SIMULINK BLOCKSET FOR FAULT DETECTION USING INTERVAL MODELS

S. Tornil, T. Escobet and V. Puig

Dept. ESAII, Universitat Politécnica de Catalunya
Rambla Sant Nebridi 10, 08222 Terrassa (SPAIN)
Tel. +34 93 7398621, Fax. +34 93 7398628
E-mail: {stornil,vpuig}@esaii.upc.es, teresa@bages.eupm.upc.es

Abstract: This paper describes a tool to aid in the analysis and design of fault detection systems for processes which present bounded structured uncertainty (only bounds for the values of their physical parameters are known). This uncertainty is considered in the fault detection methodology by using interval models to represent the processes. The tool has been implemented using the well-known Matlab/Simulink CACSD framework. This framework is very useful in the analysis and design of control systems and can also be used in the analysis and design of supervisory systems (which must include fault detection capabilities).

Keywords: Fault detection, uncertain dynamic systems, intervals, optimisation, software tools.

1. INTRODUCTION

The detection and isolation of faults (FDI) in complex industrial systems is one of the most important tasks assigned to the computers supervising such systems. The early indication of incipient faults is critical in avoiding performance degradation, product deterioration, major system damage and loss of safety in safety-critical systems. The quick and correct isolation of the faulty component allows that appropriate actions can be applied as soon as possible.

In model-based FDI methods, the actual behaviour of the system is checked for consistency with a mathematical model that describes its non-faulty operation (analytical redundancy). Consistency checking is normally achieved through a comparison between the measured value of a system variable and the estimated value obtained using the system model.

In most practical applications, it is not possible to obtain an accurate and complete model of the process (system), due to some kind of uncertainty in the available knowledge about it. Sometimes it is not possible to know the whole structure of the process (unstructured uncertainty) and sometimes the structure is known but the value of the model parameters is not exactly known (structured uncertainty). When bounded structured uncertainty is present, interval models appear as a natural framework to deal with it.

A Simulink blockset has been developed to manage interval models. The manipulation of such models involves the solution of some optimisation problems. The 'Optimization Toolbox' included in Matlab implements the classical optimisation algorithms, but these algorithms only present local convergence. In the actual version of the 'blockset', the GIA InC++ library is integrated into the Matlab/Simulink framework and its algorithms with global convergence are used.

The use of the implemented blocks can help in the analysis and design of fault detection systems for systems that present bounded structured uncertainty. The final goal behind this work is to provide a tool that allows the analysis and design of supervisory and control systems using a common framework, the well-known Matlab/Simulink CACSD framework.

The paper has the following structure. An introduction to basic model-based fault detection techniques and their extension to interval models are presented in section 2. The developed blockset, based on these techniques, is described in section 3. The results of its application to the detection of faults in an example system, a DC motor, is presented in section 4. Finally, some future work is discussed in section 5.

2. MODEL-BASED FAULT DETECTION USING INTERVAL MODELS

2.1. Structure of model-based fault FDI.

Most model-based FDI methods present the two-stage structure (Chow and Willsky, 1984) shown in figure 1.

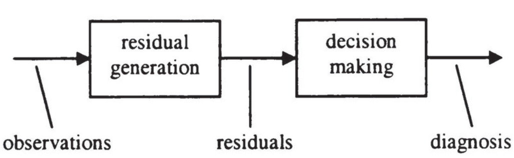

Fig. 1. General structure of model-based FDI.

Residual generation is based on the concept of analytical redundancy, which consists in the comparison of values measured using sensors with analytically computed values for the respective variables, obtained from measures of other variables and/or from previous measures of the same variable by using the relations provided by the process model. The idea can be extended to the comparison of analytically computed values obtained starting from different sets of measures (Gertler, 1998). In both cases, the resulting differences are called residuals and are indicative of faults in the system. Under ideal conditions, residuals are zero in the absence of faults and non-zero when a fault is present.

In practical situations, disturbances (inputs not considered by the model), noise and unavoidable modelling errors (differences between process and model parameters) lead the residuals to non-zero values even in the absence of faults. Therefore, the residual evaluation must contain a testing of the residuals against thresholds to avoid false alarms; these thresholds can be obtained empirically or by theoretical considerations.

Fault detection (FD) task is accomplished by the residual generation and testing of each residual against its threshold.

