



HAL
open science

Fuzzy reasoning in co-operative supervision systems

Alexandre Evsukoff, Sylviane Gentil, Jacky Montmain

► **To cite this version:**

Alexandre Evsukoff, Sylviane Gentil, Jacky Montmain. Fuzzy reasoning in co-operative supervision systems. *Control Engineering Practice*, 2000, 8 (4), pp.389 - 407. 10.1016/S0967-0661(99)00170-7. hal-01931750

HAL Id: hal-01931750

<https://hal.science/hal-01931750v1>

Submitted on 27 May 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fuzzy reasoning in co-operative supervision systems

Alexandre Evsukoff^{a,1}, Sylviane Gentil^{a,*}, Jacky Montmain^b

^aCNRS-INPG-UJF, Laboratoire d'Automatique de Grenoble, BP 46, 38402 Saint Martin d'Hères, France

^bCEA-CEN VALRHO, DCC/DRRV/SSP, Laboratoire d'Informatique Appliquée, BP 171, 30207 Bagnols-sur-Cèze Cedex, France

Abstract

This paper considers a decision support system dedicated to fault detection and isolation from a human-machine co-operation point of view. Detection and isolation are based on different models of the process (non-linear and linear causal local models). Reasoning using real numbers is often used by human beings; fuzzy logic is introduced as a numerical-symbolic interface between the quantitative fault indicators and the symbolic diagnostic reasoning on them; it also provides an effective decision-making tool in imprecise or uncertain environments while managing model uncertainty, sensor imprecision and vague normal behavior limits. Fuzzy rules are modelled geometrically; fuzzy sets are represented as points in a description space. A prototype graphical interface with structural, causal and historical views gives complete information to the human operator. In such an interface, fuzziness is displayed as a colour palette evolving with time.

Keywords: Supervision system; Fault detection and isolation; Knowledge-based systems; Fuzzy logic; Computer aided training

1. Introduction

Demands for increased productivity and flexibility has led to the operation of processes in fast changing conditions. The safety and efficiency of industrial processes are thus closely interrelated (Al Khani & Koivo, 1997). Currently, skilled human operators guarantee process availability. They must interpret complex situations through vast amounts of data, which change rapidly when a malfunction occurs and propagates throughout cascaded sub-systems.

Computer-supported co-operative work in supervision systems can improve plant reliability by combining computer and human resources in the main supervision tasks, i.e. monitoring and diagnosis. Nevertheless, the display of appropriate information in multi-human/machine interfaces is a challenge in computer supported co-operative systems (Johannsen, 1997).

It is evident that operators do not use mathematical models or precise values to make decisions but rather their reasoning is based on a symbolic interpretation of the process trend. The idea behind this remark is that like in many design or diagnostic activities in engineering, qualitative reasoning and/or order-of-magnitude reasoning are often not only sufficient in solving practical cases but also more realistic. Hence, the symbolic interpretation of data is the means of explaining the process behaviour to humans. Fuzzy reasoning techniques are a means for human-friendly computerised devices (Dubois, Prade & Yager, 1996; Iserman, 1998) because they allow the symbolic generalisation of numerical data by fuzzy sets, thus providing linguistic interpretability (Zadeh, 1996).

This paper presents a prototype system that executes the following supervisory functions as defined by Iserman and Ballé (1997), where explanation facilities have been added to original definitions to fit co-operative work:

- (a) *Monitoring*: “A continuous real-time task of determining the conditions of a physical system, by recording information, recognising and indicating anomalies in the behaviour” and generating alarms to human operators.
- (b) *Diagnosis*: “Determination of the kind, size, location and time of detection of a fault” and generation of

*Corresponding author. Tel.: + 33-476-826239; fax: + 33-476-826388.

E-mail addresses: gentil@lag.ensieg.inpg.fr (S. Gentil), jacky.montmain@site.eerie.ema.fr (J. Montmain).

¹ Present address: PEN/COPPE-UFRJ, Laboratório de Monitoração de Processos, PO Box 68509, 21945-970 – Rio de Janeiro, RJ, Brazil.

sound explanation to human operators. "Follows fault detection and includes fault isolation and identification".

Classical supervision systems present a lot of data that may evolve quickly. To focus the operator's attention, alarms are used. An alarm is generally initiated when a variable is passing over a threshold. These thresholds are difficult to choose by process engineers because they may correspond to different notions: strict safety limits, production tolerance, etc. They are vague concepts. Moreover, a small variation in a variable can make it pass over the limit and this abrupt change does not correspond to reality. Generally, the process evolves gradually into a bad situation and this progression should be expressed by the supervision system. Fuzzy sets, here again, are an appropriate tool to model gradual symbols.

Control engineering science has contributed highly to supervision thanks to analytical model-based redundancy that is probably the most active area in diagnosis (Frank, 1996). The comparison of numerical simulation of process variables with plant measurements allows numerous residuals to be generated, which makes it possible to detect and isolate faults. The difficulty is to get a complex process model. In some situations, a numerical simulator of the process is available. This is the case for the example presented at the end of this paper. This model is actually the one that is used for operator training. In other industries, a precise numerical simulator of the plant, reflecting all physical or chemical processes is not available because it would take a very long time to be developed. Nevertheless, approximate models can be developed specifically for supervision objectives, because these models may not be as precise as for control objectives (Leyval, Gentil & Feray-Beaumont, 1994). Noise corrupts measurements in every plant. Thus the comparison of system and model outputs is very imprecise. Fuzzy reasoning allows much better management of this kind of imprecision than crisp alarm thresholds.

Analytical models cannot generate a reasonable explanation of the diagnosis process. Rather, causal models can emulate expert operator's mental models in their diagnosis reasoning. A causal model is a qualitative description of the direct influences that some variables of a process have on others: one variable is the cause and the other the consequence. Causal models differ from analytical models (Rasmussen, 1993), in the notion of cause and effect sequences. Hence, causality is inherently related to time evolution. Causal model-based reasoning has been successfully used for fault detection and isolation in complex systems (Kramer & Palowitch, 1987) and recently it has been proposed to manage time dependencies explicitly with a causal graph (Leyval, Montmain & Gentil, 1994; Evsukoff, Montmain & Gentil, 1997).

In this work the causal structure of the process variables has been elicited from experts and represented as a causal graph. The nodes of the graph represent measurable variables. A simple linear dynamic model has then been linked to each edge of the graph by a classical identification procedure between the variables related to each edge. The resulting bank of linear models was used as additional information for isolation reasoning.

The proposed system is thus based on a multi-model architecture and is composed of two main modules: the *fault detection module*, based upon the analytical non-linear simulator of the process and the *fault isolation module*, based upon the causal model. The causal graph provides a conceptual tool to appropriately explain to human operators how changes propagate throughout the process.

In the proposed co-operative supervision system, fuzzy reasoning plays a central role. On the one hand, fuzzy sets are used to take into account modelling imprecision and measurement noise that corrupt the decision process. On the other hand, symbolic interpretation of reasoning conclusions is used to represent the evolution of the process as a gradual palette of colours in the graphical interface of the system. As a result, crisp thresholds are avoided, leading to stable decisions and allowing gradual information processing that can be easily understood by the human operators.

A prototype system has been implemented for a specific application in a nuclear fuel reprocessing plant and was evaluated by students training to be operators (Evsukoff, Montmain & Gentil, 1998). The paper is organised as follows: the second section presents an overview of computer supported co-operative systems for complex processes. Section 2 also introduces the basic concepts of the proposed supervision system, which are detailed in the subsequent sections. Section 3 introduces the key issues of fuzzy reasoning and its geometrical interpretation. Each of system modules is then explained. Section 4 details the fuzzification procedure. The detection module is presented in Section 5, while Section 6 is devoted to the isolation module. Section 7 illustrates the graphical interface of the prototype system with an example and comments its evaluation. Finally, some concluding remarks are made.

2. Co-operative supervisory systems

Computer-supported management and control systems for large-scale complex processes are characterised by a hierarchy containing different co-ordination levels shown in Fig. 1.

Low-level tools are close to the physical process and contain local units that execute closed-loop control of process sub-systems. Middle-level tools provide supervisory control to ensure set point changes and alarm

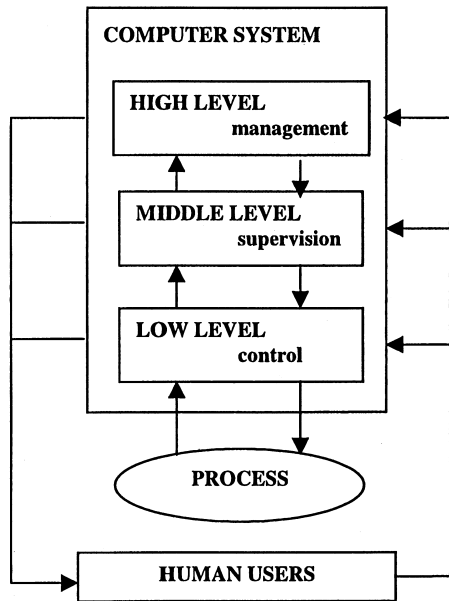


Fig. 1. Process management and control hierarchy.

handling, but for increasingly complex processes, this level must include more advanced tools in order to ensure global process availability and production quality. High-level tools include plant management and strategic decisions (Al Khani & Koivo, 1997).

In computer-supported co-operative work, all levels interact with human users who co-ordinate computer subsystems by means of multi-human/machine interfaces that must provide appropriate information (Johannsen, 1997).

The proposed co-operative supervisory system is shown in Fig. 2. In the following, each one of the modules is described as one of the five primitive canonical tasks proposed by Leitch and Gallanti (1992) for knowledge-based automation systems:

- (1) *Prediction*: generates future states from the present observed state by using an implicit or explicit model of the process,

- (2) *Interpretation*: maps sensor information in order to process state description defined by the adopted representation language,
- (3) *Decision*: generates conclusions from known process states,
- (4) *Identification*: generates unknown or unmeasurable past or present states from known or assumed current observed states,
- (5) *Execution*: is dual to interpretation, transforming symbolic conclusions into data suitable to intervention on the process.

In the system data flow presented in Fig. 2, the *prediction* task is performed by the two different models presented in the introduction (Evsukoff et al., 1997): the *analytical model*, based on physical relations; the *causal model*, representing linear approximations between internal variables. The use of the causal model is explained in greater detail in Section 6.

The *interpretation* task is performed by the *fuzzification* of some features extracted from residuals obtained from the difference between observable data and model outputs. The features extraction and the fuzzification step are presented in Section 4.

Symbolic descriptions of each process variable feed the *decision* and *identification* task modules, represented by the *detection module*, which decides whether the process state is normal or not, and the *isolation module*, which identifies the source fault. These decision modules are presented in Sections 5 and 6.

There is no *execution* task since human operators decide which recovery action is necessary and execute such action. This module is replaced in the system by an *explanation* facility for human operators, intended to avoid erroneous counteractions. Such an explanation is provided by a graphical interface.

