESULTS FOR THE SET OF CONSIDERED FAULT SCENARIOS.

Fault scenario	Fault time	Fault Diagnosis time
1	2000s	2000.04s
2	2300s	2307.33s
3	2600s	2600.03s
4	1500s	1500.07s
5	1000s	1000.01s
6	2900s	Undetected
7	3500s	3534.97s
8	3800s	3800.01s

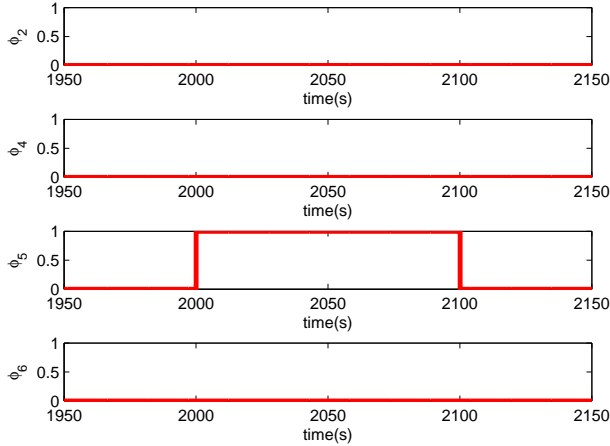


Figure 3. Evolution of ϕ_2 , ϕ_4 , ϕ_5 and ϕ_6 during the appearance of Fault 1.

Next, the results for fault isolation in two of the fault scenarios are discussed in detail. Consider the fault scenario associated to Fault 1. Figure 3 shows the evolution of the components ϕ_2 , ϕ_4 , ϕ_5 and ϕ_6 of the observed fault signature (the ones that are sensitive according to Table I) when Fault 1 appears. Notice that only ϕ_5 is activated (only the ARR r_5 is inconsistent), the other components remain not activated due to the present uncertainty. Remember that this situation is considered by the isolation procedure (Algorithm 2) in such a way that the current fault will not be excluded from the generated diagnosis although other fault candidates may be indicated. In fact, since f_5 is the only fault that affects r_5 , Fault 1 is correctly isolated (it is found as the unique fault candidate). Regarding the temporal behaviour, the fault is correctly isolated once it has been detected at $t=2000.04s$ (four samples after the fault appearance) and since ϕ_5 is active during all the time in which the fault is active, a correct diagnosis is maintained.

As a second example, Figure 4 shows the evolution of ϕ_1 , ϕ_2 , ϕ_3 and ϕ_4 when Fault 5 appears. Notice now that ϕ_3 is active during all the time the fault is present. On

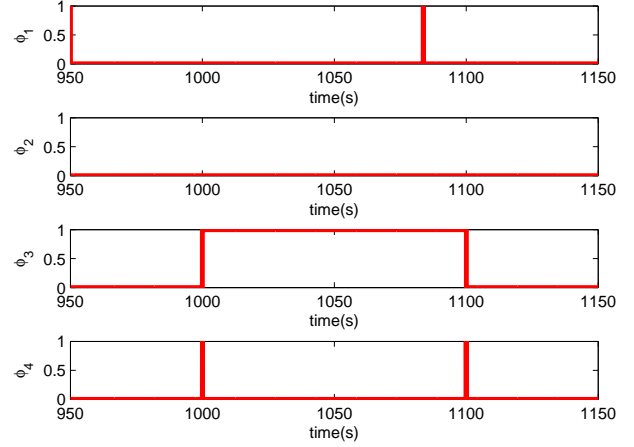


Figure 4. Evolution of ϕ_1 , ϕ_2 , ϕ_3 and ϕ_4 during the appearance of Fault 5.

other hand, ϕ_4 is activated just when the fault appears and when it disappears (this is a common behaviour, some ARRs are specially sensitive to fault magnitude changes), ϕ_1 is activated just for one time instant in the middle of the period of fault activity and ϕ_2 is not activated at all. However, since the isolation algorithm is not sensitive to lost ones in the observed signature fault and since there is no other fault affecting r_3 , Fault 5 is properly isolated.

