Computación y Sistemas Vol. 5 No. 2 pp. 128 - 143
© 2001, CIC - IPN. ISSN 1405-5546 Impreso en México

The Ca~En Diagnosis System and its Automatic Modelling Method

El Sistema de Diagnóstico Ca~En y su Método Automático de Modelización

Louise Travé-Massuyès[1], **Teresa Escobet**[2], **Renaud Pons**[1] and **Sebastián Tornil**[2]
[1]LAAS-CNRS and LEA-SICA
7 avenue du Colonel Roche, 31077 Toulouse, France
[2]UPC and LEA-SICA, Automatic Control Departament,
Rambla Sant Nebridi 10, 08222 Terrassa, Spain
e-mail : {louise, rpons}@laas.fr, {teresa, stornil}@esaii.upc.es

Article received on May 30, 2001; accepted on September 25, 2001

Abstract

Ca~En is a causal model based diagnosis system that includes a fault detection module and a fault isolation module. Both are based on models of the system at different levels of abstraction. Model based methods obviously rely on the quality of the models. This paper focuses on recent results about the Ca~En underlying modelling methodology. Each step of the modelling method is presented: the automatic generation of a causal structure from a component-oriented equation model and how to get the parameters of the causal influences, then how to automatically derive the operational detection models. The method is applied to the gas fuel system of a Frame 6 turbine of National Power (UK).

Keywords: Model-based reasoning, diagnosis, interval models, industrial applications

Resumen

Ca~En es un sistema de diagnosis basado en modelos que incluye un módulo para la detección de fallos y un módulo de aislamiento. Ambos se basan en modelos del sistema a diferentes niveles de abstracción. Obviamente los métodos basados en modelos se sustentan en la calidad de los modelos. Este artículo se centra en resultados recientes del método de modelización subyacente en Ca~En. La metodología de modelado se presenta con: la generación automática de una estructura causal a partir de ecuaciones orientadas por componentes y como obtener los parámetros de las influencias causales, y a continuación el cómo de forma automática deducir los modelos de detección operacionales. Este método se ha aplicado a un sistema de inyección de combustible en una turbina de gas tipo Frame 6 de National Power (UK).

Palabras clave: Razonamiento basado en modelos, diagnosis, modelos intervalares, aplicaciones industriales.

1 Introduction

Model based and qualitative reasoning technologies have advanced to a mature state. They have been shown to be capable of helping with many real complex problems (Struss *et al.*, 2000) (Travé-Massuyès and Milne, 1997). Several real world products and advanced systems are now available and many advanced prototypes have been demonstrated (MONET-ILC, 1998) (Cauvin *et al.*, 1998). The gaps between research and industry have been clearly identified and should help to progressively close them (Travé-Massuyès and Milne, 1998).

Diagnosis has been a major focus for the application of model based and qualitative reasoning technologies, taking significant advantages from the two main key ideas: the first one is the separate representation of process knowledge and task knowledge, and the second is the representation of process knowledge at a sufficient level of abstraction.

Although basing the reasoning on a model has many advantages, it emphasis the key role of the modelling process. As a matter of fact, modelling any nontrivial system is a complex task and is never easy. Tools for automated model building are still missing and today, modelling is often a specialised, hand crafted process dependent on the model based environment to be used. However one aspect of automated modelling which consists transforming a model in a given form into another form which is more adequate for solving the task at hand has deserved a lot of attention in the last few years. Just to mention a few of them, causal ordering techniques allow one to derive the causal structure of a model given as a set of algebraic and differential equations (Iwasaki and Simon, 1994) (Travé-Massuyès and Pons, 1997). The causal structure is then highly valuable for explanation and diagnosis purposes. It has been shown that automata models can be used to generate chronicles representing faulty or normal situations (Bibas *et al.*, 1996), the chronicles can then be used as reference in a chronicle recognition approach (Dousson *et al.* , 1993).

The Ca~En system modelling methodology takes advantage of automated modelling methods. Let us recall that the extension of the causal ordering method (Iwasaki and Simon, 1994) to hybrid systems was proposed by (Travé-Massuyès and Pons, 1997) as a requirement for Ca–En modelling. In addition to this method, other modelling features have been added which are reported in this paper. Each step of the modelling method is presented: the automatic generation of a causal structure from a component-oriented equation model and how to get the parameters of the causal influences, then how to automatically derive the operational detection models.

The method is applied to the gas fuel system of a Frame 6 turbine of National Power (UK). Part of this work was performed within the framework of the TIGER and TIGER SHEBA european projects.

2 The Frame 6 Turbine Gas Fuel System (GFS)

2.1 Description

The gas turbine control system controls the shaft speed, modifying the gas fuel flow reference. For this reason, one of the critical parts of the turbine is the Gas Fuel System supply (GFS). Indeed, the performance and efficiency of the turbine highly depends on an accurate control of the fuel input. This is just the task of the GFS.

The main components of the GFS are two actuators: the Stop Ratio Valve (SRV) and the Gas Control Valve (GCV). These valves are series connected and control the flow of gas fuel that enters in the combustion chambers. The first of these valves, the SRV, is controlled by a feedback loop that maintains constant the gas pressure at its output (pressure between the two valves) *fpg2*. This pressure being constant, the gas fuel flow is just determined by the position of the GCV. Hence, the GCV is a position controlled valve.

The SRV and GCV valves, and their associated feedback loops are shown in the figure 1. Both valves have been analysed from two different criteria: the hydraulic (h) and the mechanical part (m). The first one includes the valve seat and related components which determine the fuel flow throw the valve and the second one includes all the components which control the valve position, i.e. all the valve mechanical components (diaphragm, spring, ..), pneumatic servomotor, controller oil supply and valve position controller. The list of components is hence the following:

GCVh - Gas Control Valve (hydraulics)
SRVh - Stop Ratio Valve (hydraulics)
GCVm - Gas Control Valve (mechanics)
SRVm - Gas Control Valve (mechanics)

Notations: Variables are denoted by low case letter symbols whereas components are denoted by capital letter symbols.

For the GFS, the user's specifications state to consider faults on components: GCVm, GCVh, SRVm, SRVh, injectors and some transducers. The set of faults is hence given by F_{GFS}={GCVm, GCVh, SRVm, SRVh, Injt, T_{fsg}, T_{fsgr}, T_{fqg}, T_{cpd}}.

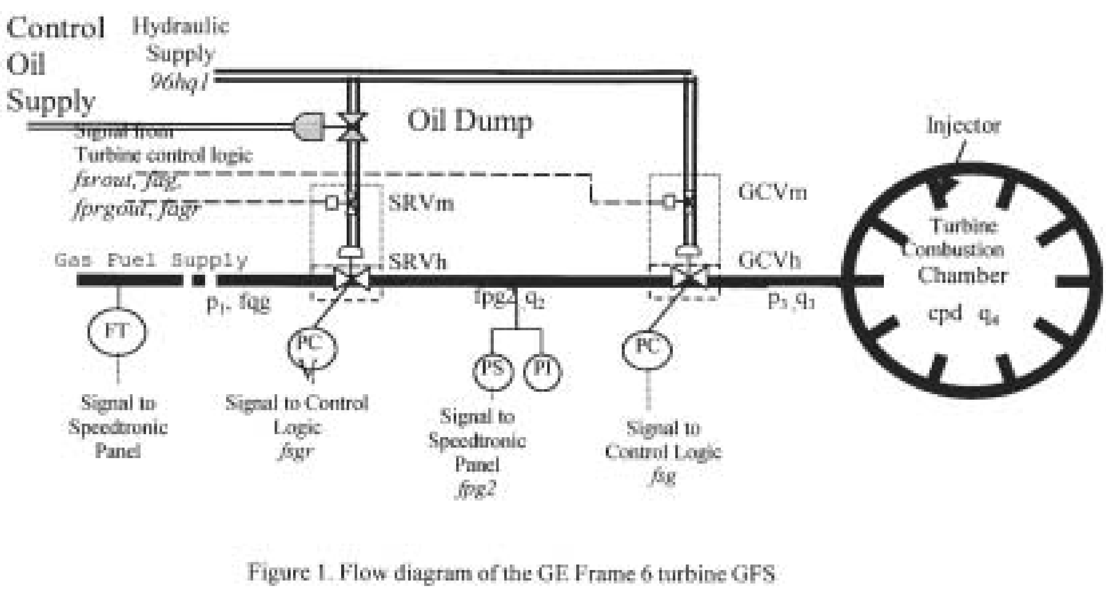

Figure 1. Flow diagram of the GE Frame 6 turbine GFS

2.2 GFS Component-Oriented Equation Model

The table 1 below provides the component-oriented model of the GFS. For every component, the behavioural relations refer to generic component models (Travé-Massuyès and Escobet, 1995). The transducers are not included. The descriptions of the variable names used in table 1 are given in table 2.