The interface module presents symbolic decisions on each variable by colour codes and displays an overall representation of fault evolution in time. The fault propagation through the process is explained by the causal graph where the fault path is displayed. Additionally, the

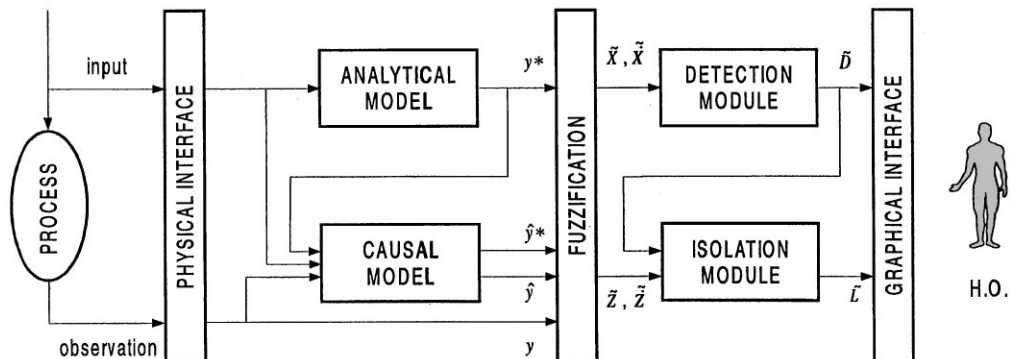


Fig. 2. The proposed system.

models are used for prognosis. Further information on the system interface is given in Section 7.

3. Fuzzy reasoning

This section details how fuzzy reasoning is used as the basis for interpretation of numerical data and for decision-making. A simple example is used throughout this section to clarify the notations.

3.1. Preliminaries

The symbolic-numerical interface in fuzzy systems is carried out by linguistic variables that take linguistic values associated with fuzzy sets. Formally, a variable v is defined by

$$v = (X, \mathbf{A}(X), \tilde{\mathbf{A}}, \Gamma). \quad (1)$$

where $X \subset \mathbb{R}$ is the *reference domain* of the variable, $\mathbf{A}(X)$ is its *descriptor set* and $\tilde{\mathbf{A}}$ is the *language* that assigns a meaning defined by a fuzzy set on X to each symbolic term in $\mathbf{A}(X)$. The descriptor set is a subset of the *term set* T , which contains all symbolic terms that can be generated by the *grammar* Γ (Dubois & Prade, 1980).

The symbolic terms in $\mathbf{A}(X)$ represent ordered linguistic concepts that describe the values in the reference domain. Each symbolic term represents a generalisation of the reference values whose meaning is a fuzzy set. For instance, the ordered descriptor set {low, normal, high} can describe a temperature.

The meaning of symbolic terms is provided by the language that associates the fuzzy set \tilde{A}_i with the term $A_i \in \mathbf{A}(X)$, for every element $x \in X$. Conversely, every element $x_0 \in X$ can be described by the fuzzy set \tilde{X}_0 defined from the descriptor set $\mathbf{A}(X)$. The language thus defines the relationship between the descriptor set $\mathbf{A}(X)$ and the reference X as the fuzzy relation $\tilde{\mathbf{A}}$ characterised by the membership function $A: X \times \mathbf{A}(X) \rightarrow [0, 1]$.

For every element $x_0 \in X$ of the reference domain, the meaning \tilde{A}_i is related to the description \tilde{X}_0 by the language, in such a way that

$$A_i(x_0) = X_0(A_i) = A(x_0, A_i). \quad (2)$$

The description \tilde{X}_0 is often called *symbolic fuzzification* (Foulloy & Galichet, 1995). For instance, Fig. 3 shows the description \tilde{X}_0 given by the descriptor set $\mathbf{A}(X) = \{A_0, A_1, A_2\}$.

The description of a numeric value must be interpreted as how such a value can be expressed by the vocabulary in the descriptor set. For example, if fuzzy sets in Fig. 3(a) denote the sense of “low”, “normal” and “high” temperatures, the fuzzy description in Fig. 3(b) is used to denote a given temperature measure $t_0 = 25^\circ\text{C}$.

The collection of fuzzy sets \tilde{A}_i associated with each term $A_i \in \mathbf{A}(X)$ forms a *fuzzy partition* whose cardinality denotes the *granularity* of the description provided by the descriptor set. The greater the cardinality of the fuzzy partition, the smaller the granularity of the description which allows more precise information to be represented but with less linguistic generalisation.

In this work, normalised orthogonal fuzzy sets are used in the meanings of variables, so that:

$$(i) \forall i, \exists x_0 \in X, A_i(x_0) = 1,$$

$$(ii) \sum_i A_i(x) = 1, \forall x \in X. \quad (3)$$

It can be useful to represent the description \tilde{X}_0 by a line vector in the unit cube I^n (Kosko, 1992), where n is the cardinality of the descriptor set $\mathbf{A}(X)$ and

$$\tilde{X}_0 = [X_0(A_1) \dots X_0(A_n)]. \quad (4)$$

For instance, if \tilde{X}_0 represents the description of a given temperature $t_0 = 25^\circ\text{C}$ in Fig. 3, it can be represented by $\tilde{X}_0 = [X_0(\text{low}) \ X_0(\text{normal}) \ X_0(\text{high})] = [0 \ 0.3 \ 0.7]$.

Linguistic variables can be related by a set of rules to describe a linguistic model between their reference domains. Such models are computed by fuzzy systems described in the next section.

3.2. Fuzzy systems

A linguistic model relating variables x and y is written as a rule base, relating the terms $A_i \in \mathbf{A}(X)$ and $B_j \in \mathbf{B}(Y)$ in rules $\tilde{A}_i \rightarrow \tilde{B}_j$, read as

$$\text{if } x \text{ is } A_i \text{ then } y \text{ is } B_j. \quad (5)$$

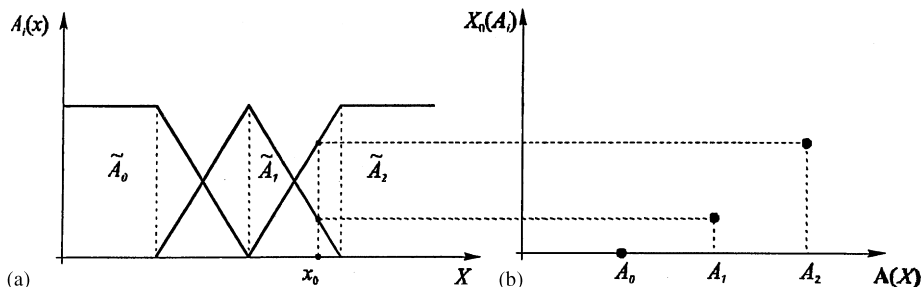


Fig. 3. Symbolic fuzzification.

The fuzzy propositions in the antecedent and consequent of rule (5) can be seen as flexible constraints in the values of the reference domains of each variable, represented by the fuzzy sets \tilde{A}_i and \tilde{B}_j .

The rule base models a linguistic relationship between the terms in the descriptor sets of the variables and is represented by the fuzzy relation $\tilde{\Phi}$ in the Cartesian product $\mathbf{A}(X) \times \mathbf{B}(Y)$ (Foulloy et al., 1995). Rules in the rule base are weighted by the membership value $\Phi(A_i, B_j)$ that represents how much the term A_i is related to the term B_j in the model described by the rule base. A value $\Phi(A_i, B_j) > 0$ means that the rule (i, j) occurs in the rule base with a weight $\Phi(A_i, B_j)$.

The rule base is thus represented by the matrix $\tilde{\Phi} \in I^{n \times m}$, where n and m are the cardinality of fuzzy partitions associated with $\mathbf{A}(X)$ and $\mathbf{B}(Y)$. For example, a simple rule base like

$$\tilde{A}_0 \rightarrow \tilde{B}_1, \quad \tilde{A}_1 \rightarrow \tilde{B}_0, \quad \tilde{A}_2 \rightarrow \tilde{B}_1 \quad (6)$$

can be represented by the matrix

| | B_0 | B_1 |
|-------|-------|-------|
| A_0 | 0 | 1 |
| A_1 | 1 | 0 |
| A_2 | 0 | 1 |

In the above example of the temperature, this rule base can express the reasoning that an alarm must be fired if the temperature is too high or too low. B_0 is associated with the concept “normal” and B_1 with “alarm”. The weights of the rules could be different from 0 or 1, to model uncertain knowledge.

In fuzzy systems, outputs $y_0 \in Y$ are computed from inputs $x_0 \in X$ in three steps:

- (1) *Fuzzification*, where the input description \tilde{X}_0 is computed from the input value $x_0 \in X$ as in Eq. (4).
- (2) *Inference*, where output description \tilde{Y}_0 is computed from \tilde{X}_0 and the rule base $\tilde{\Phi}$ using the fuzzy rela-

tional composition operator \circ as

$$\tilde{Y}_0 = \tilde{X}_0 \circ \tilde{\Phi}. \quad (7)$$

- (3) *Defuzzification*, where the output $y_0 \in Y$ is computed from the output description \tilde{Y}_0 , if a crisp output is needed.

By using the sum-product composition operator for fuzzy inference, the description \tilde{Y}_0 is easily computed for all $B_j \in \mathbf{B}(Y)$ as a vector matrix product by:

$$Y_0(B_j) = \sum_{1 \leq i \leq n} X_0(A_i) \Phi(A_i, B_j). \quad (8)$$

Eq. (8) shows that inference can be seen as a mapping between unit cubes (Kosko, 1992), which can be visualised in Fig. 4. In the above example, for input $\tilde{X}_0 = [0.3 \ 0.7]$, the output given by the rule base (6) is $\tilde{Y}_0 = [0.3 \ 0.7]$.

If more than one variable occurs in the antecedent of the rules, the input description is defined in the Cartesian product of the input descriptor sets (Babuska & Verbruggen, 1996). This will be presented later.

Defuzzification can be computed by the prototype values y_j , associated with fuzzy sets \tilde{B}_j , weighted by the description \tilde{Y}_0 (Foulloy & Galichet, 1995):

$$y_0 = \sum_{1 \leq j \leq m} Y_0(B_j) y_j. \quad (9)$$

Eq. (9) can be seen as a symbolic defuzzification when the values y_j represent terms of a smaller granularity set Y . For instance, in this work, the output descriptions are defuzzified into an index that represents colour codes to be displayed in the graphical interface.