F. Fault Estimation

Fault 2 is now considered to illustrate the fault estimation procedure. Fault 2 is a gain sensor fault that can be parametrised as $\beta_{2,m2}^f = K_{f_2} * \beta_{2,m2}$. According to the FSM shown in Table I, this fault affects the ARRs given by r_2 , r_4 , r_7 and r_8 . For the sake of simplicity, only r_7 is considered. By including the fault effect in this relation, the following fault dependent ARR is obtained:

$$r_{7,f_2}(k) = \beta_{2,m1}(k) - \frac{1}{K_{f_2}}\beta_{2,m2}(k). \quad (36)$$

The fault estimation procedure summarized in Algorithm 3 has been applied to the Fault 2 scenario data starting from $t = 2300s$. The used initial interval for K_{f_2} is $[0, 2]$, which includes the real fault magnitude ($K_{f_2} = 1.2$). Figure 5 details the evolution of the uncertain fault magnitude estimation along the first ten seconds, showing an important refinement. No further refinement is obtained after processing the rest of the scenario and the obtained final interval for the fault estimation is $[1.1115, 1.3169]$.

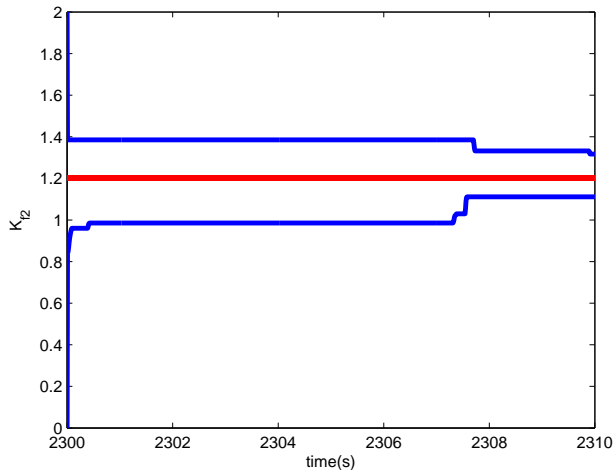


Figure 5. Evolution of the uncertain magnitude estimation for Fault 2.

VIII. CONCLUSIONS

This paper has proposed a new approach for the robust fault diagnosis of non-linear systems in presence of bounded parametric modelling errors and noises, based on using Analytical Redundancy Relations and Interval Constraint Satisfaction techniques.

The proposed fault detection procedure checks the consistency between the measurements using a set of Analytical Redundancy Relations (ARRs) derived from a normal operation model of the monitored system. In presence of bounded uncertainty, this consistency checking is formulated as an Interval Constraint Satisfaction Problem (ICSP) that can be solved using appropriate solvers. Accordingly to the bounded nature of the uncertainty, the method assures the elimination of false alarms but faults whose effects do not exceed the level of uncertainty will remain undetected. The fault isolation procedure, executed when a fault has been detected, is based in the solution of several ICSPs, one for each ARR, and in the implementation of a binary diagnostic reasoning that takes into account the fact that due to the uncertainty some of the ARRs that are sensitive to the present fault may be found consistent. The procedure may return several fault candidates, but it assures that the real fault is one of them. Finally, the fault estimation procedure uses a set of fault-dependent ARRs and again an ICSP formulation to estimate the actual fault magnitude. Moreover, the paper addresses the problem of determining the uncertainty in the parameters of the used uncertain ARRs. The proposed procedure is similar to the one used for fault estimation and it is applied off-line to data collected in fault-free operation of the system.

The usefulness of the proposed approach is illustrated through its application to the well known wind turbine benchmark. Satisfactory results have been obtained compared to those reported in the literature.