Component	Relation	Equation	Exogenous variables
Injectors	r1 r2	$q_3 = Kinj\sqrt{p_3 - cpd}$ $q_4 - Kli \times q_3 = 0$	cpd
GCVh GCVh	r3 r4	$q_2 = fsg\sqrt{fpg2 - p_2}$ $q_3 - Kl \times q_2 = 0$	
SRVh SRVh	r5 r6	$fqg = fsgr\sqrt{p_1 - fpg2}$ $q_2 - Kl' \times fqg = 0$	p1
GCVm	r7	$fsg = f(fag, 96hq1)$	96hq1
SRVm	r8	$fsgr = f(fagr, 96hq1)$	96hq1
GCVm	r9	$fsg = f(fsrout, 96hq1)$	fsrout, 96hq1
SRVm (SRVm +SRVh)	r10 r11	$fsgr = f(fprgout, fpg2, 96hq1)$ $fpg2 = f(fprgout)$	fprgout 96hq1

Table 1. GFS component-oriented model

cpd	Compressor discharge pressure
96hq1	Hydraulic pressure
fsrout	GCV position output
fprgout	SRV servo command
fsg	GCV position reference
fsgr	SRV position reference
fpg2	Intervalve gas fuel pressure input
fqg	Gas fuel flow

Table 2. List of variables

3 The Ca~En Diagnosis System

Ca~En is a causal model based diagnosis system that includes a fault detection module and a fault isolation module. The fault detection mechanism detects discrepancies between the system observed and predicted behaviour. It is based on the generation of adaptive thresholds from interval models to ground the decision problem. The fault isolation mechanism then interlinks the discrepancies to isolate the faulty components on the basis of a temporal causal model.

3.1 The Ca~En Knowledge Representation Formalism

The Ca~En formalism is based on a two-level representation scheme for the description of physical systems:

1. A causal model in which the links represent the causal influences existing among the variables, referred as the *local level*;
2. An analytical equation level which allows one to represent algebra-differential equations, referred as the *global level*.

Both levels can manage imprecise knowledge. A Ca~En program represents a formal model of the physical system built from knowledge about the physics underlying the behavior of the system.

Causal influences allow for representing causal dependency type knowledge. The Ca~En formalism offers five types of influences whose internal form is presented later:

- *dynamic 1st order*, denoted by the symbol —D1→ between the influencing and the influenced variables;
- *dynamic 2nd order*, denoted by the symbol —D2→;
- *integral*, denoted by the symbol —I→;
- *static*, denoted by the symbol —S→;
- *constant*, denoted by the symbol —C→.

Causal influences are characterized by several parameters, like a `gain`, a `delay` and a `response time` for 1st order dynamic influences. All parameters but the delays can be given an interval value when known with imprecision or a real value otherwise. They also allow for a parameter `condition`, which specifies the logical conditions under which the influence is active. This is the key for representing hybrid systems. Influences are labelled by the component(s) or process(es) which underlie them.

The following Ca~En example states that the variable `GFS_FQG2` is influenced by the variable `GFS_FPG2` through a static influence, which is active when condition `GFS_A1=1` is true. The underlying components are SRVh (Stop Ratio Valve hydraulics), Tfqg (*fqg* transducer) and GFsp (Gas Fuel supply system).

```
SRVh_Tfqg_GFsp:
    condition (GFS_A1=1)
    (GFS_FPG2 -S-> GFS_FQG2:
    gain in [-0,3630,-0,3411], delay=5;)
```

Following a component-oriented modeling approach, the Ca–En language allows the user to specify generic models (`model`), which can be invoked and instanciated on request. The definition of a model includes a list of *formal arguments*, then each instanciation is given a name, and a corresponding list of *actual arguments*. For example, a generic model `gen` can be defined as follows:

```
model gen (variable X in [0., +8], boolean
    parameter AA,constant G, constant R) (
    variable Y;
    i1: condition (AA) X-D1->Y: gain = G,
    resptime = R;
    init Y = 2; )
```

Model `gen` can then be invoked by:

```
    gg: instance gen (XX, true, 4, 2);
```

The global constraint level is composed of functional numeric constraints associated with interval domains, e.g. constraints arising from physical laws. In other words, a global constraint is any algebraic equation, which may be non-linear, in which each unknown is assumed to take on interval values. This allows us to manage imprecise knowledge at this level as well. The global constraints are expressed by means of traditional arithmetic operators: +, -, *, / and **. These operators are interpreted in the interval algebra.

As variables and parameters take interval values, one can easily adapt the model's granularity to the requirements of the faults. Hence Ca–En has a wide coverage of faults, from those radically changing the behavior of the physical system to those causing smooth deviations.

The internal structure of Ca–En presents two processing modules corresponding to the main tasks to be performed in fault diagnosis:

- A fault detection module based on a causal interval prediction mechanism;
- A fault isolation module based on an abstraction of the models in terms of temporal causal models.

The reader can refer to (Travé-Massuyès and Jimenez, 2001) for more details about the algorithms.

3.2 The Fault Detection Module

In model-based systems, the fault detection task can be accomplished through a prediction mechanism:

- the system model prediction (which may be a simulation in case of dynamic models) allows one to obtain the system expected behaviour,
- decision about the existence of a fault is based on comparing the expected and the observed behaviour and evaluating the so-called residuals.

In this section, we present how the two-levels representation of the physical system are used to simulate the system behavior and to obtain robust decisions about the existence of faults. The semi-closed loop fault detection algorithm used in Ca–En is illustrated with some examples which make clear the trade-off between sensibility and robustness.

3.2.1 The Ca–En Prediction Module

The prediction algorithm performs an estimation of the endogenous variable values across time. It can operate in an "open-loop mode", i.e. as a pure simulation, or in a "closed-loop mode", i.e. by taking into account in real time the measured variable values and performing a reset.

The temporal unit of the prediction module is the same as the data acquisition system. The input data are the causal model - including initial conditions - and the evolution of the exogenous and other measured variables over time. The output of the system is the trajectory of each process variable (Travé-Massuyès and Milne, 1997).

The prediction module can be used on its own or coupled with the fault detection module, in which case it is used in a *Semi-Closed Loop* (SCL) mode as explained in section 3.2.2.

Predicting the variable values is one of the most critical steps in the interval model-based fault detection approach. The predictions need to be fine enough to be sensitive to faults, but not too fine so as to avoid generating false alarms (Tornil *et al.*, 2001).

In Ca–En, two steps are executed to predict the variable values: at the local constraint level and at the global constraint level.

Influence	Representation Formalism	Transfer Function (Diff. or Algebraic equation)	Discrete Transfer Function (Ca–En internal form)	
Dynamic 1	x—D1—>y	$\frac{Y(s)}{X_i(s)} = \frac{Ke^{-sT_d}}{1+\tau s}$	$y(t+1) = a_{D1}\, y(t) + b_{D1}\, x_i\,(t-d)$	(1)
Dynamic 2	x—D2—>y	$\frac{Y(s)}{X_i(s)} = \frac{K\omega^2}{s^2+2\xi\omega s+\omega^2}$	$y(k+1) = a1_{D2}\, y(k) + a2_{D2}\, y(k-1) + b1_{D2}\, u(k) + b2_{D2}\, u(k-1)$	(2)
Integral	x—I—>y	$\frac{Y(s)}{X_i(s)} = \frac{K}{T_N s}$	$y(t+1) = a_I\; y(t) + b_I\; x_i\,(t-d)$	(3)
Static	x—S—>y	$\frac{Y(s)}{X_i(s)} = K$	$y(t+1) = K\, x_i\,(t+1-d)$	(4)
Constant	x—C—>y	$Y(s) = C$	$y(t+1) = C$	(5)

Table 3. Different types of Ca–En influences and their internal form

3.2.1.1 Local Constraint Level: Computation of the Updated Variable Values

From the superposition theorem that applies to the linear case, the computation of the updated value of a variable Y consists of processing the sum of the activated influences having exerted on the variable during the last time-interval.