In the example above, each term of the output descriptor set $\mathbf{B}(Y)$ can be associated with a primary colour, for example green for B_0 and red for B_1 . A colour palette of, for instance, 64 colours varying gradually from green to red, can be used to represent terms of a smaller granularity output descriptor set. This colour palette can be stored in a colour map vector where each element represents a colour in such a way that the colours associated with the terms B_0 and B_1 are respectively the first and the last elements of the colour map vector. From (9), the

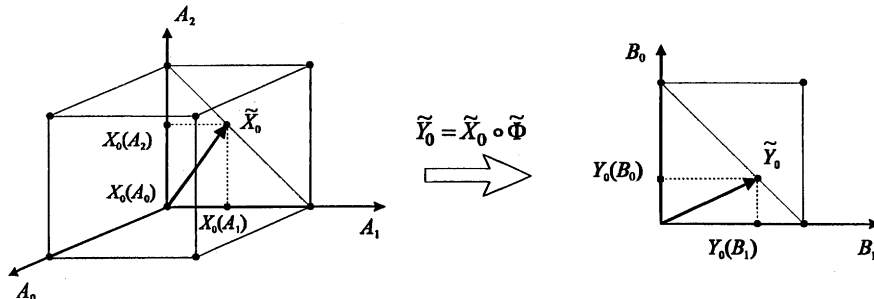


Fig. 4. Symbolic inference.

output description $\tilde{Y}_0 = [0.3 \ 0.7]$ is thus represented by the colour associated with the 45th position of the colour map vector.

Symbolic defuzzification used in this way, to represent output description as colours, is a very efficient way to improve human interpretation of machine conclusions. Colours are vague concepts by nature and can be easily interpreted by human operators of complex processes, allowing the symbolic representation of a large amount of information.

Before defuzzification, output descriptions can be used in multi-criteria decision-making as described below.

3.3. Multi-criteria decision-making

The general problem of multi-criteria decision-making can be formalised as follows (Zimmermann, 1996). Let $\omega_i \in \Omega$ be the set of possible solutions and $C_j \in \mathbf{C}$ a set of M criteria carrying some judgements on each possible solution for a given situation. The decision problem consists of choosing the “best solution” α_0 from the information contained in the criteria.

Generally, the solution to the decision problem is carried out in two steps:

- (1) The *aggregation step*: to aggregate criteria for each possible solution.
- (2) The *decision step*: to decide on the most suitable solution.

Fuzzy decision-making assumes that criteria are flexible constraints on the possible solution set represented by the fuzzy sets \tilde{C}_j , where the membership function value $C_j(\omega_i)$ represents the judgement of solution ω_i given by the criterion C_j .

The aggregation step results in the fuzzy set \tilde{G} , defined on the possible solution set Ω , whose membership function is computed by

$$G(\omega_i) = \mathbf{h}(C_1(\omega_i), \dots, C_M(\omega_i)), \forall \omega_i \in \Omega \quad (10)$$

where $\mathbf{h} : [0, 1]^M \rightarrow [0, 1]$ is an aggregation operator. Several aggregation operators have been proposed in the literature (Grabish, 1995) whose characteristics depend on the application.

The decision step is computed by a decision rule. The most common decision rule is the maximum rule, where the solution is computed by

$$\alpha_0 = \omega_i / G(\omega_i) = \max_{\forall \omega_j \in \Omega} (G(\omega_j)). \quad (11)$$

Nevertheless, other decision rules can be used (Denoex, 1997).

In this work, symbolic descriptions of the process variable states are computed by fuzzification of features extracted from the residuals. Possible solutions ω_i are taken as symbolic terms representing decisions in the detection and isolation modules. The criteria \tilde{C}_j represent

descriptions of partial conclusions computed by fuzzy inference.

4. Fuzzification

Knowledge about a given situation is generally imperfect. Doubts can arise either about knowledge validity (it is said to be uncertain), or about how to express knowledge clearly (it is vague or imprecise). These two types of imperfections are different but often intimately linked.

Noise measurements and modelling approximations are sources of uncertainty in the decision to classify on-line a process state with measurements. When the linguistic terms are used in diagnostic reasoning, difficulties arise from the lack of well-defined thresholds that might separate linguistic terms, which are vague concepts and should be modelled as such.

In this work, imprecision due to noise is taken into account by combining past and present information within residual signals. The residuals used in the detection or isolation modules are different, but the underlying uncertainty modelling is the same. Feature extraction and fuzzification will be described for a generic residual $r(k)$.

There are many works in the literature that attempt to symbolically describe the evolution of a dynamic process (Cheung & Stephanopoulos, 1990; Rengaswamy & Venkatasubramanian, 1995). Generally, extracting some features from signals, and further interpreting these as process trends is part of this symbolic description. In many cases, these features correspond to signal derivatives. Nevertheless, derivatives are corrupted if the signal is noisy, leading to further imprecision.

In this work, residual features are used to take noise into account. For each process variable, two sets of features are computed from residual values in the last instants:

1. *Residual values* in the τ_1 last instants, represented by the vector $\mathbf{v}(t)$, whose components are:

$$v_i(t) = r(t - i), \quad i = 0, \dots, \tau_1 - 1. \quad (12)$$

2. *Residual variations* with respect to present time, in the τ_2 last instants, represented by the vector $\dot{\mathbf{v}}(t)$, whose components are

$$\dot{v}_j(t) = r(t) - r(t - j), \quad j = 1, \dots, \tau_2, \quad (13)$$

where the time windows τ_1 and τ_2 are chosen a priori. Time window τ_1 is chosen as a specification of the fault persistence (Theilliol, Weber, Ghetie & Noura, 1995). Time window τ_2 is chosen to improve the robustness of the decision (while giving rise to a delay), because it corresponds to several derivative estimations from several temporal windows.

The set of features are taken as independent measurements and are used to compute partial conclusions on the process state. If the partial conclusions are very different, then there is lot of imprecision that will account for uncertainty in the final decision, otherwise the final decision is more reliable.

The reference domain of residual value and variation are noted as $X, \dot{X} \subset R$ respectively and are considered to be the same in the detection and isolation modules. Orthogonal normalised fuzzy partitions are defined in the universe of each residual feature. The descriptor sets associated with each feature fuzzy partition are:

$$\mathbf{A}(X) = \{NN, N, Z, P, PP\}, \quad \mathbf{B}(\dot{X}) = \{N, Z, P\} \quad (14)$$

related respectively to residual value and variation features. The terms describe a qualitative order of magnitude of the feature values, and their linguistic interpretations are those of current applications of fuzzy control.

In the example, symmetrical trapezoidal or triangular membership functions are used for both fuzzy partitions (Fig. 5). Symmetry is justified here by the fact that in the absence of a fault, residuals are zero, and positive or negative faults have the same importance. The symmetry leads to a simple parameterisation of the residual value fuzzy partition with only four parameters: a_1, a_2, a_3 and a_4 , corresponding to the trapezoid boundaries.

Parameter a_1 can be chosen from a rough idea of the noise standard deviation. Parameter a_4 corresponds to a threshold as in a classical alarm system. Parameters a_1 and a_4 represent the conventional thresholds for normal and faulty states. Taken as being equal to a_3 , parameter a_2 can be equal to the mean of a_1 and a_4 , leading to a triangular membership function for P and N in $\mathbf{A}(X)$, thus avoiding further parameterisation.

The parameters for the residual variation partition are defined from the residual value partitions. As symmetrical trapezoidal membership functions are used, the three fuzzy sets can be defined with two parameters \dot{a}_1 and \dot{a}_2 . It can be seen from (13), that a deviation in the residual affects the variation attribute twice, hence the parameters for residual variation partition can be derived from those of the residual value partition as $\dot{a}_1 = 2a_1$, $\dot{a}_2 = 2a_2$.

At a given instant t , the residual value and variation features are computed in the respective time windows. For simplicity, the time instant t will be omitted in some

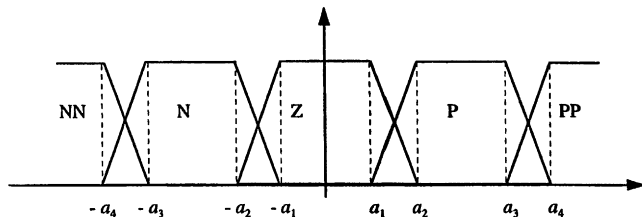


Fig. 5. Fuzzy partition.

equations. The fuzzification of residual features leads to the descriptions:

$$\tilde{V}_i = [NN(v_i) \dots PP(v_i)] \quad (15)$$

for residual value and

$$\tilde{V}_j = [N(\dot{v}_j) \dots P(\dot{v}_j)] \quad (16)$$

for residual variation.

The linguistic description of the residual in a process variable at each instant is represented by a fuzzy proposition written from residual features as

$$v_i(t) \text{ is } A \text{ and } \dot{v}_j(t) \text{ is } B \quad (17)$$

where $A \in \mathbf{A}(X)$ and $B \in \mathbf{B}(\dot{X})$. Fuzzy propositions like (17) are used as inputs to fuzzy inference in detection and isolation modules.

At each instant, each of the ordered pairs $(v_i(t), \dot{v}_j(t))$ of feature values leads to the description of the residual in the process variable computed as the vector:

$$\tilde{W}_k = [t(NN(v_i), N(\dot{v}_j)) \dots t(PP(v_i), P(\dot{v}_j))] \quad (18)$$

where $\tilde{W}_k \in I^{15}$ is the fuzzy description computed as the conjunction of the descriptions $\tilde{V}_i \in I^5$ and $\tilde{V}_j \in I^3$ by the t -norm t . In the application described in this work, the product operator was used as t -norm.

In order to represent residual imprecision, all $M = \tau_1 \tau_2$ feature combinations are taken into account as independent measurements of the process state. The resulting input descriptions, represented by the matrix $\tilde{W} \in I^{M \times 15}$ are used to compute partial conclusions in the detection and isolation modules. Partial conclusions are then aggregated into a final decision as described in the next sections.

5. Detection module

The analytical model used for detection is a numerical non-linear simulator of the process that provides the reference for the fault-free behaviour. The simulator is fed on-line the set points chosen by the operators and the measurable disturbances.

For each variable, the *measurement* value $y(t)$ at time t can be written from the *reference* value $y^*(t)$ given by the numerical simulator output at the same time as:

$$y(t) = y^*(t) + \delta(t), \quad (19)$$

where $\delta(t)$ is the *total residual*.

Detection decisions can be formulated linguistically as: “if $\delta(t)$ is near to zero then the variable follows its normal behaviour, including modelling errors and measurement noise; otherwise, there is a fault, or a non-measurable disturbance on the variable, that must be detected”.

To take imprecision into account, residual value and variation features for the detection module, noted as $x_i(t)$ and $\dot{x}_j(t)$ respectively, are computed as in (12) and (13).

Fuzzy partitions are defined over the universe of residual features for each process variable as in (14). At each instant, the residual feature descriptions \tilde{X}_i and \tilde{X}_j are computed as in (15) and (16).

A rule base written as follows expresses partial conclusions about the variable state:

$$\text{If } x_i(t) \text{ is } A \text{ and } \dot{x}_j(t) \text{ is } B \text{ then state is } S \quad (20)$$

where $A \in \mathbf{A}(X)$, $B \in \mathbf{B}(\dot{X})$ and the conclusion $S \in \{OK, FA\}$ represents the variable state, where OK accounts for the normal state and FA for the faulty state. For a better performance, weights are associated with each rule in the rule base; this is represented in Table 1. For instance, if the residual is very negative and its variation is negative too, the variable is certainly in alarm, while if the residual is very negative with a positive variation, the alarm is not so drastic.