2.2. Residual generation.

Modern fault detection systems are implemented on digital systems. For this reason they work with sampled variables and the models that they use to implement the analytic redundancy schema correspond to the discrete equivalent representation of the process, which is normally continuous. Discrete-equivalent models of a continuous system can be obtained using the Z-transform (Gertler, 1998).

A SISO (single-input single-output) linear time-invariant discrete model is characterised by a difference equation of the form:

$$y(k)+\sum_{i=1}^{n} a_i * y(k-i) = \sum_{j=0}^{m} b_j * u(k-j) \qquad (1)$$

where n is the system order and m<=n.

Using the model defined in (1), two alternatives can be considered to estimate the actual value of y(k): use the measured values for precedent system outputs (y(k-1),..., y(k-n)) or use their previously estimated values. The first option is called output estimation or output prediction and the second one is called simulation. The obtained expressions for both options are, respectively:

$$\tilde{y}(k) = \sum_{i=1}^{n} -a_i * y(k-i) + \sum_{j=0}^{m} b_j * u(k-j) \qquad (2)$$

$$\tilde{y}(k) = \sum_{i=1}^{n} -a_i * \tilde{y}(k-i) + \sum_{j=0}^{m} b_j * u(k-j) \qquad (3)$$

Residuals can be directly obtained from the precedent expressions, called parity relations or parity equations, as the difference between the measured and the estimated values for the output:

$$r(k) = y(k) - \tilde{y}(k) \qquad (4)$$

These residuals are called MA (moving average) primary residuals and ARMA (auto-regressive moving average) primary residuals, respectively (Gertler, 1998).

But more parity relations can be obtained from (1). Applying (1) in a recursive way, expressions of y(k) as function of y(k-1-L), ...,y(k-n-L) and u(k), ..., u(k-m-L), where L is the number of times that (1) is recursively used, are obtained. The general expression for these relations is the following:

$$y(k)=c*\begin{pmatrix} A^{L+1}*x(k-1-L) \\ +A^{L}*B*v(k-1-L) \\ +... \\ +A*B*v(k-2) \\ +B*v(k-1) \end{pmatrix}+d*v(k) \quad (5.1)$$

where:

$$x(k)=\begin{pmatrix} y(k) \\ ... \\ y(k-n) \end{pmatrix},\ v(k)=\begin{pmatrix} u(k) \\ ... \\ u(k-m) \end{pmatrix},$$

$$A=\begin{pmatrix} -a_1 & -a_2 & ... & -a_{n-1} & -a_n \\ 1 & 0 & ... & 0 & 0 \\ 0 & 1 & ... & 0 & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & ... & 1 & 0 \end{pmatrix},$$

$$b=\begin{pmatrix} b_1 \\ ... \\ b_m \end{pmatrix}, \quad (5.2)$$

$$c=\begin{pmatrix}1 & 0 & ... & 0\end{pmatrix},$$
$$d=\begin{pmatrix}b_0 & 0 & ... & 0\end{pmatrix}$$

These expressions are called L^{th} order parity relations and can be used to obtain L^{th} order MA and ARMA residuals, by taken $x(k)=(y(k),...,y(k-n))^T$ and $x(k)=(\bar{y}(k),...,\bar{y}(k-n))^T$, respectively.

2.3. Robustness.

Robustness in model-based detection systems is the property of operate in the presence of disturbances, noise and modelling errors whilst maintaining sensitivity to faults (Chen and Patton, 1999). The goal of robust model-based detection methods is to reduce simultaneously false and missing alarms (undetected faults).

De-coupling techniques can be applied in the residual generation step to reduce the effect of the disturbances in the residuals. On the other hand, when the statistical properties of noise are known, effective techniques based on (temporal sliding) window averaging and statistical testing can be applied in the decision-making stage to deal with it.

But the most crucial point in model-based FDI is robustness against modelling errors (Chen and Patton, 1999; Gertler, 1998). Robustness against modelling errors can be considered at the residual generation stage (active robustness) and at the decision making stage (passive robustness) (Chen and Patton, 1999; Puig et al., 1997).