Let's first consider the case in which y is influenced by one variable only, say x, through an influence of a given type. Depending on the type, table 3 provides the discrete internal form of the influence used by Ca–En as well as the continuous counterpart.

The symbols used in table 3 have the following meanings:

- T_s is the sampling period parameter;
- K and τ are the gain and the time constant of the 1[st] order transfer function. τ corresponds with good accuracy to $T_r/3$, where T_r is the response time (parameter `resptime`);
- ξ and ω are the damping ratio and the undamped natural frequency of the 2[nd] order transfer function;
- $a_{D1} = e^{-T_s/\tau} \approx e^{-3T_s/T_r}\; b_{D1} = K(1-a_{D1})$
- $a_I = 1\; b_I = KT_s$
- $a1_{D2}, a2_{D2}, b1_{D2}, b2_{D2}$
- are the 2[nd] order discrete transfer function parameters, whose expression depends on the type of function.

Let us now generalize to a variable y influenced (actively) by a set of *variables* $X=\{x_i\ i=1,\ldots,n\}$. The influences I_i $i=1,\ldots,n$, may be of different type. Let us define X_D, X_I, X_S and X_C as the subsets of variables of X influencing y through dynamic, integral, static and constant influences, respectively. Dynamic influences ($I_i \in I_D$), integral influences ($I_i \in I_I$), static ($I_i \in I_S$) and constant influences ($I_i \in I_C$) must be combined. Every influence is first materialised by an intermediate variable which stands for its associated *marginal influence* (this step is not necessary for static and constant influences). The sets of intermediate variables are V_D and V_I for dynamic and integral influences, respectively. The combination is then performed by adding up all the marginal influences by means of static influences, as illustrated in figure 2 where double arrows stand for sets of influences.

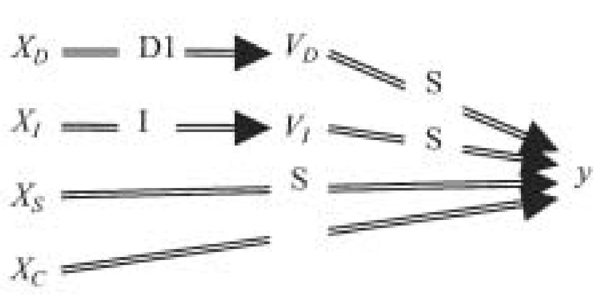

Figure 2. Ca–En influences combination

$v_i \in V_{li}$ is updated according to (1) or (2) in table 3, depending on the order of the transfer function. $v_j \in V_j$ is updated according to (3). Finally, at each sampling instant, y given by:

$$y(t) = \sum_{v_i \in V_{li}} v_i(t) + \sum_{v_j \in V_j} v_j(t) + \sum_{x_k \in X_k} K_k x_k(t - d_k) + \sum_{l / I_l \in I_c} C_l \qquad (6)$$

where K_k are the gains of the static influences and C_l the constants of the constant influences.

Note that the temporal features, captured by the delays and response times, are automatically taken into account. The result is an interval.

3.2.1.2 Global Constraint Level: Refinement of the Updated Variable Values

The numeric intervals obtained for the updated values (Equation (1)) are refined with the global constraints by performing a tolerance propagation algorithm (Hyvönen, 1992) on the set of variables. The effect of the tolerance propagation algorithm is to filter (reduce) for consistency the values $y(t)$ using the global constraints.

The simulation results produced by the Ca~En prediction module are envelopes (see figure 3) for the variables of interest. The envelopes provide the upper and lower bounds of the variable values at each sampled instant. As a consequence of the interval-based reasoning used in Ca~En, the results are complete but not correct (Armengol *et al.*, 2000).

Figure 3 below is a screen from the TIGER system that illutrates the enveloppes predicted by Ca~En, which provide adaptive thresholds for the measured signals.

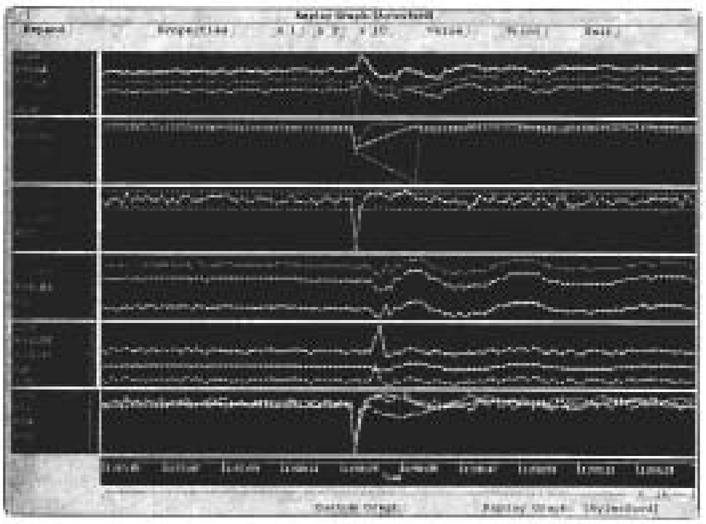

Figure 3. Ca~En enveloppes

3.2.2 Ca~En SCL Fault Detection Strategy

The Ca~En fault detection procedure is based on models of normal behaviour. These are interval models, which capture imprecise knowledge in the interval parameter values. The on line predictions obtained from these models is the basis of a discrepancy detection procedure based on *adaptive thresholds*, which allows us to track the physical system. This is performed by comparing the predicted and observed values of variables across time so that static as well as dynamic discrepancies are detected. This is essential for controlled systems such as turbines. The controller indeed tends to compensate for the faults in such a way that the fault might only be observable, and hence detectable, during the transient response of the turbine. The variables then generally stabilise at normal values. A classic limit checking diagnosis system is often inefficient in this kind of situation.

The ultimate goal of Ca~En being to isolate the fault(s), a decoupling is performed at the level of every measured variable. This means that variable measured values are always used to determine the prediction for their causally downstream variables. For example, for a static influence where $K=[K_{inf}, K_{sup}]$, we have:

$$\hat{y}_{pred}(t+1) = K x_{meas}(t-d) \qquad (7)$$

The suffixes "pred" and "meas" stand for "predicted" and "measured", respectively.

At each instant t and for every measured variable y, Ca~En checks whether the measured value $y_{meas}(t)$ (a real number) belongs or not to the predicted value $\hat{y}_{pred}(t)$ (an interval). If not, variable y is said to be *alarming* at time t. The set of alarming variables is denoted by $\mathcal{A}l$. This is equivalent to the calculation of an interval residual:

$$r_y(t) = y_{pred}(t) - y_{meas}(t) \qquad (8)$$

where $0 \not\subset r_y(t)$ in the faulty case and $0 \subset r_y(t)$ otherwise. Let's define as $\mathcal{A}l$ the set of variables such that $0 \not\subset r_y(t)$. From the graphical point of view, this is interpreted as the observed trajectory going out of the predicted curve envelope at time t as shown in figure 4.

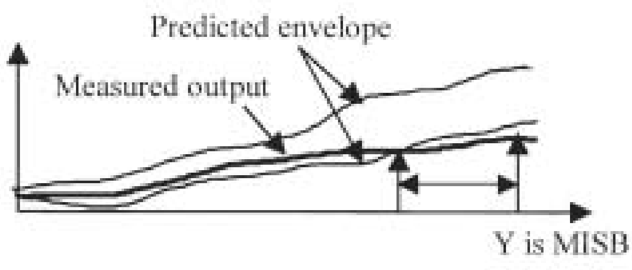

Figure 4. Fault detection is based on adaptive thresholds

In practice, noise in the measurements or other kind of disturbances must be contemplated in some way. In these situations, a local incompatibility between prediction and observation at some instant t does not necessarily mean that the system is faulty. In real applications, it is very important to have a robust fault detection system because a system which would untimely report faults would rapidly lose the confidence of the operator and engineering staff. Hence, we use a more robust indicator than just alarming variables. A fault is reported when a variable has remained alarming during a whole temporal interval T of length significantly greater than the sampling period (cf. figure 4). The variable is then said to be *misbehaving*. The length of T may be regarded as a multiple of the sampling period, i.e. $T=\nu T_s$ and it should be adjusted according to the technology of the sensors (the choice of ν is left to the user). ν is called the alarming-misbehaving threshold.