At each instant, residual descriptions computed from $\tilde{X}_i \in I^5$ and $\tilde{X}_j \in I^3$ (15), (16), are represented by the matrix $\tilde{\mathbf{W}} \in I^{M \times 15}$. In this matrix, rows correspond to time instants and columns correspond to the feature combinations (18). Residual descriptions are used as inputs to the fuzzy system that computes the membership values to the states OK and FA by the relational composition:

$$\tilde{\mathbf{S}} = \tilde{\mathbf{W}} \circ \tilde{\Phi}, \quad (21)$$

where $\tilde{\Phi} \in I^{15 \times 2}$ is the matrix representation of the rule base (Table 1) and $\tilde{\mathbf{S}} \in I^{M \times 2}$ contains descriptions of partial conclusions about the process variable state. The rows of $\tilde{\mathbf{S}}$ correspond to the fuzzy description (see (4)) of the rule base conclusions at each time instant. The first column of $\tilde{\mathbf{S}}$ is associated with the conclusion OK and the second column to the conclusion FA . Since normalised orthogonal fuzzy partitions are used, the partial conclusions associated with each residual feature combina-

tion are complementary so that

$$S_k(FA) = 1 - S_k(OK), \quad k = 1, \dots, M. \quad (22)$$

The final decision takes all the past M sampling times into account. This allows the filtering of noise. Moreover, in order to represent uncertainty in the final decision, two final conclusions are computed by the aggregation of all partial conclusions about the variable state descriptions by two ordered weighted aggregator (OWA) operators (Yager, 1988):

(a) *robust* final conclusion \tilde{D}^- as:

$$D^-(FA) = OWA^-(S_1(FA) \dots S_M(FA)). \quad (23)$$

$$D^-(OK) = OWA^+(S_1(OK) \dots S_M(OK)). \quad (24)$$

(b) A *sensitive* final conclusion \tilde{D}^+ as

$$D^+(FA) = OWA^+(S_1(FA) \dots S_M(FA)), \quad (25)$$

$$D^+(OK) = OWA^-(S_1(OK) \dots S_M(OK)), \quad (26)$$

where OWA^+ and OWA^- are respectively an *or-like* OWA operator and an *and-like* OWA operator (Yager, 1993). These operators can be chosen in such a way that final conclusions remain complementary.

$$D^-(OK) + D^-(FA) = 1, \quad (27)$$

$$D^+(OK) + D^+(FA) = 1. \quad (28)$$

In the application described in Section 7, the max and the min operators used were OWA^+ and OWA^- respectively.

At each instant, uncertainty is bounded by the robust and sensitive final conclusions. From the difference between Eqs. (27) and (28), uncertainty bounds can be computed as

$$\zeta = D^+(FA) - D^-(FA) = D^-(OK) - D^+(OK) > 0. \quad (29)$$

Table 1
The detection symbolic rule base^a

| $\tilde{\Phi}$ | | \dot{x}_j | | | | | | | | | | | | | |
|----------------|-----|-----------------|------|------|-----------------|-----|-----|-----------------|-----|-----|-----|-----|-----|------|------|
| | | N | | | Z | | | P | | | | | | | |
| x_i | NN | 0/OK + 1/AL | | | 0/OK + 1/AL | | | 0.2/OK + 0.8/AL | | | | | | | |
| | N | 0/OK + 1/AL | | | 0.4/OK + 0.6/AL | | | 0.6/OK + 0.4/AL | | | | | | | |
| | Z | 0.8/OK + 0.2/AL | | | 1/OK + 0/AL | | | 0.8/OK + 0.2/AL | | | | | | | |
| | P | 0.6/OK + 0.4/AL | | | 0.4/OK + 0.6/AL | | | 0/OK + 1/AL | | | | | | | |
| | PP | 0.2/OK + 0.8/AL | | | 0/OK + 1/AL | | | 0/OK + 1/AL | | | | | | | |
| Φ | | $x_i \dot{x}_j$ | | | | | | | | | | | | | |
| | | NN-N | NN-Z | NN-P | N-N | N-Z | N-P | Z-N | Z-Z | Z-P | P-N | P-Z | P-P | PP-N | PP-Z |
| OK | 0,0 | 0,0 | 0,2 | 0,0 | 0,4 | 0,6 | 0,8 | 1,0 | 0,8 | 0,6 | 0,4 | 0,0 | 0,2 | 0,0 | 0,0 |
| AL | 1,0 | 1,0 | 0,8 | 1,0 | 0,6 | 0,4 | 0,2 | 0,0 | 0,2 | 0,4 | 0,6 | 1,0 | 0,8 | 1,0 | 1,0 |

^aThe transpose of the matrix Φ describing the rule base.

The final decision step follows the aggregation step (see (11)). Uncertainty modelling makes it possible to work on different limits to detect a variable in alarm or to decide to turn off the alarm.

A fault is detected on a variable if the *FA* membership is significant. The robust final conclusion is used for fault detection and to trigger an alarm in a process variable by the following decision rule:

$$D^-(FA) - D^-(OK) > \varepsilon_1, \quad (30)$$

where ε_1 is a *safety factor*, chosen according to required robustness, that acts as an ambiguity rejection (Frélicot, Masson & Dubuisson, 1995). Note that if $\varepsilon_1 = 0$, Eq. (30) reduces to the max decision rule. This crisp decision is used to represent the alarm status on the classical interface or on the graph that is presented in Section 7.

The supervision system follows the process on-line and the sensitive final conclusion is used to turn the alarm off if

$$D^+(OK) - D^+(FA) > \varepsilon_2. \quad (31)$$

The I^2 decision space is shown in Fig. 6, where the vectors $\tilde{D}^+ = [D^+(OK) \ D^+(FA)]$ and $\tilde{D}^- = [D^-(OK) \ D^-(FA)]$ are represented in two situations (alarm, normal). Relations (27)–(29) are visualised in this figure. Normal, faulty and ambiguous regions are highlighted. An alarm is set off if the robust final conclusion description vector is in the faulty region. Otherwise, an alarm is turned off if the sensitive description vector is in the normal region.

Fig. 7 presents the residual evolution during a fault simulation. Figs. 8 and 9 show the evolution of robust and sensitive final conclusions in the simulation of the fault shown in Fig. 7, where an ambiguity band can be observed. It can be seen that in spite of measurement noise that is more or less constant over the temporal window, uncertainty in the decision occurs only between the ambiguity bounds.

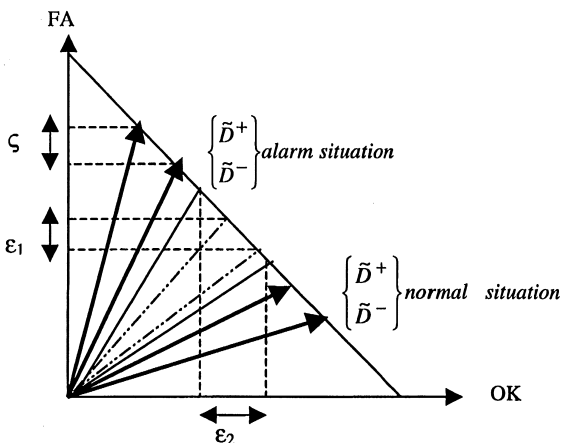


Fig. 6. Decision interpretation.

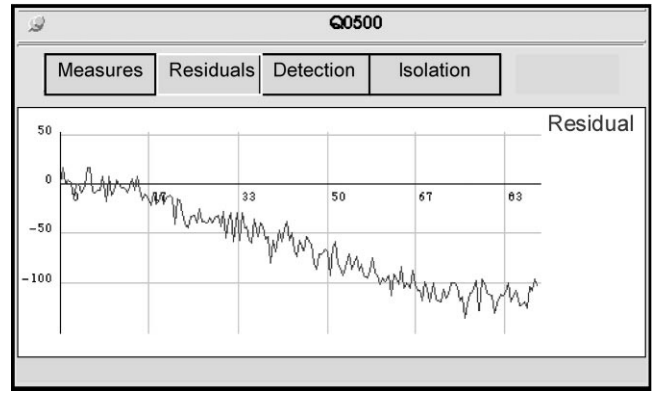


Fig. 7. Residual evolution in a variable.

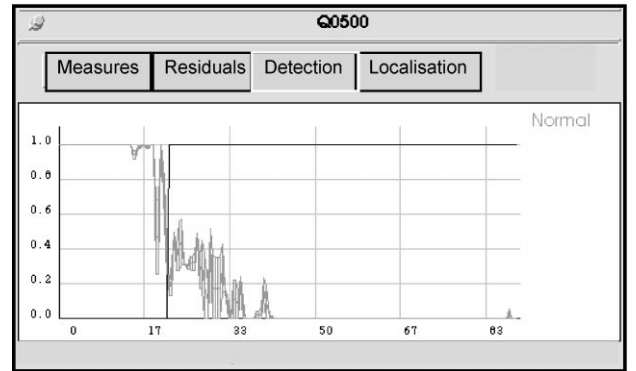


Fig. 8. *OK* decision evolution.

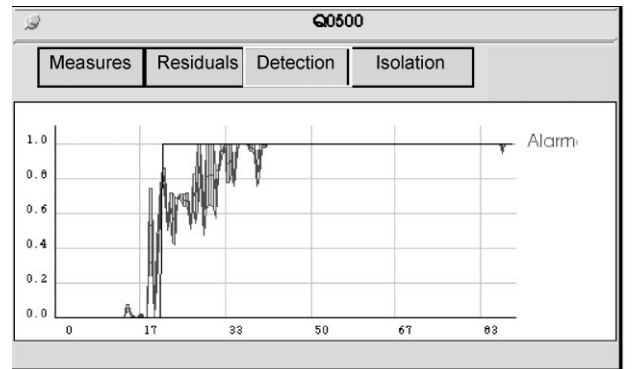


Fig. 9. *FA* decision evolution.

The mean description $\tilde{D} = \frac{1}{2}(\tilde{D}^+ + \tilde{D}^-)$ is defuzzified into an index representing a colour code to be displayed in the graphical interface (see Section 7).

A meta-algorithm is given in the appendix to depict the various computational steps.

The detection module provides a unordered list of the simultaneous faults that affect the process. The isolation module aims to convert the set of detected variables into a graph whose root is the primary fault, and whose paths illustrate the fault propagation.

6. Isolation module

Causal modelling requires the description of a complex process with a chain of elementary partial effects. For instance, the output flow of a tank explains its level; the level is explained by the input flow; the valve opening explains the input flow. Therefore, the causal model provides a conceptual tool with which to reason how changes propagate or how they may be explained.

