Proceeding of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space: i-SAIRAS 2003, NARA, Japan, May 19-23, 2003

# Fuzzy inductive reasoning and possibilistic logic for space systems failure smart detection and identification

A.Ercoli Finzi [1] - M.R.Lavagna [1]– G.Sangiovanni
[1]Politecnico di Milano - Dipartimento di Ingegneria Aerospaziale
Via La Masa 34, 20158 Milano, Italia
lavagna@aero.polimi.it

**Keywords:** Autonomous diagnosis, failure detection, failure identification, possibilistic logic, inductive reasoning

**Abstract**

The increased complexity of space systems implied, in recent years, a growing demand for an on-board *artificial intelligence*, including fault-monitoring systems more and more autonomous, aimed both to enhance the mission performance and to support technicians during the plant operative life.
The paper proposed a mixed approach to deal with possible symptoms *Detection* from sensor readings and –consequently- failure Identification in terms of either occurred or incipient fault of a particular device, driven by a qualitative knowledge base of the system to be monitored.
The *Detection* mechanism is based on the *Inductive reasoning* approach, supported by the *Fuzzy Logic* theory to deal with uncertainties intrinsic in data coming from sensor readings. The *Identification* is accomplished by applying algorithms from the *Possibilistic Reasoning* theoretic field. Uncertainties related both to the classification of telemetry data as failure symptoms and to the inductive relationships from the current symptoms pattern to the actual faulty scenario are completely taken into account. Moreover, false alarms can be avoided, as fixed threshold for sensor reading deviation are present: a dynamic threshold for symptoms decree is created systematically, according to the current available I/O time series data.
Results presented are related to a real space scenario: the GOCE spacecraft – an European Space Agency project carried on by Alenia Spazio in Turin – has been taken as a test bed.
In particular, Failure Detection & Identification of real failures related to the Electric Power Subsystem are successfully managed. The Smart-FDIR here proposed easily detects and rapidly identifies failures really injected in the system thanks to the EPS simulator implemented in Alenia Spazio.

## 1. Introduction

The increased complexity of engineering systems implied, in recent years, a growing demand for fault monitoring systems more and more autonomous, aimed to support technicians during the plant operative life. At least, on-line fault detection and identification performance are required. Autonomous failure recovery performance is a benefit to the Failure
Monitoring Systems, currently under study, especially for those systems that operates in hostile environment such as space systems. Within the space system domain, autonomous Failure Detection Identification &Recovery (FDIR) performance, entails the possibility to make feasible space missions far from the Earth, as the lack of timely radio-contacts - because of the huge relative distance (O(10^1 AU)) - to support abnormal system behaviors is completely overcome. Moreover, any kind of space mission could be better exploited; increased product return is strictly related to soft failure tolerance by keeping the system active according to the current failure, instead of switching off any activities but those related to safeness maintenance.
Currently, there is an intensive research activity in the FDI area, the FDIR being more difficult to be implemented.
In particular, within the space domain, the most relevant example – as it has been successfully tested in-flight- is the Mode Identification & Reconfiguration module, included in the Remote Agent implemented for the NASA DS1 mission-1998 [1], [2].
The approach there assumed greatly differs from the usual FDIR management, based on FMECA off-line analysis and telemetry monitoring with the central role of the ground operators. Instead of working with a set of pre-fixed rules, it uses a behavioral model of the system that assure a better actual system state monitoring and reconfiguration.
The current paper, starting from the available literature proposes a mixed approach to enhance current FDI tools within the space applications, and to increase their autonomy level.
A model-based approach has been selected to better answer requisites of false alarm containment, abrupt and incipient fault mode distinction and eventually, best tailored reconfiguration.
The detection mechanism is based on the computation and judgment of the residual between sensor readings and expected outputs from the nominal mode model.
To identify the nominal mode model to refer to, a qualitative approach is here proposed applied to I/O time series.
Even if quantitative modeling seems to be, at a first glance, the obvious and most rigorous approach, it has to be reminded that it entails a complete knowledge of analytical dependencies among involved variables. Whenever plants are highly complex (as a spacecraft is), involving components related to different disciplines domains, that analytical knowledge could not be so obvious.
Moreover, to deal with that kind of uncertainty humans are often present in the loop, coping with their expertise the lack of analytical knowledge.
That is why qualitative modeling could be a valid tool to deal with both system model uncertainty, and processes that involve, somehow, the human brain causal mechanism simulation. They work on a symbolic knowledge of the plant to be modeled. As opposite to quantitative methods, qualitative models make use of heuristic knowledge instead of differential equations. Linguistic variables specify the input and output signals using linguistic terms. This enhances the robustness of the model versus unknown or time-dependent parameters of the system. As in the quantitative approach, an observer is defined to generate the residual vector from actual