More formally, the set $\mathcal{MISB}$ of misbehaving variables can be defined as follows:

$$y \in \mathcal{MISB} \text{ at time } t \text{ if } y \in \mathcal{AL} \text{ since } t-\nu, \quad \text{i.e. } i=0,\dots,\nu,\ 0 \not\subset r_y(t-i) \qquad (9)$$

Within the above presented framework, the Ca~En fault detection strategy is a mixed strategy which combines an observer type strategy (closed-loop mode) with a simulation strategy (open-loop mode) to determine the residuals and further assesses variable state. We call this strategy a *Semi-Closed Loop(SCL) strategy* (Escobet at al., 2001).

The mode control (open-loop or closed-loop) depends on whether the observed value of y is in the predicted envelope (normal situation) or out of the predicted envelope (alarming situation) as illustrated below for a 1st order transfer function:

If $y \notin \mathcal{AL}$ then closed-loop mode, then

$$y_{pred}(t+1)=a_D y_{meas}(t)+b_D x_{meas}(t-d) \quad (10)$$

If $y \in \mathcal{AL}$ then open-loop mode, then

$$y_{pred}(t+1)=a_D y_{pred}(t)+b_D x_{meas}(t-d) \quad (11)$$

where $a_D \in [a_{D\min}, a_{D\max}]$ and $b_D \in [b_{D\min}, b_{D\max}]$

The two mentioned modes correspond to the schemas in figure 5a (closed-loop mode) and figure 5b (open-loop mode). The intuition behind this mixed strategy is related to two issues:

- The closed-loop mode runs on one step ahead predictions only, obtaining this way a good precision, which is critical when using interval models (Armengol *et al.*, 2000).
- As soon as the variable becomes alarming, running on a closed-loop mode would drive the prediction to follow the fault, turning the fault detection procedure insensitive to the fault.

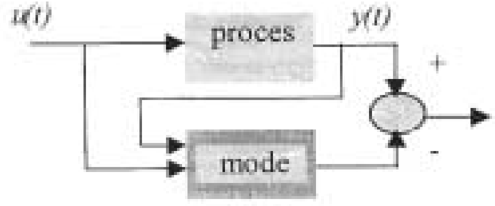

Figure 5a. Close loop mode

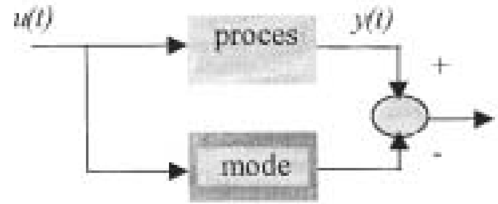

Figure 5b. Open loop mode

A comparison between open-loop and closed-loop modes is shown in figure 6. The plotted envelopes correspond to a dynamic influence with interval parameters K=[0.75,1.25] and τ=[1.75,2.25] (sampling period is 0.5). It can be seen that closed-loop mode (in continuous line) produces less conservative envelopes.

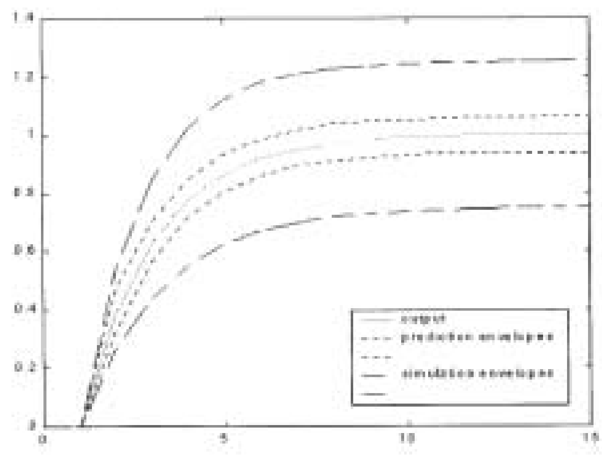

Figure 6. Closed-loop (continuous) vs. open-loop (dashed) predictions

The Ca~En SCL strategy is compared with the closed-loop strategy in the following figures. In figure 7.a, noise is added to the step response of a first order system with $K=1$ and $\tau=2$. The behaviour of the closed-loop and SCL strategies in this noisy situation are shown in figures 7.b and 7.c, respectively (with $K=[0.75, 1.25]$ and $\tau=[1.75, 2.25]$).

The upper subfigures of figures 7.b and 7.c show the output envelopes using interval prediction and the SCL strategy, respectively. The middle ones show the binary signal that indicates when the system output goes outside the envelopes (alarming signal). Finally, the lower ones show the misbehaving signals. It can be observed that, using the Ca~En SCL strategy, the system output can remain outside the envelopes up to 2 consecutive samples. This means that false alarms are avoided using an alarming-misbehaving threshold equal to 3. It can also be observed that a much higher threshold is needed to avoid false alarms when closed-loop strategy is used, leading to a higher detection time.

The SCL strategy used with $\nu=3$ produces better envelopes and decreases the number of false alarms.

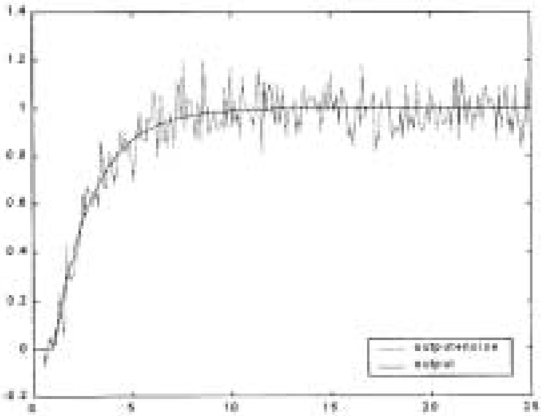

Figure 7.a. Noisy first order step response

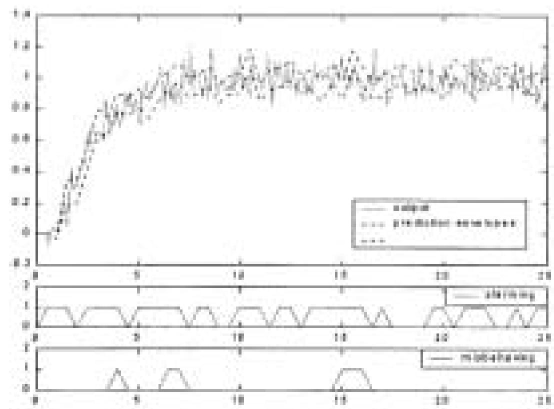

Figure 7.b Closed-loop detection strategy

When a fault is present, the SCL strategy is able to discriminate between noise and the effect of the fault. In figures 8a and 8b, the behaviours of the SCL strategy and simulation are compared when an abrupt fault occurs (at t=3 sec). The same alternatives are compared in figures 9a and 9b in the case of a "drift-type" fault. In both cases, it can be observed that the SCLleads to lower detection times.