A digraph, whose nodes represent the variables and whose arcs represent directed relationships between variables, can describe the causal model. Note that contrary to fault-symptom trees, the causal graph represents the normal behaviour of the process (Leyval et al., 1994).

In this work, the causal links of the graph represent the local linear discrete model of each measured variable, whose transfer functions have been identified as approximations of the non-linear simulator behaviour. Additionally, the causal model is fed with the outputs of the analytical model, which allows the linear approximation errors to be estimated.

For each variable, its *predicted* value $\hat{y}(t)$ is computed from the measurements of each antecedent, by the sum of the components associated with each entering arc (Fig. 10) as

$$\hat{y}(t) = G(q)\mathbf{u}(t), \quad (32)$$

where $\mathbf{u}(t) = [u_1(t) \dots u_n(t)]'$ is the measurement of each antecedent and $G(q) = [G_1(q) \dots G_n(q)]$ is the local model of the variable, written in the shift operator q . Expressing the measured values of the antecedents according to (19):

$$\hat{y}(t) = G(q)\mathbf{u}^*(t) + G(q)\boldsymbol{\delta}_u(t), \quad (33)$$

where $\boldsymbol{\delta}_u(t) = [\delta_1(t) \dots \delta_n(t)]'$ are the total residuals in the antecedents.

Assuming linearity allows the effects of each potential antecedent on a variable to be separated. This means that antecedent faults propagate and that their effects are summed on the successor node.

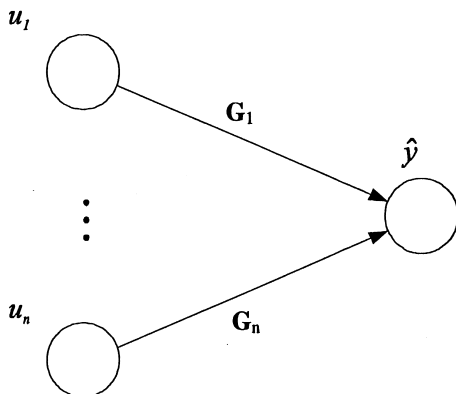


Fig. 10. Causal model.

The predicted value of the variable can be rewritten from (33) as

$$\hat{y}(t) = \hat{y}^*(t) + \hat{\delta}(t), \quad (34)$$

where $\hat{y}^*(t)$ is the *estimation of the reference* value and $\hat{\delta}(t)$ is the *propagated residual* that accounts for propagated effects of faults in variables upstream in the graph.

Linear approximation errors in each variable can be estimated by the difference between the reference value computed by the analytical model and the estimation of the reference value computed by the local model of the variable as

$$y^*(t) - \hat{y}^*(t) = \Delta\hat{y}^*(t). \quad (35)$$

The isolation procedure is carried out for all variables in the process. For each variable, the *local residual* $\Delta\delta(t)$ is defined as the effect of local faults only. It can be computed by the difference between the total residual $\delta(t)$ (19), which is sensitive to all faults, and the propagated residual $\hat{\delta}(t)$ (34), which is only sensitive to upstream faults. Taking into account linear approximation errors, the local residual can be written as

$$\Delta\delta(t) = \delta(t) - \hat{\delta}(t) = y(t) - \hat{y}(t) - \Delta\hat{y}^*(t). \quad (36)$$

A purely numerical interpretation of (19), (34) and (36) should be the most accurate way to process the initial/final-node consistency-test necessary for the isolation step if the model was perfect and the measurements were very precise. Nevertheless this purely numerical interpretation cannot be soundly performed if the thresholds of these various residuals, which are connected by arcs, are poorly balanced. Indeed when a malfunction or a unmeasurable disturbance occurs, inappropriate thresholds may prevent the defect from being immediately detected on the root variable, or may upset the chronological sequence of fault occurrences in the causal chain: the primary fault may be detected first on one of its successors on the graph. To avoid a wrong diagnosis situation, these thresholds should be determined by using the very accurate quantitative information obtained from a sensitivity analysis. This analysis is a rather dissuasive prospect for a process with a large number of variables.

These considerations explain why the isolation procedure is modeled (as is the previous detection step) as a decision-making process in an imprecise context, and the isolation decision is then linguistically stated as follows: “*if the local residual is near zero, then the detected disturbance is due to the propagation of upstream faults; otherwise there is a local fault*”. Note that even in the presence of multiple faults, the local residual should isolate local faults.

The rule base for the isolation module is expressed as follows:

$$\text{if } z_i(t) \text{ is } A \text{ and } \dot{z}_j(t) \text{ is } B \text{ then fault is } F \quad (37)$$

where $z_i(t)$ and $\dot{z}_j(t)$ are the local residual features computed as in (12) and (13); $A \in \mathbf{A}(X)$ and $B \in \mathbf{B}(X)$ are terms in the fuzzy partitions (14), which are the same as those used in the detection module, avoiding new parameters. The conclusion $F \in \{UP, LO\}$, where *UP* stands for *upstream* faults and *LO* for *local* faults.

The input description matrix $\tilde{\mathbf{W}} \in I^{M \times 15}$ is now computed from feature descriptions $\tilde{Z}_i \in I^5$ and $\tilde{Z}_j \in I^3$ as (18). The fuzzy inference for the isolation module can use the same rule weights as the detection module avoiding further parameterisation. Thus, Table 1 represents the rule base for localisation too, with *UP* instead of *OK* and *LO* instead of *FA*. For instance, if the local residual is very negative and its variation is negative too, the fault is certainly local, while if the residual is near zero with a positive variation, the fault is certainly upstream.

The fuzzy description of the fault isolation is computed by

$$\tilde{F} = \tilde{\mathbf{W}} \circ \tilde{\Phi}, \quad (38)$$

where $\tilde{F} \in I^{M \times 2}$. The first column of \tilde{F} is associated with the conclusion *UP* and the second column with the conclusion *LO*.

Obviously, a decision on whether the fault is local or upstream depends on the decision that a fault has occurred in the process variable, computed by the decision rule (30).

The membership values of decisions *LO* and *UP* are computed in such a way that

$$F(LO) + F(UP) = D(FA), \quad (39)$$

where $D(FA)$ is the membership of *FA* in the mean description in the detection module. Due to fuzzy partitions properties (3), equality in (39) is achieved by multiplying the result of (38) by $D(FA)$.

The final conclusions of the isolation are computed in the same way as the detection ones. A robust description \tilde{L}^- and a sensitive description \tilde{L}^+ are computed by the aggregation of partial conclusions as in (23)–(26). The final decision that a fault is local is computed as in (30) and a mean valued description is also used for defuzzification into a colour code to be used in the graphical interface.

A meta-algorithm is given in Appendix to depict the various computational steps.

7. Prototype description

7.1. Process description

An application of the method was carried out at a nuclear fuel reprocessing plant (Fig. 11). Spent fuel is reprocessed to recover the uranium and plutonium still present in the fuel rods irradiated in nuclear reactors, and to isolate the remaining fission products. The pulsed

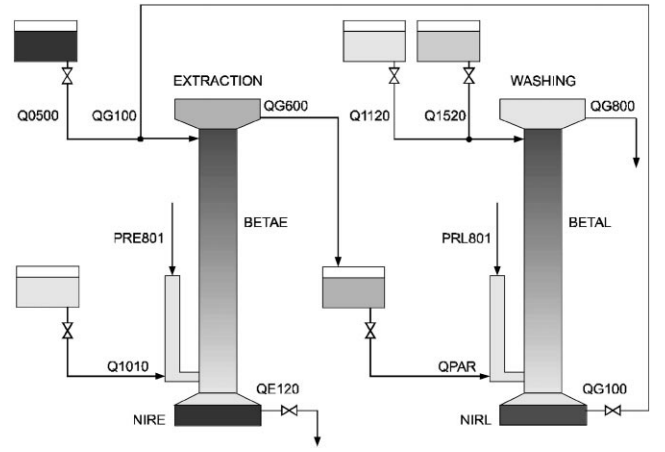


Fig. 11. Schematic view of the plant.

column facility considered here includes extraction columns and fission product scrubbing columns, and is designed to separate the uranium and plutonium from the fission products (FP) by selective extraction.

A pulsed column is a liquid-liquid extraction device. The spent fuel (comprising uranium, plutonium and FP) is dissolved in nitric acid (Q0500), and the extraction column selectively transfers the uranium and plutonium to an organic phase (inlet Q1010 and outlet QG600) consisting essentially of tributylphosphate (TBP); thus most of the fission products remain in the aqueous phase (QE120).

The extraction step requires that the aqueous and organic phases be thoroughly mixed to maximize the contact surface area between the two solvents and thereby optimise the chemical exchange phenomena. The two phases tend to separate by gravity, as nitric acid has a higher density than TBP.

In order to ensure countercurrent flow, the light phase (Q1010) is injected at the base of the column and the heavy (acid) phase at the top (Q0500 and QG100). The organic phase is the continuous phase; it initially fills the entire volume of the extraction column, and remains the predominant phase after injection of the aqueous phase: acid droplets are dispersed in the TBP. The resulting mixture is subjected to periodic pressure pulses (PRE801) to form an emulsion in order to delay the descent of the heavy phase and to mix it with the light phase. The interface — the surface physically separating the two phases — is located in the settler at the bottom of the column (NIRE), and is regulated by drawing off the aqueous phase (QE120) at a suitable rate. In the application considered here, all liquid transfers are ensured by airlifts.

The facility is comprised of two columns coupled top-to-bottom by means of a transfer system. The extraction column described above is followed by a scrubbing column designed to enhance the decontamination of fission

products from the solvent (aqueous phase: inlet Q1120 and Q1520, outlet QG100; organic phase: inlet QPAR; outlet QG800). Traces of fission products carried out from the extraction column (QG600) are rinsed from the organic phase containing the uranium and plutonium.

The columns are connected top-to-bottom (QG100) in order to recover the aqueous phase containing the fission products through a single outflow (QE120), and to increase the process efficiency by recycling the traces of uranium and plutonium in the aqueous phase from the scrubbing column.

A simulator of the process, based on a set of non-linear numerical models, is used to provide reference behaviour and also to simulate faulty behaviour. Nuclear physicists wrote this simulator; it is based on sound mathematical models of the plant. The process simulator computes 50 variables, which are used by the prototype system and can simulate several kinds of faults (sensor, actuator and process faults).

A careful physical analysis (Leyval et al., 1994) of the process under study has resulted in a causal graph linking input and output flows, levels and column weights (Fig. 12).

7.2. Application of detection and isolation

At each time, the fuzzy reasoning procedure for detection and isolation presented above is carried out for each process variable. As an example a drift fault on the variable Q0500 is analysed (Fig. 13; see in Fig. 7 the residual evolution); this fault occurs during a ramp set point change, which is very difficult to detect.