measurements. The different stays in the observer identification. Successfully applied observer identification can be accomplished by applying different algorithms, based on the Fuzzy Logic Theory.

In particular, according to the modellization issue, within the current work the approach based on the Fuzzy Inductive Reasoning turned out to be the best tailored.

Within the FIR approach, three different methodologies have been implemented to deal with the qualitative I/O dependence settlement, specifically the Markov chain single dependency, the Shannon's minimum entropy, and the Mean Square Error approaches. The FIR has been hence applied to compute the residual for the possible fault detection.

In particular, detection is managed by the *envelope* computation, as proposed by [3],[4], [5]. The *envelope* approach allows detecting both abrupt and incipient fault modes, depending on the cumulative error time window tuning.

Whenever a fault mode is detected, the *Identification* mechanism should be activated; identification is a quite complex component to be implemented in the framework of an autonomous diagnosis system.

The highest difficulty stays in the management of complexity that rises from uncertainty intrinsic in the process.

Two levels of uncertainty can be identified: the first one related to the behavioral knowledge representation. The behavioral knowledge is the causal dependency among inputs and outputs normally expressed into a declarative language. Within the KB not causal dependencies can be represented as implications, hence some faults may be present both if a symptom is present and absent. Hence, starting from observations, nothing can be surely said about faults not strictly related to (i.e. that do not entail) the current symptoms vector.

The second level of uncertainty deals with limited observable quantities. The aforementioned complexity levels are normally coped by human expertise in component plant behaviors and operational modes. That expertise allows recognizing particular observation patterns to rapidly rank possible failure scenarios, detecting unknown modes and record them for future identification, to admit multiple fault scenarios and uncertain fault situations. Those performances assure the human-driven FDI to be timely, flexible and safe respectively.

Hence, to automate the identification process, those requisites should be met.

Assuming a model-based framework, a structural (i.e. system components) and behavioral models have been settled. The behavioral models represent the system knowledge about causal dependencies between inputs and outputs; they are here represented by a logical formulation.

Logical formulation can be easily implemented by state variable representation, good for a re-utilization of the base of knowledge for autonomous reconfiguration performance. Within the logical model, also faulty configurations are taken into account in order to be able to manage the identification by applying both a consistency-based and an abductive algorithm. In particular, failures and nominal mode are logically modeled by the state variable attributes and the degree of possible dependence of state variable attributes coming from different modeled devices.

The identification module works on that behavioral knowledge logical model: starting from observations, by applying the *Possibilistic* Logic mechanisms, it identifies possible causes (the unknown included), ranked by their degree of necessity to be true. In the followings, the two mechanisms are presented and some significant results are given and commented.

## 1. The Detection Mechanism

The *Detection* is aimed to recognize that the plant behavior is abnormal, and therefore that the system is not working properly. In other words, its main task stays in classifying observations as symptoms.

In order to implement an autonomous detection module, some knowledge of the system to be monitored must be modeled. More specifically, nominal and abnormal modes should be, somehow, either represented or recognizable. In particular, both a nominal mode model should be identified off-line, and an actual system mode detected on-line.

According to the simplest FMECA, a sort of table can be codified to represent both nominal and failed behavior.