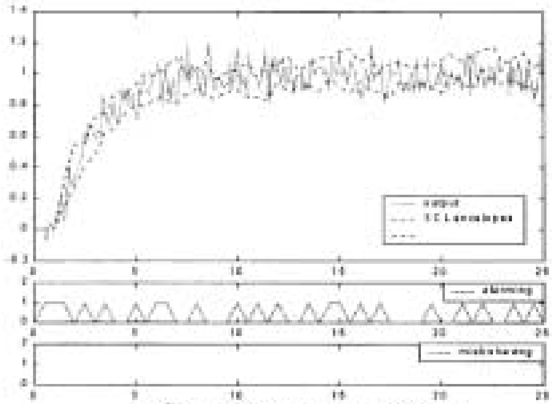

Figure 7.c. SCL detection strategy

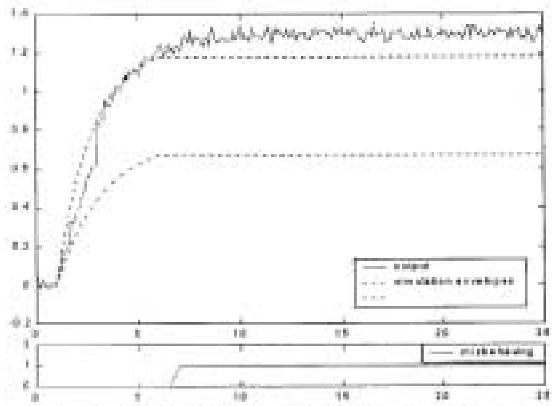

Figure 8.a. Simulation-based detection of an abrupt fault

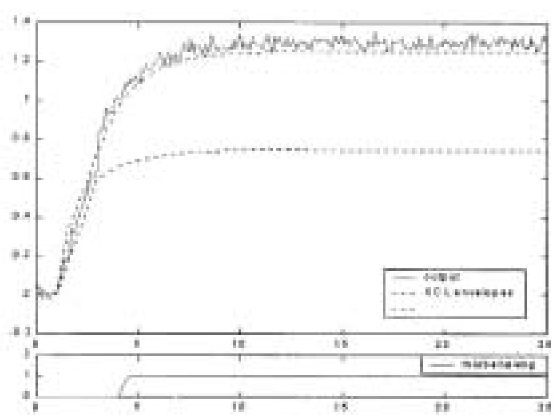

Figure 8.b. SCL-based detection of an abrupt fault

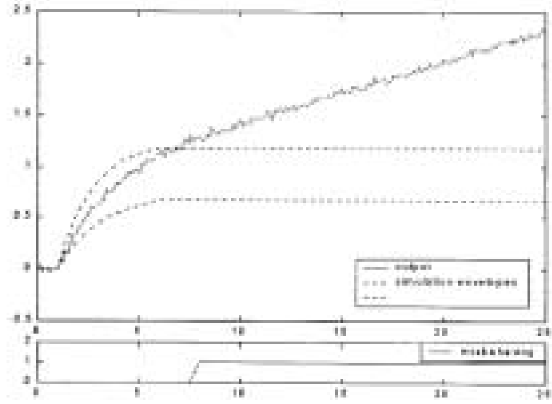

Figure 9.a. Simulation-based detection of a drift fault

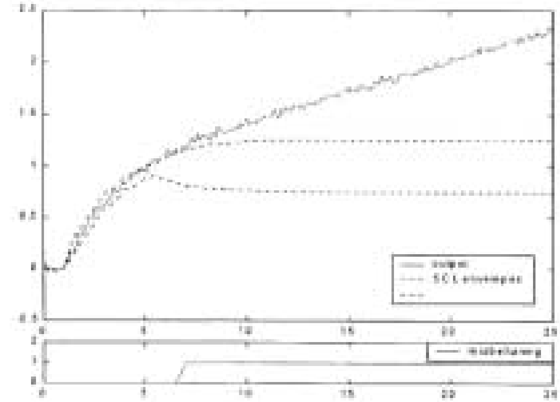

Figure 9.b. SCL-based detection of a drift fault

3.3 The Fault Isolation Module

Having detected one or more misbehaving variables, our system searches for the original possible cause(s) and elaborates a list of potential diagnoses. A diagnosis is a minimal set of components for which the invalidation of the normal behaviour assumption yields *(SD, COMP, OBS)* consistent, where *SD* is a formal description of the system including assumptions of normal behaviour for the set *COMP* of components and the components in *COMP* are the elementary diagnosis units. In the Ca~En diagnosis approach, the causal structure, including temporal aspects, acts as the *SD* and the influences themselves are the elements of *COMP*. Faulty influences are turned back into their corresponding faulty components. Temporal aspects include delay times, as well as the dynamics introduced by the different operating modes captured by the influences activation conditions and associated activation condition.

Diagnoses are computed from the collection of *conflict sets*, i.e. sets of components such that the observations indicate that at least one of the components in the set must be behaving abnormally, by an incremental hitting set algorithm (Levy, 1989). They are given as sets of faulty components labelled by their corresponding time of failure.

The diagnosis process is initiated as soon as a variable is reported as misbehaving. For this variable, say X, the conflict generation procedure traces backward in the causal graph, following the intuition that the influences which may be at the origin of the misbehaviour of X are those related to the edges belonging to the paths reaching node X from the first upstream measured variable nodes. This set of influences, called the *ascendant influences* of X, is recorded as a conflict set.

Proposition 3.1. The set of *ascendant influences* of a misbehaving variable X is a conflict set.

The conflict sets characterized by proposition 3.1, correspond to the discrepancies reported as misbehaving variables.

Let us consider the causal graph in figure 10 where all the variables are measured but U, and labels I_1, I_2, I_3, I_4, and I_5 denote the influences corresponding to the concerned edges. Assume that variable X misbehaves and that Y is correct. Then, according to proposition 3.1, $\{I_1, I_2, I_4\}$ is a conflict set.

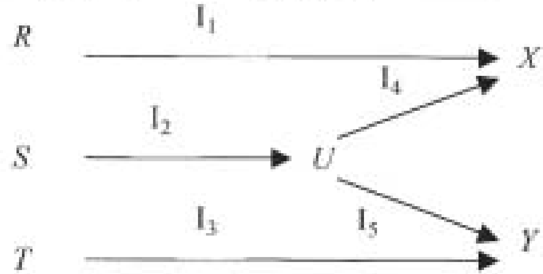

Figure 10. X is misbehaving, Y is not misbehaving, U is unmeasured

When generic fault models are considered, another type of conflict set can also be outlined by accounting for non-misbehaving variables. Knowing that some variable is not misbehaving may be informative for refining the diagnosis if some specific assumptions can be made. However, not all the non-misbehaving variables are useful; intuitively, only the variables which have at least one causally upstream unmeasured variable in common with a misbehaving variable must be considered. This is indeed the case of Y with respect to X in our example.

In our domain, and given the type of influences that are considered, the *single fault exoneration assumption* is of special interest. This is interpreted by the assertion that one and only one faulty influence always manifests by the misbehaviour of its downstream measured variables. This is also known as the single fault ARR-based[1] exoneration assumption (Cordier *et al.*, 2000).

Let us come back to our example and assume influences I_1 and I_4 to be functioning correctly, this implies that the faulty influence is I_2. Since U is not measured, this is detected by means of the misbehaviour of X. Meanwhile, Y is correct, which is not consistent with the exoneration assumption if influences I_3 and I_5 are supposed to be correct. Therefore, I_3 and I_5 must be faulty for compensating the propagation of the abnormality of I_2. Hence, if we assume I_1 and I_4 to be correct, then I_3 or I_5 cannot be assumed to be so. Therefore, $\{I_1, I_4, I_3, I_5\}$ is another conflict set. This reasoning can be generalised with the following proposition:

Proposition 3.2. If a non-misbehaving variable Y has at least one causally upstream unmeasured variable in common with a misbehaving variable X, then the symmetric difference[2] of the sets of the ascendant influences of Y and that of X is a conflict set.

In summary, It has been shown that two types of conflict sets can be outlined when a variable misbehaves. Both conflict sets are not based on the same concept. The first one is based on the fact that the misbehaviour of a variable is explained by at least one ascendant influence being faulty, independently of any assumption. The second conflict set is supported by the fact that a faulty influence induces the misbehaviour of downstream variables unless compensated by another faulty influence. Note that, it could be possible that the fault is hidden due to the unsoundness of interval computation, which can generate spurious behaviours (although the probability can be considered very low). This explains that the conflicts obtained by tracing back in the causal graph from misbehaving variables are called *hard conflicts* whereas the conflicts accounting for non-misbehaving variables are called *soft conflicts*.

As will be explained later, this distinction provides the basis for defining a preference criterion for the generated diagnoses.

The diagnosis generation is based on generating the minimal hitting sets of the collection of conflicts generated by the above algorithm. As new symptoms for a given fault can appear across time, it is important the diagnosis procedure be incremental. In Ca~En, we use Levy's algorithm (Levy, 1989) which is an incremental revised version of Reiter's original one (Reiter, 1987). The diagnoses are classified according to whether their elements belong solely to hard conflict sets or not. Let us define a *soft conflict element* as one that belongs to a soft conflict and does not belong to any hard conflict. Then, the *preference class* of a diagnosis set is defined as the number of soft conflict elements that it contains. The smaller the preference class of a diagnosis set, the more the concerned diagnosis is preferred.

Figure 11 below is a screen from the TIGER SHEBA system that illutrates the isolation procedure on the Frame 6 turbine.