Active robustness techniques are de-coupling techniques that try to eliminate the effects of modelling errors in the obtained residuals. But de-coupling from modelling errors presents two main drawbacks. The first and main problem (Horak, 1988) is that de-coupling resolve the problem of modelling errors by avoiding rather than accounting for them. Obtained residuals are insensitive to uncertain parameters and then also insensitive to faults in these parameters that must be detected (modelling uncertainties and multiplicative errors are both changes in the parameters of the system). The second problem is that there is no a residual generation algorithm which is robust under arbitrary model error conditions (Gertler, 1998). Perfect de-coupling is only possible under certain restrictive conditions and only approximate de-coupling can be obtained in most cases.

In practical applications, in the presence of noise, disturbances and modelling errors, perfect robust residual generation can not be obtained and residuals are non-zero even in the absence of faults. In this case, threshold values are set equal to the maximum evaluation of their associated residuals in the non-faulty situation. Then, false alarms are avoided and missing alarms are minimised.

When perfectly robust residual generation can not be obtained, passive robustness techniques can be applied to enhance the robustness of the FD system. One way to enhance robustness against modelling errors is based on the use of adaptive thresholds (Horak, 1988; Chen and Patton, 1999).

When perfect de-coupling from modelling errors can not be obtained, the magnitude of the residuals is function of the process input. In this case, the use of fixed thresholds large enough to avoid false alarms when the input is high can result in missing faults when the input is low. In general, there is not a fixed threshold that gives tolerable false and missing alarm rates. Better results can be obtained using adaptive thresholds, which vary according to the state and the inputs of the system. The advantages of using adaptive thresholds are shown in figure 2.

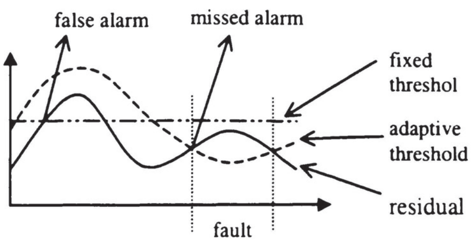

Fig. 2. Fixed vs. adaptive threshold.

2.4. Interval models.

When structured uncertainty is present in the knowledge of a physical system, sometimes the bounds for this uncertainty are known. This is the case, for example, when the component tolerance is known or when bounded approaches to system identification (Milanese et al., 1996) are used. In these cases, interval models appear as a natural framework to represent the system.

Interval models are models such as their parameters are intervals instead of precise values. An example of a continuous transfer function can be the following:

$$G(s) = \frac{[2,3]}{[1,2]s+1} \tag{6}$$

The response of these models to a certain temporal input is defined by two curves, called envelopes (Armengol et al., 1999; Puig et al., 1997), which determine at each time instant the maximum and minimum achievable values by the outputs of the precise models defined by the interval one (those whose their precise parameters lie inside the interval parameters of the interval model). As example, the step response of the interval system defined in (6) is shown in figure 3.

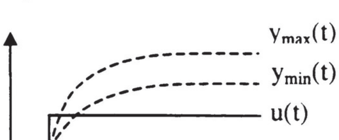

Figure 3. Envelopes for an interval system.

The envelopes give, at every time instant, the maximum effect of the uncertainty in the system output. Hence, they define the optimal (adaptive) thresholds that avoid false alarms due to uncertainty and minimise missing alarms.

2.5. Extension of the residual generation techniques to interval models.

Difference equations of the form given in (1) can also be obtained when bounded structured uncertainty is present in the system. But an interesting effect must be taken into account. If the uncertainty in the physical parameters of the system is directly known (i.e. the nominal values and tolerances of the components of an electrical circuit), then this uncertainty can be directly used in (1) by expressing the coefficients $a_1,\ldots, a_n, b_0,\ldots, b_m$ as function of the intervals for the physical parameters. But, if bounding approaches for parameter estimation are used, an overbounded description of the real uncertainty is obtained, due to limitations of the interval mathematics. The increment of uncertainty is bigger using discrete-time parameter estimation than using continuous-time parameter estimation. This increment of uncertainty leads to an increment of the possible missing alarms. For more details about this effect, see (Tornil et al., 2000).

Envelopes for uncertain systems can be obtained using the MA or the ARMA parity relations defined by (5). These two options are called interval prediction and simulation with sliding window, respectively. In both cases, the envelope generation problem is transformed into two optimisation problems. The general expression for both options is the following:

$$\tilde{y}(k) = {}^{min}_{max}\left\{ c * \begin{pmatrix} A^{L+1} * \dot{x}(k-1-L) \\ + A^{L} * B * v(k-1-L) \\ + \ldots \\ + A * B * v(k-2) \\ + B * v(k-1) \end{pmatrix} + d * v(k) \right\} \tag{7}$$

where $x(k) = (\tilde{y}(k),\ldots, \tilde{y}(k-n))^T$ in interval simulation and $x(k) = (y(k),\ldots, y(k-n))^T$ in interval prediction.