The parameters of fuzzy partitions have been chosen without any precise quantitative analysis for the total residual. Before the fault, residual values due to noise give an idea of the value of parameter a_1 (see Fig. 5). The

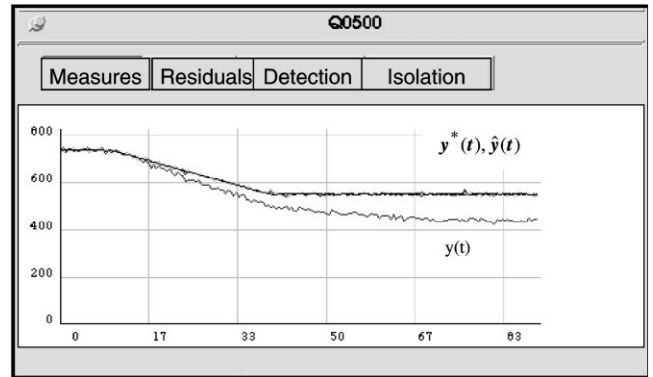


Fig. 13. Q0500 flow behaviour during a drift fault and a ramp set point change.

variable is certainly faulty when the residual value is greater than -50 , what defines the value of parameter a_4 . The same procedure has been applied to each process variable. Simulation results have shown that the diagnostic performance is not very sensitive to such parameters. The values of parameters τ_1 and τ_2 have been chosen by analysing the sensitivity and robustness of the detection for some variables. The values $\tau_1 = 2$ and $\tau_2 = 4$ have shown good results and were adopted for all variables.

The variable Q0500 is the input flow solution containing radioactive products. The effects of a fault that affects this variable is propagated over the extraction column and Fig. 14 shows the behaviour of the variable QE120, which is the output flow of the heavy phase.

An arc in the causal model of the process links the variables Q0500 and QE120. The fault appeared first on variable Q0500 and can be detected in variable QE120 some instants later. The isolation reasoning is illustrated in Figs. 13 and 14. In Fig. 14, the predicted value $\hat{y}(t)$ of variable QE120 is close to the measured $y(t)$,

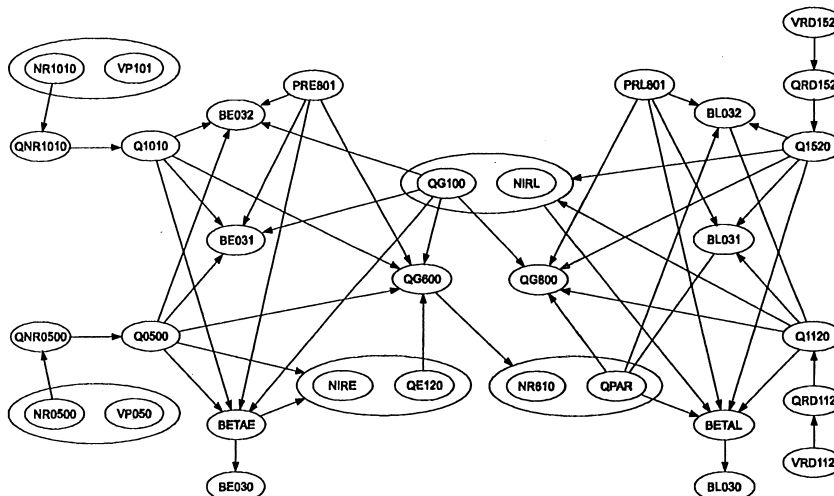


Fig. 12. Causal model of the plant.

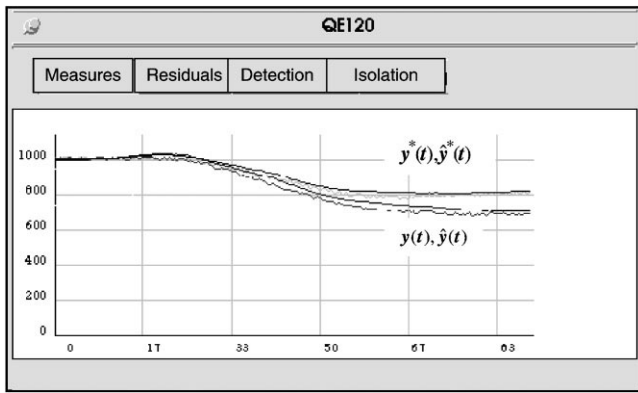


Fig. 14. QE120 flow behaviour (same fault as Fig. 13).

demonstrating that its behaviour can be explained by upstream variable behaviours, the fault is thus upstream. This is not the case in Fig. 13 for Q0500, where the difference between the measured value $y(t)$ and the predicted value $\hat{y}(t)$ is large, indicating that the fault is local. The isolation module reproduces such reasoning for every variable of the process.

7.3. Graphical interface

The information about the process is presented in three different abstract layers in the prototype system. The layers present different information in the screen

area. The basic frame window presents the list of the process variables on the right side. By clicking on each button, the corresponding variable history information is displayed (see Figs. 13 and 14).

In the first layer, the plant is represented according to a structural decomposition as in classical monitoring interfaces (Fig. 15). In this view, the variables are represented by a rectangle containing their measured value or a colour representation obtained from the fuzzy reasoning in the detection module. The primary fault, found by the isolation module, is shown in this view by a red contour around the name of the variable.

The illustrated example corresponds to a pressure-pulse primary fault problem of the scrubbing column (PRL801). As a quasi-immediate effect, the heavy phase inner flow of the scrubbing column is affected, and consequently the column weights (BL030, BL031 and BL032) characteristic of the emulsion operating mode are abnormally modified. To keep the level (NIRL) of the settler at the bottom of the column constant, the heavy-phase outlet (QG100) is finally subject to abnormal variations. This view only proposes the set of the faulty variables at a given time.

In the second layer, a first abstraction is made and the process variables are represented as nodes in the causal graph. The output of the fuzzy reasoning in the isolation module is displayed in the graph in the following way. If the node is not red, the variable is in a normal state. If the contour of the node is red, the alarm in the variable does

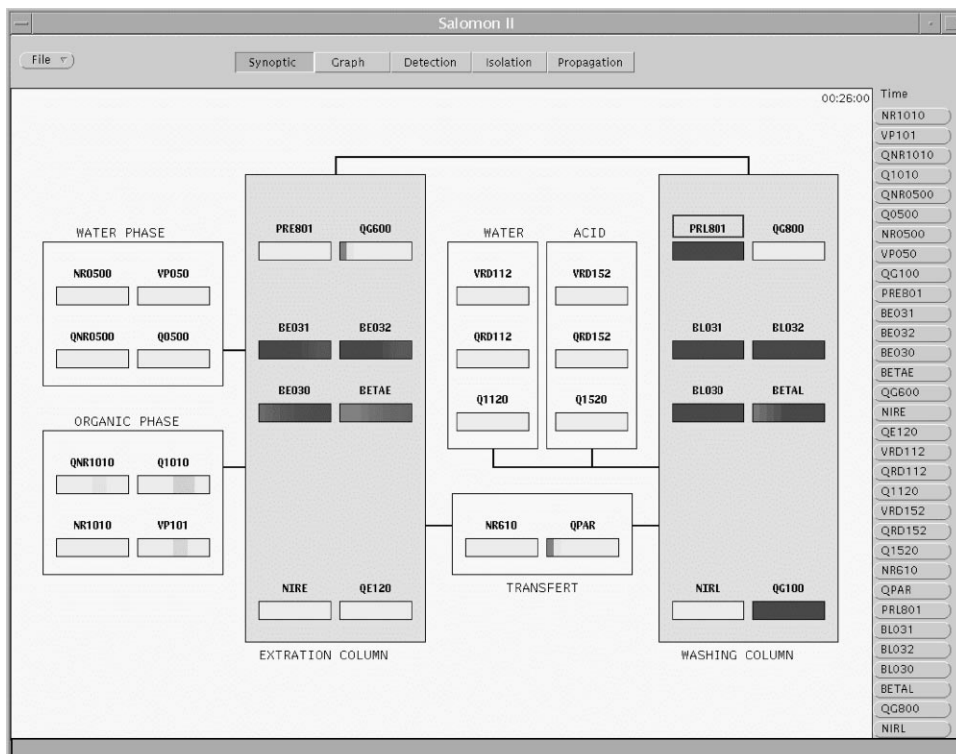


Fig. 15. The first layer, structural modules of the plant.

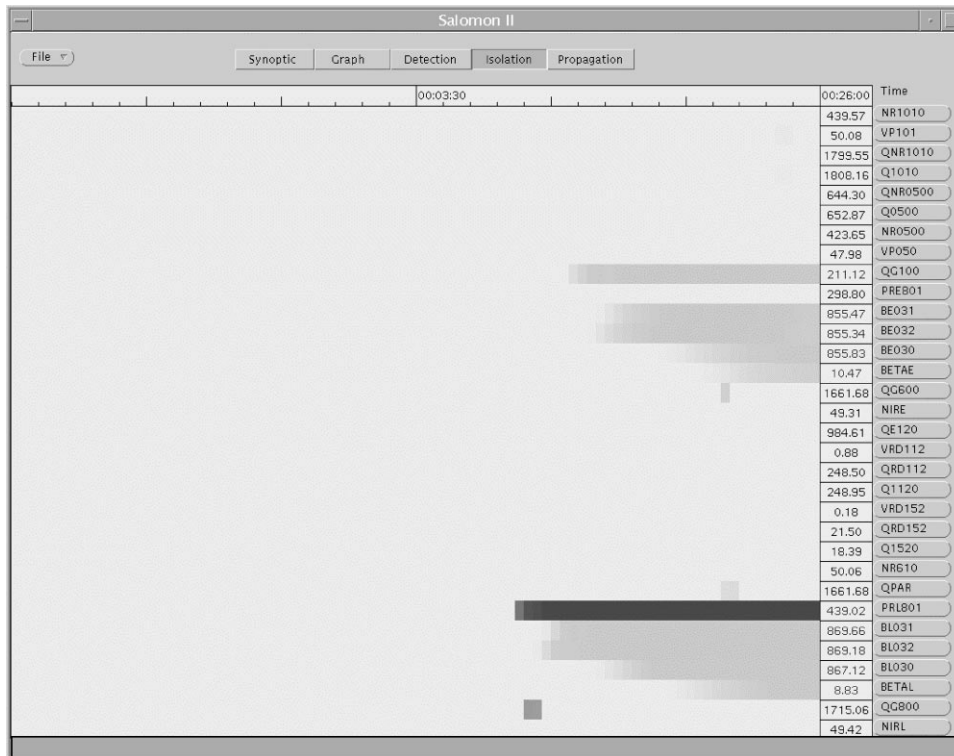


Fig. 18. Isolation view.

to red (faulty state). In the isolation view (Fig. 18), the conclusions of the decision are defuzzified into an index of a colour map from blue (upstream fault), to red (local fault). A colour palette composed of 64 colours is used.