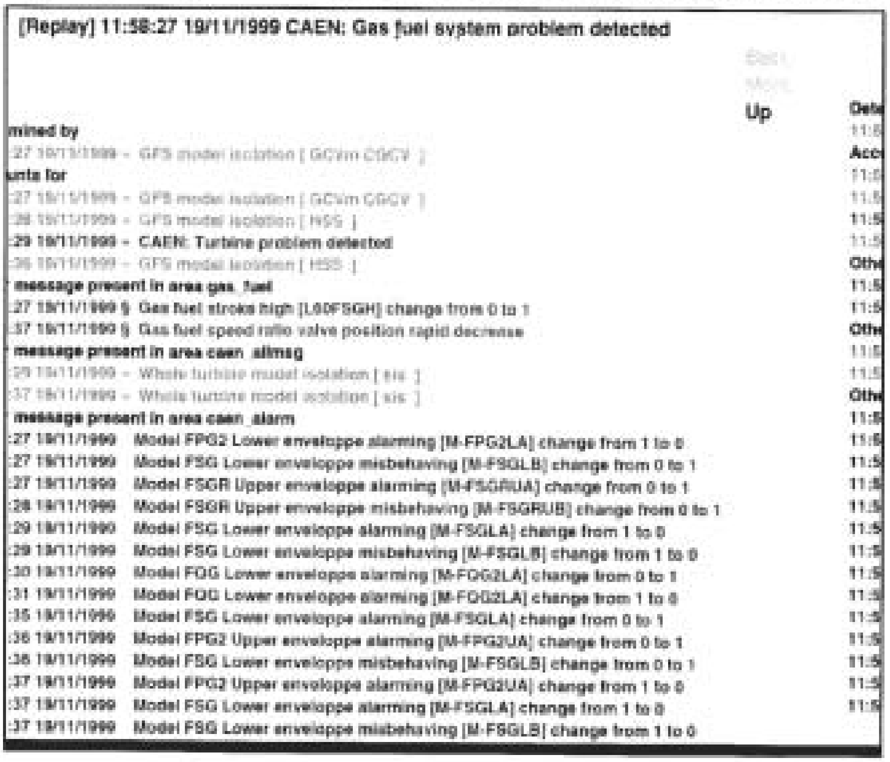

Figure 11. Faults reported by the TIGER SHEBA system

[1] ARR stands for *Analytical Redundant Relation*.

[2] The symmetric difference of two sets is their union without their intersection.

4 The Ca~En Modelling Methodology

The Ca–En system modelling methodology takes advantage of automated modelling methods. Let us recall that the extension of the causal ordering method (Iwasaki and Simon, 1994) to hybrid systems was proposed by (Travé-Massuyès and Pons, 1997) as a requirement for Ca–En modelling. In addition to this method, other modelling features have been added which are reported in this paper. Each step of the modelling method is presented: the automatic generation of a causal structure from a component-oriented equation model and how to get the parameters of the causal influences, then how to automatically derive the operational detection models.

4.1 Generation of the Local Level Causal Structure with Causalito

The problem of causal ordering has been approached by several authors generally for providing an explanation of why a device produces the behaviour it does. Among all existing approaches, we focused on the causal ordering of Iwasaki and Simon (1994). We agree with their main idea, that is to derive the causal ordering from a structural analysis of the equations. Our software *Causalito* implements an extended causal ordering algorithm (Travé-Massuyès and Pons, 1997).

The first requirements for Ca–En come from the fact that the prediction at the basis of the fault detection mechanism is performed along the causal structure. Ca–En hence needs a full causal structure in the sense that any endogenous variable must be reachable from the set of exogenous variables. So, we need to determine one possible interpretation around the loops. Notice that different interpretations require different propagation functions, leading in fine to the same predictions.

The second requirement is that we must provide a causal ordering for multiple mode systems (hybrid systems). By multiple mode systems, we mean systems in which there are some components like switches or valves that may be opened or closed, adding or retracting new branches to the circuit. The equational model of such systems have conditions associated to some of the equations. Instead of generating a new causal structure for every mode, Causalito performs an incremental generation of the causal structure. It first computes a causal structure in a given mode, switches to another mode and computes the minimal changes in the causal structure that represent the mode switching. Every possible configuration of the circuit is considered. Hence the influences of the final causal structure are labeled with an activation condition reflecting the (discrete) state of the multiple mode components.

Causalito takes as input the occurrence matrix of the system in which element m_{ij} is 1 or 0 if the variable X_i appears or not in the equation E_j. Causalito builds a bipartite graph G in which each variable X_i and equation E_j are represented by a node, and it exists an edge between X_i and E_j if m_{ij} is 1. Finding the corresponding causal structure starts with computing a perfect matching in the graph G. By orienting the edges of the perfect matching from the equation nodes to the variable nodes and the other edges from the variable nodes to the equation nodes, we obtain a first causal structure in which the equations (relations) appear explicitly. This structure is also known as the Resolution Process Graph (RPG)(Cassar and Staroswiecki, 1997).

This causal sructure states the flow of computations for variables. It is made of alternated levels of variables and relations. To solve a relation r_i for its matched variable, all the r_i's input variables have to be solved. Every relation is labelled with its corresponding component.

The final causal influence structure is obtained by aggregating the equation node and the variable node associated by the perfect matching. Detailled procedures are presented in (Travé-Massuyès and Pons, 1997).

4.2 Deriving the Operational Detection Models

Whereas the component-oriented model (primary relations) and its associated causal structure, that we will call the *generic causal structure*, as generated above (c.f. section 4.1) are suitable for fault isolation, the models that can be used for fault detection, i.e. the *operational detection models*, are generally highly dependent on the available sensors, which may vary from one system to another in the same class, i.e. gas turbines.

The methodological study presented in this section examines how to determine, for a given system, the operational detection model structure and the components to be associated to every operational relation. This is done from the generic (and unique) component-oriented model structure. This is then illustrated in section 4.2.1.2 on the Frame 6 turbine GFS.

4.2.1 Methodology to Derive the Operational Models

In Ca–En, the model prediction supporting fault detection requires to know the interval values of the parameters involved in the relations. In many applications, none of the parameters involved in the primary relations of the component-oriented models is known; i.e. their values are not even known to the user. Hence, the only relations that can be operationally used for fault detection are the ones whose parameters can be estimated from the data. This rests on the condition that the variables appearing in the relations are measured or can be considered as constant.

Parameter estimation procedures state that the relations are causal, i.e. the variables involved in the relations are either input or output variables and the output variables causally depend on the input variables. Ca–En detection models hence have an explicit representation of the underlying causal structure. This causal structure can be automatically derived from the generic causal structure by an aggregation operation. In some cases, it might be more suitable to use the aggregated causal structure for fault isolation as well. In this case, the component labels to be associated to every operational relation must also be retrieved. These two steps are detailed below.

4.2.1.1. Deriving the Causal Structure of the Operational Detection Models

Let us consider the generic causal structure obtained from the component-oriented model by applying the causal ordering procedure of (Travé-Massuyès and Pons, 1997). Then the operational detection model causal structure is obtained by the following aggregation process:

For all the non measured variables, do:

Step 1: Empty the non measured variable nodes.

Step 2: Aggregate the paths including an empty node into single causal influences.

Step 3 (optional): Label the new influences with their associated components.

The step 1 consists in removing in sequence the nodes corresponding to non measured variables from the generic causal structure. Once these nodes are discarded, the causal paths in which these nodes appeared must be restored in step 2. That is, for each discarded variable Y, we have to replace any causal path $X \rightarrow Y \rightarrow Z$ by the causal influence $X \rightarrow Z$. In step 3, the component labels associated to the "new" influences must be determined from the old ones. The old influences that are common to several paths distribute their associated component label to all the new influences replacing these paths, as shown in figures 11a and 11b.

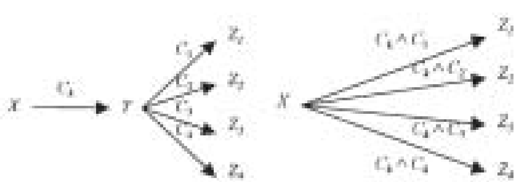

Figure 11a. Causal structure aggregation operation 1

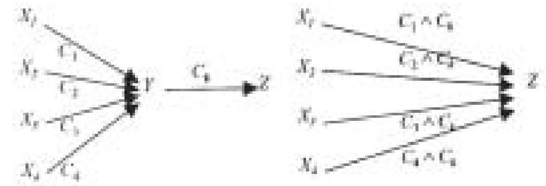

Figure 11b. Causal structure aggregation operation 2

Note that the component labels to be associated to operational relations can also be obtained from the generic causal structure by tracing back from the relation's output variable(s) to the input variable(s) and recording the components associated to all the relations that are in between.