There are two main directions for future research. First, the multiple fault assumption has to be considered. Although algorithms for multiple fault isolation are available in the literature, the problem of estimating the magnitude of simultaneous faults in presence of uncertainty has to be studied. In particular, it is necessary to study under which conditions it is possible to identify and quantify the contribution of each fault. Second, fault isolability is currently limited by the use of a standard binary FSM. According to [30], information about the sensitivities of the ARRs against the faults and/or information about the dynamics of the ARRs during the appearance of faults can be used to improve fault isolation and estimation. This will be integrated with our approach in the future.

REFERENCES

- [1] J. Chen and R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, 1999.
- [2] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, 1998.
- [3] J. Chen, R. Patton, and H. Zhang, "Design of unknown input observers and robust fault detection filters," *International Journal of Control*, vol. 63, no. 1, pp. 85–105, 1996.
- [4] M. Basseville and I. Nikiforov, *Detection of abrupt changes: theory and applications*. Prentice Hall, 1993.
- [5] V. Puig, J. Quevedo, T. Escobet, and S. De las Heras, "Robust fault detection approaches using interval models," *Proceedings of IFAC World Congress*, 2002.
- [6] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, *Bounding Approaches to System Identification*. Plenum Press, 1996.
- [7] M. Witczak, J. Korbicz, and R. Patton, "A bounded-error approach to designing unknowninput observers," *Proceedings of IFAC World Congress*, 2002.
- [8] E. Gelso, E. Frisk, and J. Armengol, "Robust fault detection using consistency techniques with application to an automotive engine," in *Proceedings of IFAC World Congress*, Seoul (Korea), 2008.
- [9] S. Tornil-Sin, C. Ocampo-Martinez, V. Puig, and T. Escobet, "Robust fault detection of non-linear systems using set-membership state estimation based on constraint satisfaction," *Engineering Applications of Artificial Intelligence*, vol. 25 (1), pp. 1–10, 2012.
- [10] B. Rinner and U. Weiss, "Online monitoring by dynamically refining imprecise models," *IEEE Trans. Syst., Man, Cybern.*, vol. 34, pp. 1811 – 1822, 2004.
- [11] M. Staroswiecki and G. Comtet-Varga, "Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems," *Automatica*, vol. 37, no. 5, pp. 687–699, 2001.
- [12] S. Ploix and O. Adrot, "Parity relations for linear uncertain dynamic systems," *Automatica*, vol. 42, p. 1553 1562, 2006.
- [13] M. Staroswiecki and P. Declerck, "Analytical redundancy in non-linear interconnected systems by means of structural analysis," *IFAC/IMACS/IFORS Conference AIPAC'89*, 1989.