That kind of approach, however, does not admit any deviation from a static system representation, and it is normally highly conservative: no reasoning on the detected deviations is considered, they are directly codified as a dangerous plant status and all activities are interrupted to keep the system in a safe mode. However, it has to be considered that, whatever model has been applied to represent the system dynamics, either measurements or model errors can occur that do not entails, always, a fault occurrence. In other words, the domain uncertainty could lead to false alarms, hence to system goals degradation as the plant is not exploited at the best. Detected deviations between expected and sensor readings can also happen either suddenly or smoothly, corresponding to an abrupt or an incipient failure respectively. Incipient failures are obviously more difficult to be detected as they imply time series analysis and evaluation, usually of quantities with a time domain larger than Boolean.

Moreover, to distinguish between deviations that could imply either soft or hard faults would represent an enhancement with respect to a simple Boolean detection faulty-nominal status. Soft failures represent those failures that can be partially admitted by the system while accomplishing its functionalities; whenever hard failures occur, the system can restore no functionality but safely surviving.

According to the autonomy requisite, the detection module, in particular, should simultaneously enhance current FMECA performance and simulate operators' expertise in monitoring systems. In particular, the former highlighted performance can be translated into requisites of robustness, sensitivity and flexibility. Hence, the first crucial point for robust deviation detection can be translated into a good plant dynamics and functional modellization. A flexible and sensitive criterion settlement to judge detected deviations follows as the second step of the intelligent module implementation. As already pointed out the detection mechanism here proposed is based on the FIR approach as, as it will soon clearly appears, it answers all aforementioned requisites.

More specifically, both the system modellization and the system behavior forecast are accomplished by the FIR approach, while the expected-actual output values discrepancy is managed by the *envelope* technique. As a first step, the modellization process is presented, followed by the on-line symptoms decree process.

### 1.1 The Modellization Technique

As the FIR technique is well known and literature is available on that topic, short guidelines related to the basis of this

technique are here given; for further details the reader is addressed to the related papers [3], [4], [5],[6], [7], [8].

The FIR is a qualitative modeling and a simulation methodology, belonging to the General System Problem Solver. Its main task stays firstly in identifying qualitative causal rules between either observed or input/output variables, by applying Inductive Reasoning techniques In Inductive reasoning, conclusions are inferred from premises. FIR just attaches to the inference a degree of truth represented by the membership value of the fuzzy mapping. Fuzzy Logic is a powerful tool whenever analytic models are not available, but qualitative dependencies can be highlighted between system variables. Moreover, it allows converting those qualitative relationships into mathematical formulation good for computer manipulation.

The overall architecture of a FIR algorithm comprehends four modules: the fuzzyfication module, the qualitative modeling engine (QME), the qualitative simulation engine (QSE), the defuzzyfication module. Fig. 2and

Fig. 3 show the overall architectures for the modellization and forecast processes respectively. According to the modellization process, the FIR uses a sort of data mining to build the set of qualitative rules that model the system. By referring to Fig. 2, the first module just maps each quantitative data point into a qualitative triple, namely class (previously selected, such as low, medium, high), membership degree (usually Gaussian memberships are selected) and side (left-center-right according to the qualitative class they belong to).

The QME tries to discover the behavioral pattern among the observations by manipulating their correspondent qualitative triples; in other worlds, it builds the inferential motor connecting input triples to output triples starting from the sensor data records. In particular, it works on a matrix of temporal recorded inputs/outputs to encode a matrix, called *mask* the elements of which represent the rules between inputs and output. Just to clarify, given for example, a four inputs single output system, having recorded input/outputs data from instant $t_{i-k}$ to $t_i$ a qualitative dependency is aimed, to simulate the system behavior, for example:

$$y_1(t)=f(u_3(t_i-t_{i-2}), u_1(t_i-t_{i-1}), u_4(t_i-t_{i-1}), u_1(t_i), y_1(t_i-t_{i-1})) \quad \text{eq. 1}$$

The former has not be read as a classic analytical function; it just highlights – with a qualitative perspective – which quantities, in which instant are determinant for the current detected output value. That qualitative dependence is encoded in the mask:

$$\text{mask} = \begin{array}{c} t/x \\ t_{i-2} \\ t_{i-1} \\ t_i \end{array} \begin{array}{c} u_1 \quad u_2 \quad u_3 \quad u_4 \quad y_1 \\ \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & -1 & -1 \\ -1 & 0 & 0 & 0 & +1 \end{bmatrix} \end{array} \quad \text{eq. 2}$$

where:

$u_i$ = Input (-)
$y_i$ = Output (+)
$t_i$ = Current instant
$t_{i-n}$ = Previous observed instants

Within the mask, the '$m_{kj}$=-1' means a dependence exists between input $u_j$ at time $t_{k-i}$ and the output y at time $t_i$, while the '$m_{kj}$=0' assure no relationship exists. The modellization needs, necessarily the definition of the optimum mask.