4.2.1.2. Application to the Frame 6 turbine GFS

In the GFS, there are 5 exogenous variables {CPD, FSROUT, FPRGOUT, P1, 96HQ1} and 4 measured endogenous variables {FSG, FPG2, FQG, FSGR}. The GFS generic causal structure is presented in figure 12 (Travé-Massuyès *et al.*, 2001).

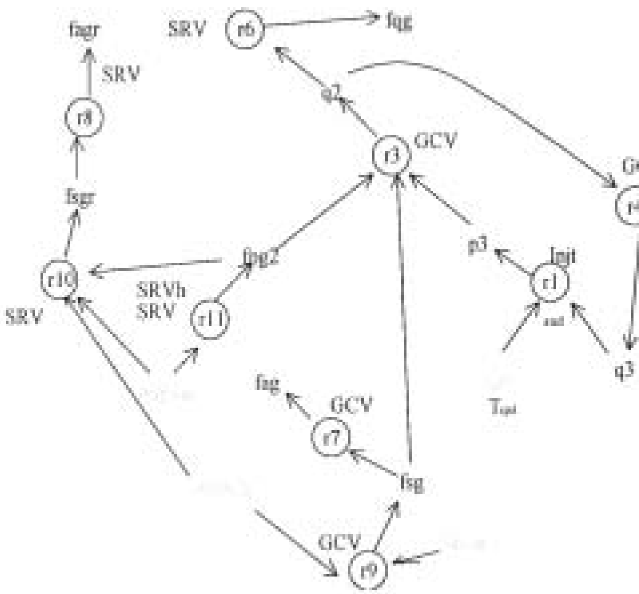

Figure 12. GFS causal structure

The exogenous variables 96HQ1 and P1 are measured or not, depending on the turbine. The isolation and detection operational causal structure given in figure 13 shows four operational relations:

(GFS_op1) FSG=f(FSROUT, 96HQ1)
(GFS_op2) FPG2=f(FPRGOUT)
(GFS_op3) FSGR=f(FPG2, FPRGOUT, 96HQ1)
(GFS_op4) FQG1=f(CPD, FSG, FPG2)

(GFS_op1), (GFS_op2) and (GFS_op3) are primary relations. (GFS_op4) is a combined relation obtained by combining primary relations. This is valid upon the assumption that all the primary relations are invertible or that they can be linearized around some operating point. In this later case, the validity domain of the detection operational model is limited to the neighbourhood of the operating point.

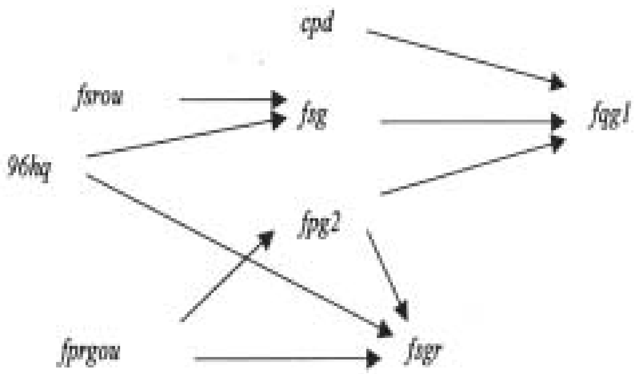

Figure 13. GFS operational model

To determine the component label for GFS_op4, the search from FQG back to CPD, FSG and FPG2 is obtained applying the aggregation operations on the AND-OR Graph. The same procedure applied to (GFS_op1), (GFS_op2), and (GFS_op3) leads us to conclude on the associations given in table 4.

Operational relations	Associated operational influences	Associated components
GFS_op1	Influences on FSG	GCVm $\mid$ T_{FSG}
GFS_op2	Influences on FPG2	SRVh $\mid$ SRVm
GFS_op3	Influences on FSGR	SRVm $\mid$ T_{FSGR}
GFS_op4	Influences on FGR	SRVh $\mid$ GCVh $\mid$ Injt $\mid$ T_{CPD} $\mid$ T_{FSG} $\mid$ T_{FQG}

Table 4. Operational relations and associated influences and components

5 Application Results

The presented methodology has been applied in the gas turbines application domain in the TIGER and TIGER-SHEBA european projects (Travé-Massuyès and Milne, 1997)(Milne et al., 2001). The TIGER system is commercialized by IA Ltd all around the world. In the framework of the TIGER project (1992-1996), Ca~En was successfully tested on several subsystems of a 28-MW General Electric Frame 5 gas turbine operating at the Exxon Chemical Fife Ethylene Plant (UK) and of a Dassault Aviation Auxiliary Power Unit manufactured by MicroTurbo (F). It was then fully integrated to TIGER within the TIGER SHEBA project (1998-2000) and runs on-line on the Frame 6 gas turbine of the National Power's cogeneration plant at Aylesford (UK).

Figures 14 and 15 in next page illustrate the fault detection and fault isolation results provided by Ca~En and reported by the TIGER SHEBA system.

6 Conclusions

Model based technologies fully rely on the quality of the models and hence call for automated modelling methods. These methods should allow the users:

- to compose the model of a complex system from model fragments on one hand
- to reuse and automatically transform existing models;
- to derive the model features from existing data.

This paper presents the diagnosis system Ca~En and its automated modelling method, which offers several interesting features. After the presentation of the main theoretical principles involved in Ca~En, we show how a generic causal structure can be automatically generated from a component-oriented equation model and how we can derive the operational causal model used by Ca~En. This approach has been applied to many systems and the Frame 6 turbine Gas Fuel System is presented as an example. The results obtained in the gas turbine application domain are discussed.

In relation with this work, the important issue of diagnosability and sensor placement has also been approached. The method proposed in (Travé-Massuyès *et al.*, 2001) is based on a causal model similar to the one used by Ca~En.

On-going work is now considering the diagnosis of hybrid systems. Current developments integrate the Ca~En approach for continuous systems to a discrete automata reasoning level to provide diagnosis conclusions and perform system tracking. This is applied to the autonomous satellites domain (Benazera *et al.*, 2001).

Acknowledgements

We thank each of the partners of the recent TIGER SHEBA european project, National Power Cogeneration and Kaverner Energy Ltd., with special thanks to R. Milne and C. Nicol from Intelligent Applications Ltd for their help and comments referring to the application work.