- [14] B. Pulido and C. Alonso, "Possible conflicts, arrs, and conflicts," *International Workshop on Principles of Diagnosis*, pp. 122–128, 2002.
- [15] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics*. London: Springer-Verlag, 2001.
- [16] E. Hyvnen, "Constraint reasoning based on interval arithmetic: The tolerance approach," *Artificial Intelligence*, vol. 58, pp. 71–112, 1992.
- [17] L. Granvilliers and F. Benhamou, "Realpaver: An interval solver using constraint satisfaction techniques," *ACM Transactions on Mathematical Software*, vol. 32, no. 1, pp. 138–156, 2006.
- [18] E. Chow and A. Willisky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Transactions on Automatic Control*, vol. 29, no. 7, pp. 603–614, 1984.
- [19] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*, 2nd ed. Berlin, Heidelberg: Springer-Verlag, 2006.
- [20] M. Cordier, P. Dague, F. Lévy, J. Montmain, M. Staroswiecki, and L. Travéé-Massuyès, "Conflicts versus analytical redundancy relations: A comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, vol. 34, no. 5, pp. 2163–2177, 2004.
- [21] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, New Jersey: Prentice Hall, 1987.
- [22] M. Kendall and A. Stuart, *The Advanced Theory of Statistics*. Charles Griffin, 1979, vol. 2.
- [23] M. Dalai, E. Weyer, and M. Campi, "Parametric identification of nonlinear systems: guaranteed confidence regions," in *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 2005, pp. 6418–6423.
- [24] W. Reinelt, A. Garulli, and L. Ljung, "Comparing different approaches to model error modeling in robust identification," *Automatica*, vol. 38, no. 5, pp. 787–803, 2002.
- [25] L. Jaulin, M. Kieffer, I. Braems, and E. Walter, "Guaranteed nonlinear estimation using constraint propagation on sets," *International Journal of Control*, vol. 74, no. 18, pp. 1772–1782, 2001.
- [26] P. 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*, 2009, pp. 155–160.
- [27] J. Blesa, V. Puig, J. Romera, and J. Saludes, "Fault diagnosis of wind turbines using a set-membership approach," in *Proceedings of the 18th IFAC World Congress*, 2011.
- [28] M. Blanke and T. Lorentz, "Satool: A software tool for structural analysis of complex automation systems," in *6th IFAC SAFEPROCESS*, 2006, pp. 673–678.
- [29] C. Svard and M. Nyberg, "Automated design of an fdi-system for the wind turbine benchmark," in *Proceedings of the 18th IFAC World Congress*, 2011, pp. 8307–8315.
- [30] J. Meseguer, V. Puig, and T. Escobet, "Fault diagnosis using a timed discrete-event approach based on interval observers: Application to sewer networks," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 40, no. 5, pp. 900–916, 2010.



Sebastian Tornil-Sin received the Computer Engineering degree and the Ph.D. degree in Control, Vision and Robotics from the Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 1996 and 2006, respectively. He is currently Assistant Professor at the Automatic Control Department (ESAI) at UPC and member of the Advanced Control Systems research group. His main research interests include the application of Interval Analysis in the Automatic Control field and the Fault Diagnosis and Fault Tolerant Control of dynamic systems.



Carlos Ocampo-Martinez received his Electronic Engineering degree and his MSc degree in Industrial Automation from the National University of Colombia, Campus Manizales, in 2001 and 2003, respectively. In 2007, he received his PhD Degree in Control Engineering from the Technical University of Catalonia (Barcelona, Spain). After a year as postdoctoral fellow of the ARC Centre of Complex Dynamic Systems and Control (University of Newcastle, Australia), he was with the Spanish National Research Council (CSIC) at the Institut de Robòtica i Informàtica Industrial (IRI) in Barcelona as a *Juan de la Cierva* research fellow. Since 2011, he is Assistant Professor at the Technical University of Catalunya, Automatic Control Department (ESAI). His main research interests are in the areas of constrained model predictive control, large-scale systems management, nonlinear dynamics and industrial applications.



Vicenç Puig received the Telecommunications Engineering degree and the Ph.D. degree in control engineering from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 1993 and 1999, respectively. He is currently an Associate Professor of automatic control and the Leader of the Advanced Control Systems Research Group, Universitat Politècnica de Catalunya, Terrasa, Spain. He is also with the Institut de Robòtica i Informàtica Industrial (CSIC), Universitat Politècnica de Catalunya, Barcelona. He has been involved in several European projects and networks and has published several papers in international conference proceedings and scientific journals. His main research interests are fault detection and isolation and fault-tolerant control of dynamic systems.



Teresa Escobet received a degree in Industrial Engineering at Universitat Politècnica de Catalunya (UPC) in 1989 and PhD at the same University in 1997. She began to work at UPC as an Assistant Professor in 1986 and she became Associate Professor in 2001. Her teaching activities are related to issues of Automatic Control. She is a member of the research group Advanced Control Systems (SAC). Her main research interests are in dynamic system modelling and identification applied to fault detection, isolation, fault-tolerant control and condition-based maintenance. She has been involved in several International and national research projects and networks. Recent publication includes 13 articles in journals and 100 papers at international conferences.