In a normal situation the whole screen is green in both detection and isolation views. In the detection view, when a fault occurs, the lines turned to red represent the variables in alarm. The evolution of the colour intensity represents the abruptness of the fault. In the isolation view, the variables in green in the detection view remain green. The variables in red remain red if the fault is local or turn blue if the fault is upstream. By changing from the detection to isolation view the effect of alarm filtering provided by the system can be evaluated.

The isolation view and causal graph representation can be used concurrently in isolation task reasoning. With the isolation view it is very relevant to analyse the times of appearance and disappearance of correlated alarms to get the right isolation reasoning. But this implies that all the relationships between variables are well assimilated by the operators. The causal graph highlights such relationships for novice operators and shows how the fault propagates throughout the process.

The detection and isolation views and the causal graph are different representations of the pressure pulse failure mode at the same time. The causal graph is a powerful tool to express the links between abnormal or normal variable behaviours. The physical influence of the pressure pulse on the outlet QG100 is not necessarily obvious

to a novice operator in abnormal situations. The *column reading* of the detection or isolation view provides a natural representation of simultaneous or past events essential in diagnosing the temporal reasoning of a process expert.

The last view of the third layer shows the predicted behaviour of the process (Fig. 19). In this view, current time measurements are shown in the middle of the screen. The left-side screen shows past behaviour while the right-side screen shows the behaviour predicted by the causal model, according to the same colour representation as in the detection view.

The third layer attempts to present the timed behaviour of the process state. The objective of this interface is to present a large amount of qualitative information about the fault's evolution to help the operator analyse the chain of alarms. Moreover, the isolation interface shows an image that can be seen as a graphical representation of the fault. In the propagation view, the operator is informed at the very beginning of the fault if its consequences will affect a large number of variables. In a practical application, this view could be used to test alternative actions.

7.4. Evaluation

A rough evaluation of the prototype system was made with eight students at a technical school (Evsukoff et al., 1998). The students were arranged into two groups with

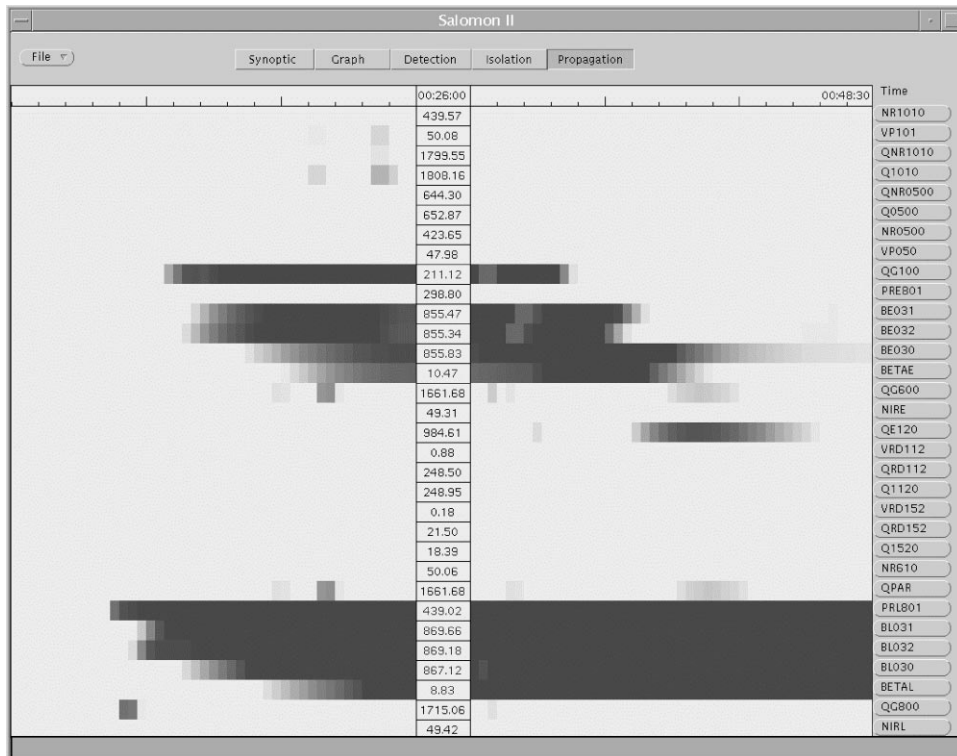


Fig. 19. Propagation view.

equivalent skills. Each group tried a version of the system: the first group tried the full system, while the second group did not have access to the second layer with the causal graph information.

Experimentation was carried out with one student at a time. All students were confronted with the same four fault scenarios of increasing difficulty. At the end of each scenario, the students were asked to explain the process evolution. At the end of each experiment, the students were asked to answer general questions about the interface and more specific questions about the evolution of some variables in each scenario.

The aim of the experimentation was to test if students could understand the notions of model-based supervision and causal reasoning. It is clear that the limited number of students and the short time given to each of them did not allow precise statistics but only a qualitative evaluation of the relevance of the different ideas for future process operators.

Classically, the main representation of the physical phenomena propagation used by students is the flow diagram. So when the fault effects can be observed on a single phase, students easily manage to make the right diagnosis because the fault propagation matches the flow diagram circulation.

When the fault is more difficult to diagnose, involving dynamic relationships that are not included in the flow diagram (such as a fault in a regulation loop), it was difficult for the students, who are not experts in

liquid-liquid extractors, to give the right explanation. Nevertheless, those using the graph interface gave a clear description of the underlying phenomena.

Consequently the causal graph seems to constitute a good support to explain the fault propagation through the process when the problem to be diagnosed is complex. Moreover, the causal graph is a natural way to explain alarm filtering to the operator.

Obviously, experimental conditions were far from ideal. Nevertheless, at the end of the relatively short period of experimentation, first-time users of the prototype system were able to understand most of complex physical behaviour of the process and to diagnose various types of faults.

8. Conclusion

This paper has presented an alarm filtering method based on different models of a process: a non-linear precise model for detection and a bank of approximate local linear models structured as a causal graph for isolation. These tasks have been designed from a cooperative human/machine point of view. Consequently, their first objective is to make the operators focus on suspicious variables before the discrepancy with normal behaviour becomes too significant, thus enabling them to act appropriately, sure of the real cause of the observed malfunction. For such a system to be efficient,

explanations of primary fault localisation must be given to the operator. One foundation of this system is causal reasoning, visualised as a cause-effect graph that is the key to explaining isolation reasoning.

Fuzzy reasoning is another basis of the human/machine co-operation because it allows the symbolic interpretation of numerical data such as model residuals and the implementation of gradual decisions. Residuals are basic quantities for model based diagnostics; most works focus on structured residual generation. In classical control approaches, residuals are considered to be Boolean quantities (equal to zero or not) for detection and isolation reasoning. A zero residue means no fault while a non-zero residue means a fault is detected; when structured residuals are obtained, several residuals are not zero at the same time and this allows fault localisation. This is clearly a purely theoretical approach because measurements are noisy, models are imprecise and thus residuals are never zero. Few authors have emphasised the decision problem following residual generation. Most of the time, replacing the zero by a fixed value is proposed, which corresponds to a vague concept and is difficult to choose. Using crisp values as thresholds results in abrupt decisions and eventually unstable decisions when small variations due to noise occur around the limit. The use of fuzzy sets to represent residuals consequently begins to be considered necessary to obtain gradual decisions. This does not solve problems due to noise because fuzzifying a noisy value leads to a noisy membership value. This is why in this paper it has been proposed that decisions be made based on the residual values over a temporal window. Moreover, this paper proposes modelling the uncertainty of the decision. This allows a supervision-system designer to evaluate the alarm-filtering efficiency, and makes it possible to give the operator robust or sensitive decisions following the case study. This approach is very general and could be applied to any kind of model based diagnosis system.

Fuzzy rule bases have been shown to be easily expressed as matrix algebra, modelling a mapping between input descriptors and conclusion descriptors. This allows a computationally efficient real-time implementation of the detection and localisation algorithms.

Communication between the computer and the operators is accomplished owing to several interfaces that illustrate each of the underlying concepts. Colours codes have been widely used in human-machine interfaces to represent qualitative numerical results. Using colours enables concepts that gradually evolve with time to be represented. A graduated colour code translates the detection and isolation algorithm results into a mass visualisation of the variable states, which allows operators to analyse a large amount of qualitative information. A causal graph with coloured suspected nodes or arcs allows the visualisation and physical understanding of the isolation algorithm results.

Previous experiments with novice operators demonstrate the validity of the approach. The use of a model as a reference of normal behaviour was very easily understood. The colour code was easily interpreted and allows operators to anticipate their actions compared to a classical binary alarm signal. The causal graph is a powerful tool for those problems that are not easily managed by novice operators such as faults in regulation loops.

An analysis of a hierarchical representation of causal graphs has been undertaken in order to allow the representation of a complete facility.

Acknowledgements

This work has been supported by the *Direction de la Recherche et de la Technologie de la Division Générale de l'Armement (DRET)* — grant 93-098. The authors are grateful to the kind co-operation of the students and teachers at *Lycée Jean-Baptiste Dumas* (Alès, France) during the evaluation described in this work. The authors are also grateful to important comments and suggestions of the reviewers to improve the paper.

This work has been done when Alexandre Evsukoff was Ph.D. student in *Laboratoire d'Automatique de Grenoble*, supported by the Brazilian National Research Council CNPq, under the grant 200482/95-5.

Appendix A

Detection algorithm applied to all variables at time t :

1. Data acquisition gives $y(t)$; computation of the reference model value gives $y^*(t)$. The total residual is computed as

$$\delta(t) = y(t) - y^*(t).$$

See Fig. 7 as an illustration.

2. Updating of the residual features:

$$x_i(t) = \delta(t - i), \quad i = 0, \dots, \tau_1 - 1,$$

$$\dot{x}_j(t) = \delta(t) - \delta(t - j), \quad j = 1, \dots, \tau_2.$$

3. Fuzzification of the residual features (fuzzy partitions defined by Eq. (14)) leading to fuzzy descriptions of residual features:

$$\tilde{X}_i = [NN(x_i) \dots PP(x_i)], \quad \tilde{X}_i \in I^5,$$

$$\tilde{X}_j = [N(\dot{x}_j) \dots P(\dot{x}_j)], \quad \tilde{X}_j \in I^3.$$

4. Computation of the fuzzy description matrix $\tilde{\mathbf{W}}$, whose components are residual features descriptions in the time window $M = \tau_1 \tau_2$. Each row of $\tilde{\mathbf{W}}$ is computed by product of the combinations of the feature membership values and:

$$\tilde{\mathbf{W}} = \begin{bmatrix} (NN(x_0) \cdot N(\dot{x}_1)) & \cdots & (PP(x_0) \cdot P(\dot{x}_{\tau_2})) \\ \vdots & \ddots & \vdots \\ (NN(x_{\tau_1} - 1) \cdot N(\dot{x}_1)) & \cdots & (PP(x_{\tau_1 - 1}) \cdot P(\dot{x}_{\tau_2})) \end{bmatrix},$$

$$\tilde{\mathbf{W}} \in I^{M \times 15}.$$

5. Computation of partial conclusions about variable state, represented by the matrix $\tilde{\mathbf{S}} \in I^{M \times 2}$, in one pass of $\tilde{\mathbf{W}}$ through the rule base matrix $\tilde{\mathbf{\Phi}} \in I^{15 \times 2}$:

$$\tilde{\mathbf{S}} = \tilde{\mathbf{W}} \circ \tilde{\mathbf{\Phi}} = \tilde{\mathbf{W}} \cdot \tilde{\mathbf{\Phi}}.$$

6. Aggregation of all partial conclusions to compute robust and sensitive final decisions,

$$D^-(FA) = \min(S_1(FA) \dots S_M(FA)),$$

$$D^-(OK) = \max(S_1(OK) \dots S_M(OK)),$$

$$D^+(FA) = \max(S_1(FA) \dots S_M(FA)),$$

$$D^+(OK) = \min(S_1(OK) \dots S_M(OK)).$$

7. Computation of the crisp decisions:

If $D^-(FA) - D^-(OK) > \varepsilon_1$ then an alarm is triggered.