Considering that the monitored system is linear time-invariant (LTI), the parameters that appear in the expressions to optimise in different instant times correspond to the same values. This fact has an implication in interval simulation: the results of successive optimisations are not independent. But the problem is that they are considered as independent using the sliding window technique. The results for the envelopes obtained in a time instant k_0 correspond to a certain set of values of the parameters; these results are used as starting point in time instant k_0+L and the new results can correspond to a different set of values of the parameters. Hence, in the general case, the obtained envelopes are overbounded and their values can't be generated by an LTI system. In some cases these envelopes can be even divergent (Puig et al., 1997). As bigger is the window length less effect have the precedent problem and closer are the obtained envelopes to the good ones. It is demonstrated (Saludes et al, 1997) that there exist a minimum value for L such the obtained envelopes are stable (not divergent) and another one such the obtained envelopes present slight errors.

The precedent problem does not appear in interval prediction because the use of measures "breaks" the dependence beetween the results of successive optimisations.

Interval prediction, due to the fact that uses more information, produces more restrictive envelopes. This provides to this method a better sensitivity respect faults (less possible missing alarms) and a less detection time. On the other hand, it presents a problem that must be considered. If in a determined instant of time a fault appears in the system, the output y(k) is not valid any more, although it is possible that it remains inside the envelopes. When this happens, then the prediction algorithm gives the output as good and uses its value as input in a posterior prediction step. Therefore, it could be possible that the obtained envelopes "follow" and cover the system output even in a faulty situation.

The problem of envelope generation is expressed as two optimisation problems. Hence, the characteristics of the optimisation method used determines the obtained envelopes. Classical optimisation methods (gradient descent based) methods only assure local convergence and their use can produce underbounded envelopes. Modern methods based on interval arithmetic and 'brand and bound' strategies (Hansen, 1992) can be used to assure global convergence.

Fault detection using interval simulation is proposed in (Puig et al.,1997). On the other hand, interval prediction using multiple sliding time windows is proposed in (Armengol et al., 2000). A hybrid approach is used in the detection module of the CA~EN FDI system (Travé-Massuyès and Milne, 1997). In this system, interval prediction with L=1 is used while the system output is inside the envelopes and a change to interval simulation is applied when it goes outside. This change is used to avoid the "fault following" problem when it has been detected that the system can be faulty.

3. THE SIMULINK BLOCKSET

The implemented blockset contains blocks that implement interval simulation and interval prediction for first order and second order, continuous and discrete, dynamic systems with interval parameters. A general view of these blocks is included in figure 4.

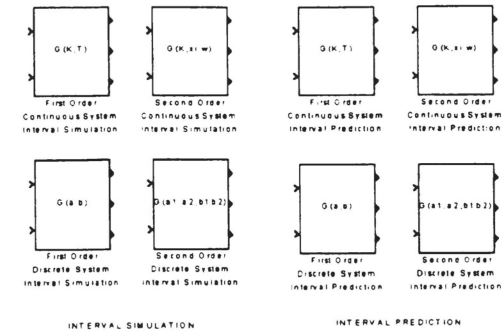

Fig. 4. Simulink blockset for envelope generation.

Every block in the blockset has two inputs and three outputs. The inputs must be connected to the output and to the input of the block that simulates the monitored process. First two outputs are the obtained envelopes and the third one is a logical signal that indicates if the output of the process block is inside (no fault) or outside (fault) these envelopes. In order to generate this signal, the output of the process block is needed as input even for the simulation-based envelope generation blocks.

A Simulink example for a second order continuous system is shown in figure 5. The envelopes obtained using interval simulation with L=10, when the coefficients of this system are K=1, ξ=[0.35,0.65] and ω=1, are shown in figure 6.

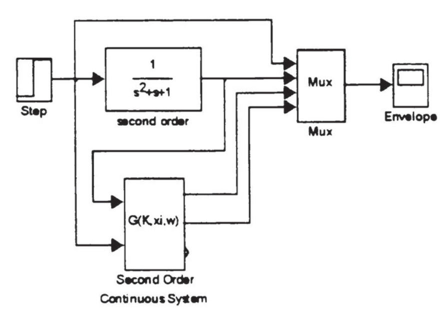

Fig. 5. Example of use of an envelope generation block.