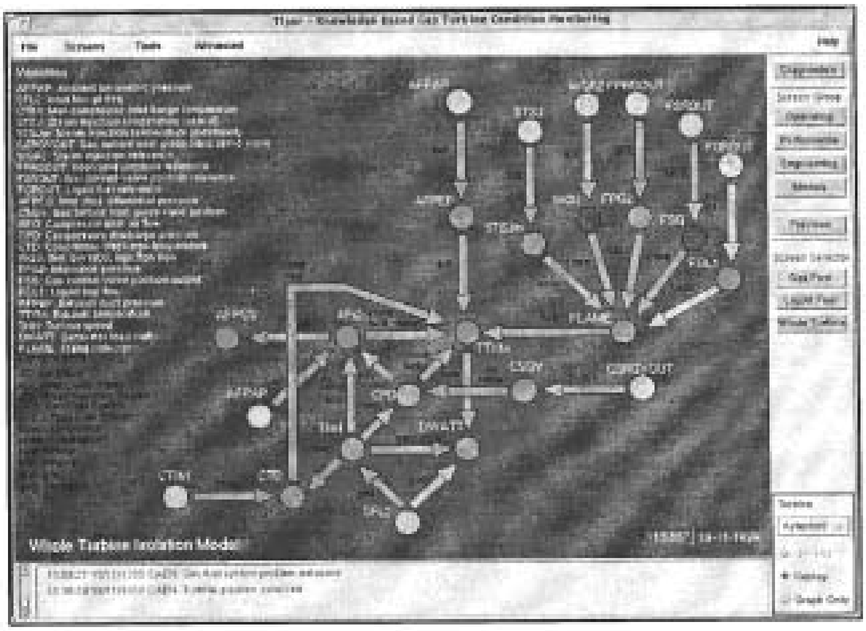

Figure 14. Turbine incidents reported in red by TIGER -SHEBA

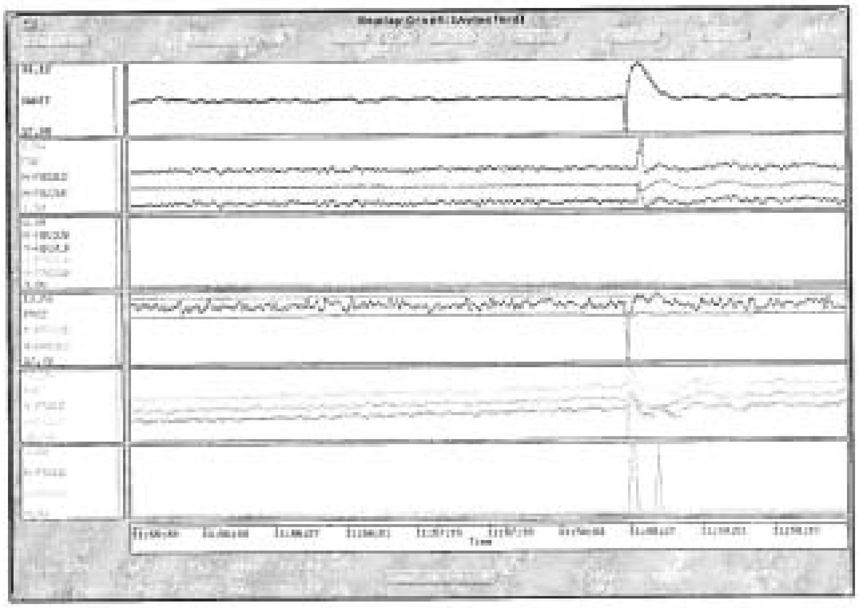

Figure 15. Fault detection envelopes reported by TIGER-SHEBA

References

Armengol J., L. Travé-Massuyès, J. Vehí, and J. Lluís de la Rosa, A survey on interval model simulators and their properties related to fault detection. *Annual Reviews in Control 24*, pp. 31-39, 2000.

Benazera E., Travé-Massuyès L., and Dague P., Hybrid model based diagnosis for autonomous spacecrafts, ESA Workshop on « On-board Autonomy », Noordwick (The Netherlands), pp. 279-286, 17-19 October 2001.

Bibas S., Cordier M-O., Dague P., Dousson C., Lévy F., and Rozé L., Alarm Driven Supervision For Telecommunication Networks: I- Off-Line Scenario Generation, *Annales des Télécommunications, vol. 9/10*, pp. pp.493-500, Oct. 1996.

Cassar J.P. and **Staroswiecki M.** A structural approach for the design of failure detection and identification systems. In *Proc. IFAC, IFIP, IMACS Conference on 'Control of Industrial Systems'*, Belfort, France, pp. 329-334, 1997.

Cauvin S., M-O. Cordier, C. Dousson, P. Laborie, F. Levy, J. Montmain, M. Porcheron, I. Servet, and **L. Travé-Massuyès**, Monitoring and alarm interpretation in industrial environments. *AI Communications 11(3, 4)*, 1998.

Cordier M.-0., Dague P., Dumas M., Levy F., Montmain J., Staroswiecki M., and **Travé-Massuyès L.**, A comparative analysis of AI and control theory approches to model-based diagnosis. *ECAI '00*, Berlin, Germany, August pp.20-25, 2000.

Dousson. C., P. Gaborit, and **M. Ghallab**, Situation recognition: representation and algorithms, *Proc. 13th IJCAI Conference*, 166-172, Chambéry, France, 1993.

Escobet T., L. Travé-Massuyès, S. Tornil, and **J. Quevedo**, Fault detection of a gas turbine fuel actuator based on qualitative causal models, *European Control Conference (ECC'01)*, Porto (Portugal), 4-7 Septembre 2001, pp.2741-2746.

Hyvönen E., Constraint reasoning based on interval arithmetic: the tolerance propagation approach, *Artificial Intelligence 58*, pp 71-112, 1992.

Iwasaki Y., and **H. A. Simon**. Causality and model abstraction, *Artificial Intelligence 67*, pp. 143-194, 1994.

Lévy F., *Contribution à la réalisation d'un raisonneur à profondeur variable*, Université Paris Nord PhD dissertation thesis, 1989.

Milne R., C. Nilcol, and **L. Travé-Massuyès**, TIGER with model based diagnosis: initial deployment, *Knowledge-Based Systems 14(2001)*, pp. 213-222.

MONET-ILC98, *Compendium of Industrial Applications of Model Based Systems and Qualitative Reasoning*, prepared by the MONET Industrial Liaison Committee, 1998.

Pons R. and **L. Travé-Massuyès**, Causal Ordering for Multiple Mode Systems. *11th International Workshop on Qualitative Reasoning (QR'97)*, Cortona, Italy, 1997.

Struss P., M. Sachenbacher, and **C. Carlen**, Insights from building a prototype for model-based on-board diagnosis of automotive systems, *In Working Papers of the 11th International Workshop on Qualitative Reasoning (DX'00)*, Morelia, Mexico, 2000.

Tornil S., T. Escobet, L. Travé-Massuyès, and **E. Benazera**, Robust Interval Model Based Fault Detection – Satellite Attitude and Orbit Control System Case Study, *15th IFAC World Congress on Automatic Control*, Barcelona, Spain, 21-26 July 2002 (submitted).

Travé-Massuyès L., T. Escobet, and **R. Milne**. Model-based diagnosability and sensor placement. Application to a frame 6 Gas turbine sub-system, *7th International Joint Conference on Artificial Intelligence (IJCAI'01)*, Seattle (USA), 4-10 August 2001, 551-556.

Travé-Massuyès L. and **T. Escobet**, *Tiger Deliverable D523.36, APU Fuel System Model*, Esprit Project 6862: Tiger, Intelligent Applications Ltd., 1995.

Travé-Massuyès L. and **Jimenez J.A.** Fault detection and isolation in the Ca~En system. *LAAS-CNRS technical report*, Toulouse, 2001.

Travé-Massuyès L. and **R. Milne**. Gas Turbine Condition Monitoring Using Qualitative Model Based Diagnosis, *IEEE Expert Intelligent Systems & Applications*, pp. 22-31, May-June 1997.

Travé-Massuyès L. and **R. Milne**. Gaps between research and industry related to model based and qualitative reasoning, *In the proceedings of the ECAI'98 workshop on Model Based Systems and Qualitative Reasoning (QR'98)*, Brighton(UK),August1998.

Louise Travé-Massuyès *was born in July 1959, in Manresa, Spain. She received an Engineering Degree specialized in control, electronics and computer science in 1982 and a Ph.D. degree in control in 1984, both from the Institut National des Sciences Appliquées (INSA); Award from the Union des Groupements d'Ingenieurs de la Region Midi-Pyrénées; D.E.A. in control from Paul Sabatier University in 1982, all in Toulouse, France. She is currently a Research Director of the Centre National de Recherche Scientifique (CNRS), working at LAAS, Toulouse, France, in which she has led the "Qualitative Diagnosis, Supervision and Control" Group for several years. Her main research interests are in qualitative and model-based reasoning and applications to dynamic systems supervision and diagnosis. Her current responsibilities include: Co-director of the European Laboratory LEA-SICA; Chairperson of the IEEE SMC Technical Committee on Qualitative Reasoning; member of the IFAC Safeprocess Technical Committee. She is a Senior Member of the IEEE Computer Society.*

Teresa Escobet-Canal *was born December 1961, in Berga, Spain. She received the Industrial Engineering Degree and Ph.D from the Universitat Politècnica de Catalunya (UPC) in 1989 and 1997, respectively. She is currently a full-time associate professor in the Automatic Control Department (ESAII) at the UPC, in which she has been leading the "Advanced Control Systems" Group since October 2000. Her main research interests are in dynamic system modelling and identification applied to fault detection and isolation.*

Renaud Pons *was born in Belfort, France, in 1971. He received a D.E.A. in control (1995) and a Ph.D. degree in control and computer science (2000), both from Paul Sabatier University in Toulouse, France. He has currently a post doctoral position in the Laboratoire d'Analyse et d'Architecture des Systèmes of Centre National de la Recherche Scientifique (LAAS-CNRS) in Toulouse. His main research interests are in qualitative and model-based reasoning and applications to supervision and diagnosis.*

Sebastián Tornil *was born in Barbastro, Spain, in 1972. He received the M.D. degree in Computer Science from the Technical University of Catalonia (UPC) in 1996. He is currently a full-time assistant professor in the Automatic Control Department (ESAII) at UPC. His main research interests are modelling, simulation and fault diagnosis of uncertain dynamic systems.*