If $D^+(OK) - D^+(FA) > \varepsilon_2$ then the alarm is turned off.

These decisions are transmitted to the classical interface (Fig. 15 is an illustration) and to the causal graph interface to colour the contour of the node in red (Fig. 16 is an illustration).

8. Computation of the colour index associated to the mean decision:

$$\tilde{D} = \frac{1}{2}(\tilde{D}^+ + \tilde{D}^-)$$

$$\varphi_d(t) = D(OK)\varphi_{OK} + D(FA)\varphi_{FA},$$

where $\varphi_d(t)$ is the colour code to be associated to the variable in the detection graphical interface; φ_{OK} is the colour index associated to the colour green and φ_{FA} is the index associated to the colour red in the detection colour palette (see Figs. 17–19 as illustrations).

Isolation algorithm applied to all variables at time t :

1. Data acquisition gives measures $y(t)$; computation of the reference model value gives $y^*(t)$; the local causal model using measures of antecedents as inputs gives $\hat{y}(t)$; the local causal model using reference model values of antecedents as inputs gives $\hat{y}^*(t)$ (see Eqs. (32)–(34)). The local residual is computed as

$$\Delta\delta(t) = \delta(t) - \hat{\delta}(t) = (y(t) - y^*(t)) - (\hat{y}(t) - \hat{y}^*(t)).$$

2. Updating of the residual features:

$$z_i(t) = \Delta\delta(t - i), \quad i = 0, \dots, \tau_1 - 1,$$

$$\dot{z}_j(t) = \Delta\delta(t) - \Delta\delta(t - j), \quad j = 1, \dots, \tau_2.$$

3. Fuzzification of the residual features (fuzzy partitions defined by Eq. (14)) leading to fuzzy descriptions of residual features:

$$\tilde{Z}_i = [NN(z_i) \dots PP(z_i)], \quad \tilde{Z}_i \in I^5,$$

$$\tilde{Z}_j = [N(\dot{z}_j) \dots P(\dot{z}_j)], \quad \tilde{Z}_j \in I^3.$$

9. Computation of the fuzzy description matrix $\tilde{\mathbf{W}}$, whose components are residual features descriptions in the time window $M = \tau_1\tau_2$. Each row of $\tilde{\mathbf{W}}$ is computed by product of the combinations of the feature membership values and:

$$\tilde{\mathbf{W}} = \begin{bmatrix} (NN(z_0) \cdot N(\dot{z}_1)) & \cdots & (PP(z_0) \cdot P(\dot{z}_{\tau_2})) \\ \vdots & \ddots & \vdots \\ (NN(z_{\tau_1} - 1) \cdot N(\dot{z}_1)) & \cdots & (PP(z_{\tau_1 - 1}) \cdot P(\dot{z}_{\tau_2})) \end{bmatrix},$$

$$\tilde{\mathbf{W}} \in I^{M \times 15}.$$

4. Computation of partial conclusions about variable state, represented by the matrix $\tilde{\mathbf{F}} \in I^{M \times 2}$ in one pass of $\tilde{\mathbf{W}}$ through the rule base matrix $\tilde{\mathbf{\Phi}} \in I^{15 \times 2}$:

$$\tilde{\mathbf{F}} = \tilde{\mathbf{W}} \circ \tilde{\mathbf{\Phi}} = \tilde{\mathbf{W}} \cdot \tilde{\mathbf{\Phi}}.$$

5. Aggregation of all partial conclusions to compute robust and sensitive final decisions:

$$L^-(LO) = \min(F_1(LO) \dots F_M(LO)) \cdot D(FA),$$

$$L^-(UP) = \max(F_1(UP) \dots F_M(UP)) \cdot D(FA),$$

$$L^+(LO) = \max(F_1(LO) \dots F_M(LO)) \cdot D(FA),$$

$$L^+(UP) = \min(F_1(UP) \dots F_M(UP)) \cdot D(FA).$$

6. Computation of the crisp decisions:

If $L^-(LO) - L^-(UP) > \varepsilon_1$ then the variable is declared the cause of the fault.

If $L^+(UP) - L^+(LO) > \varepsilon_2$ then the variable is declared a consequence of the fault.

These decisions are transmitted to the classical interface to highlight the contour of the variable name if the variable is declared the source fault (Fig. 15 is an illustration). In the causal graph interface the crisp decisions are used to colour the node in red when the variable is declared the source fault (Fig. 16 is an illustration).

7. Computation of the colour index associated to the mean decision:

$$\tilde{L} = \frac{1}{2}(\tilde{L}^+ + \tilde{L}^-), \quad \varphi_i(t) = L(UP) \cdot \varphi_{UP} + L(LO) \cdot \varphi_{LO},$$

where $\varphi_i(t)$ is the colour code to be associated to the variable in the isolation graphical interface; φ_{UP} is the colour index associated to the colour blue and φ_{LO} is the index associated to the colour red in the isolation colour palette (see Figs. 17–19 as illustrations).

References

- Al Khani, A., & Koivo, H. (1997). Safety and management of complex process: supervision, control, optimisation and intelligent process data management. *Proceedings of the IFAC symposium on fault detection, supervision and safety for technical process — SAFE-PROCESS'97*.
- Babuska, R., & Verbruggen, H. B. (1996). An overview of fuzzy modelling for control. *Control Engineering Practice*, 4(11), 1593–1606.
- Cheung, J. T.-Y., & Stephanopoulos, G. (1990). Representation of process trends — part I. A formal representation framework. *Computers Chemical Engineering*, 14(4/5), 495–510.
- Denoeux, T. (1997). Analysis of evidence-theoretic decision rules for pattern classification. *Pattern Recognition*, 30(7), 1095–1107.
- Dubois, D., & Prade, H. (1990). *Fuzzy sets and systems*. New York: Academic Press.
- Dubois, D., Prade, H., & Yager, R. R. (1996). Information engineering and fuzzy logic. *Proceedings of the international conference on fuzzy systems — FUZZ IEEE'96*.
- Evsukoff, A., Montmain, J., & Gentil, S. (1997). Dynamic model and causal knowledge-based fault detection and isolation. *Proceedings of the IFAC symposium on fault detection, supervision and safety for technical process — SAFEPROCESS'97*.
- Evsukoff, A., Montmain, J., & Gentil, S. (1998). Causal model based supervising and training. *Proceedings of the IFAC workshop on supervision of chemical industries*.
- Foulloy, L., & Galichet, S. (1995). Typology of fuzzy controllers. In H. Nguyen, M. Sugeno, R. Tong, & R. Yager, *Theoretical aspects of fuzzy control*. New York: Wiley.
- Frank, P. (1996). Analytical and qualitative model-based fault diagnosis, a survey and some new results. *European Journal of Control*, 1(2), 6–28.
- Frélicot, C., Masson, M. H., & Dubuisson, B. (1995). Reject option in fuzzy classification rules. *Proceedings of EUFIT'95*, 3, 1459–1464.
- Grabish, M. (1995). On equivalence of fuzzy connectives — the case of fuzzy integrals. *IEEE Transactions on Fuzzy Systems*, 3(1), 96–109.
- Iserman, R. (1998). On fuzzy logic applications for automatic control, supervision and fault diagnosis. *IEEE Transactions on Systems, Man and Cybernetics — Part A: Systems and Humans*, 28(2), 221–234.
- Iserman, R., & Ballé, P. (1997). Trends in the application of model-based fault detection and diagnosis of technical processes. *Control Engineering Practice*, 5(5), 709–719.
- Johannsen, G. (1997). Conceptual design of multi-human machine interfaces. *Control Engineering Practice*, 5(3), 349–361.
- Kosko, B. (1992). *Neural networks and fuzzy systems*. Englewood Cliffs, NJ: Prentice-Hall.
- Kramer, M. A., & Palowitch, B. R. (1987). A rule based approach to fault diagnosis using the signed directed graph. *AIChE Journal*, 33(7), 1067–1078.
- Lambert, M., Riera, B., & Martel, G. (1997). Conception d'une interface de nouvelle génération pour la supervision d'un procédé de retraitement de combustibles nucléaires. *6^e Congrès Français de Génie des Procédés*.
- Leitch, R., & Gallanti, M. (1992). Task classification for knowledge-based systems in industrial automation. *IEEE Transactions on Systems Man and Cybernetics*, 22(1), 142–152.
- Leyval, L., Gentil, S., & Feray-Beaumont, S. (1994). Model based causal reasoning for process supervision. *Automatica*, 30(8), 1295–1306.
- Leyval, L., Montmain, J., & Gentil, S. (1994). Qualitative analysis for decision making in supervision of industrial continuous processes. *Mathematics and Computers in Simulation*, 36, 149–163.
- Rasmussen, J. (1993). Diagnostic reasoning in action. *IEEE Transactions on Systems Man and Cybernetics*, 23(4), 981–991.
- Rengaswamy, R., & Venkatasubramanian, V. (1995). A syntactic pattern-recognition approach for process monitoring and fault diagnosis. *Engineering Applications of Artificial Intelligence*, 8(1), 35–51.
- Theilliol, D., Weber, P., Ghetie, M., & Noura, H. (1995). A hierarchical fault diagnosis method using a decision support system applied to a chemical plant. *IEEE international conference on systems, man, and cybernetics*.
- Yager, R. R. (1988). On ordered weighted averaging operators in multi-criteria decision making. *IEEE Transactions on Systems Man and Cybernetics*, 18(2), 183–190.
- Yager, R. R. (1993). Families of OWA operators. *Fuzzy Sets and Systems*, 59, 125–148.
- Zadeh, L. (1996). Fuzzy logic computing with words. *IEEE Transactions on Fuzzy Systems*, 4(2), 103–111.
- Zimmermann, H.-J. (1996). *Fuzzy set theory and its applications*. Dordrecht: Kluwer Academic Publishers.