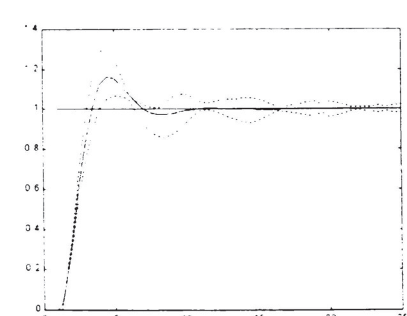

Fig. 6. Obtained result for the example of fig. 5.

Two versions of the blockset, using different optimisation tools have been developed. The classical optimisation methods implemented in the Matlab 'Optimization Toolbox' (The Mathworks, 1998) were used in the initial version. In a newer version, the algorithms based on interval arithmetic and branch and bound implemented in the GIA In C++ library (Delisoft, Ltd, 1999) were used to avoid the problems with local minima. The results presented in this paper have been obtained using this second version.

4. EXAMPLE

As example of use of the blockset for fault detection, the results of its application to the detection of some faults in a DC motor are shown in this section.

Consider the schema of a DC motor with a mechanical load shown in figure 7. The physical meaning of the parameters and the values used in the example (considering uncertain load parameters) are the following:

- R_a: armature resistance; 1 Ω
- L_a: armature inductance; 0.5 H
- K_b: back emf constant; 0.01 V/rad/sec
- K_i: torque constant; 0.01 (N-m/amp)
- J_m: moment of inertia of the rotor; 0.01 kg-m^2
- b_m: damping ratio of the rotor; 0.1 N-m/rad/sec
- N/M: reduction constant; 1/10
- J_l: moment of inertia of the load; [20,30] kg-m^2
- b_l: damping ratio of the load; [10,15] N-m/rad/sec

In the armature controlled configuration (i_f generates a constant magnetic flow and the speed is controlled by the armature voltage), the transfer function, where the rotation speed ($w_l(t)$) is the output and the armature voltage ($e_a(t)$) is the input, is the following (Ogata, 1996):

$$G(s)=\frac{\dfrac{K_i*\frac{N}{M}}{L_a*J_{eq}}}{s^2+\dfrac{L_a*b_{eq}+R_a*J_{eq}}{L_a*J_{eq}}+\dfrac{R_a*b_{eq}}{L_a*J_{eq}}} \tag{8.1}$$

where:

$$J_{eq}=J_m+\left(\frac{N}{M}\right)^2*J_l \tag{8.2}$$

$$b_{eq}=b_m+\left(\frac{N}{M}\right)^2*b_l$$

Three faults have been considered:

- actuator fault: bias equal to 1 volt (additive)
- process fault: an increment of 20% in the torque constant (multiplicative)
- sensor fault: a bias equal to $5*10^{-3}$ (additive)

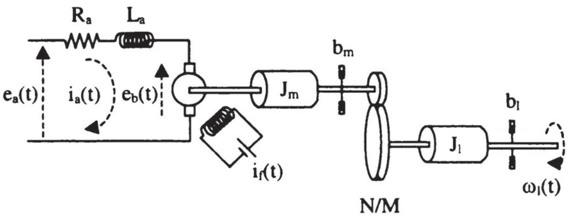

Fig. 7. Schema of the DC motor with a mechanical load.

The fault scenarios used are defined by the following considerations:

- the input applied to the system is a step of amplitude 5 activated in t=1 s.
- the fault appear in t=2 s.
- single fault condition (only one faulty component) is assumed.

The results obtained in the non-faulty case and in the described faulty situations are presented in the following figures. In each figure, the first graph shows the system output and the envelopes obtained using interval simulation with L=10 (dashed), interval prediction with L=1 (dotted) and interval prediction with L=3 (dot-dash). The other three graphs show when the system output is outside of these the envelopes.

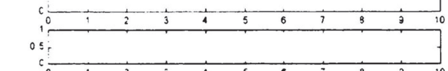

Fig. 8. Results in the non-faulty case.

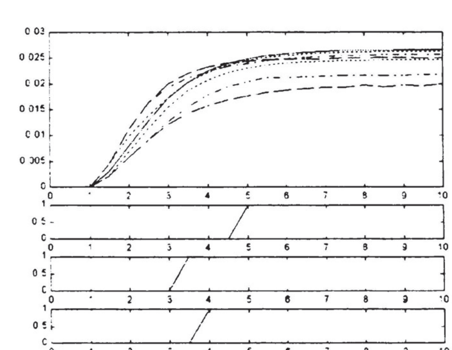

Fig. 9. Result for the actuator fault.

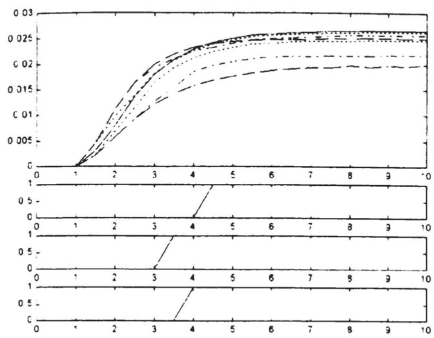

Fig. 10. Result for the process fault.

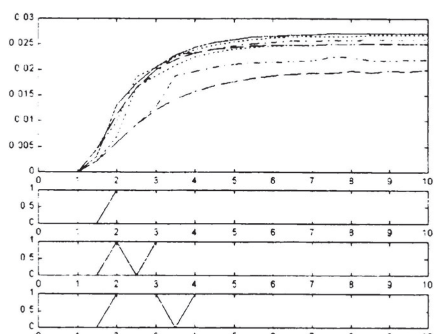

Fig. 11. Result for the sensor fault.

5. FUTURE WORK

Two main limitations in the actual version of the blockset must be considered for the future work:

- the uncertainty is only considered in the system parameters; uncertainty can also be considered in the measured variables (system inputs and outputs).
- only blocks for SISO systems have been implemented; an extension to MIMO systems can be considered.

REFERENCES

Armengol, J., L. Travé-Massuyès, J. Vehí, J.L. de la Rosa (1999). A survey of interval model simulators and their properties related to fault detection. In: 14th IFAC World Congress, Beijing, China.

Armengol, J., J. Vehí, L.Travé-Massuyès and M.A. Sainz (2000). Interval Model-Based Fault Detection using Multiple Sliding Time Windows. In: *Preprints of the 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes – SAFEPROCESS 2000* , Budapest, Hungary, pp. 168-173.

Chen, J. and R.J. Patton (1999). *Robust Model-Based Fault diagnosis*. Kluwer Academic Publishers, Boston.

Chow, E.Y. and A.S. Willsky (1984). Analytical Redundancy and the Design of Robust Failure Detection Systems. *IEEE Transactions on Automatic Control*, **AC-29**, pp. 603-614.

Delisoft, Ltd (1999). *GIA InC++ User's Manual.*

Gertler, J.J. (1998). *Fault detection and Diagnosis in Engineering Systems*. Marcel Dekker, New York.

Hansen, E. (1992). *Global Optimization Using Interval Analysis*. Marcel Dekker, New York.

Horak, D.T. (1988). Failure Detection in Dynamic Systems with Modeling Errors. *Journal of Guidance, Control and Dynamics*, **11**, pp. 508-516.

Milanese, M., J. Norton, H. Piet-Lahanier, E. Walter (1996). *Bounding Approaches to System Identification*. Plenum Press, New York and London.

Ogata, K. (1996). *Modern Control Engineering*. Prentice Hall.

Puig, V., J. Saludes and J. Quevedo. (1997). Applications in fault detection and diagnosis of a new algorithm for adaptive threshold generation. In: *TEMPUS Workshop MODIFY'97*.

Saludes, J., V. Puig and J. Quevedo. (1997). Determination of window length for a new algorithm in adaptive threshold generation. In: *TEMPUS Workshop MODIFY'97*.

The MathWorks, Inc. (1998). *Optimization Toolbox Users' Manual*.

Tornil, S., T. Escobet and V. Puig (2000). Fault Detection Using Interval Models. In: *Preprints of the 4th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes – SAFEPROCESS 2000*, Budapest, Hungary, pp. 1180-1186.

Travé-Massuyès, L. and R. Milne (1997). TIGERTM: Gas turbine condition monitoring using qualitative model based diagnosis. *IEEE Expert Intelligent Systems & Applications*, May-June 1997, pp. 22-31.