The search of the optimum mask is accomplished either by an exhaustive search of the candidate masks or by heuristics applied to the search space. A possible heuristic is based on Markov chains of single dependency. All maxima of the likelihood function versus Markov chain order are assumed present in the mask [5].

Within heuristics, again based on a probabilistic approach, it can be found the minimum Shannon Entropy criterion [3]. A new successful approach is here proposed based on enlarging the mask element domain to real numbers instead of limiting them to the integer domain. The mask element settlement is supported by an *ad hoc* supervised Artificial Neural Network; the weights of that net represent the mask elements and they are tuned according to a given I/O pattern to be matched [9]. In particular, the gradient method has been revised by applying the following recursive equation to compute the mask elements:

| Sampling instant (sec) | Expected by the "reference model" (Watts) | Read from the sensors (simulator outputs) (Watts) | Envelope Min | Envelope Max |
|---|---|---|---|---|
| 1992 | 426.075 | 426.070 | 425.298 | 426.855 |
| 2002 | 426.039 | **408.718** | **425.266** | **426.821** |
| 2012 | 426.005 | **408.686** | **425.236** | **426.786** |
| 2022 | 425.974 | **408.654** | **425.205** | **426.752** |
| 2032 | 425.935 | **408.625** | **425.173** | **426.687** |

$$w_i^{t+1} = w_i^t - \eta \cdot f_{att} \cdot \frac{\frac{\partial E}{\partial w_i^t}}{\max \frac{\partial E}{\partial w_i^t}} \quad \text{eq. 3}$$

$$f_{att}(iter, N_{it}) = \frac{\log(1+\frac{N_{it}-iter}{iter-1+\varepsilon})}{\log(K+\frac{N_{it}-iter}{iter-1+\varepsilon})}$$

K>1 and $\varepsilon=10^{-14}$

With:

| | |
|---|---|
| $w_i^t$ | Element to be filled into the mask matrix (w=[0 1]) |
| $\eta$ | Learning rate |
| E | MSE between estimated and real output |
| iter | Current iteration number |
| $N_{it}$ | Number of iterations |

The presence/absence of a particular sensor reading in eq.1 is no more Boolean, and the (nxm-1) degrees of freedom give rise to $\infty^{(nxm-1)}$ masks for the system model identification.

Apart from a better off-line system modellization, the most important benefit stays in that the system acquired learning performance to be easily accomplished during the system operative mode. The base of knowledge nested in the mask, in fact, can be redefined whenever required, as soon as new I/O patterns became available. Apart from the method to compute it, the mask is the fundamental tool to build the inferential motor of the fuzzy system model: by shifting the selected mask over the recorded data triple, static relationships between inputs and outputs can be obtained in terms of class, membership degree and side. Static relations represent the

base of knowledge to work on to recognize patterns during the dynamics and to forecast the outputs. In Fig. 1, according to the mask in eq.2, the behavior matrix is given:

| Sampling instant (sec) | Expected by the "reference model" (Watts) | Read from the sensors (simulator outputs) (Watts) | Envelope Min | Envelope Max |
|---|---|---|---|---|
| 1992 | 426.075 | 426.070 | 425.298 | 426.855 |
| 2002 | 426.039 | **408.718** | **425.266** | **426.821** |
| 2012 | 426.005 | **408.686** | **425.236** | **426.786** |
| 2022 | 425.974 | **408.654** | **425.205** | **426.752** |
| 2032 | 425.935 | **408.625** | **425.173** | **426.687** |

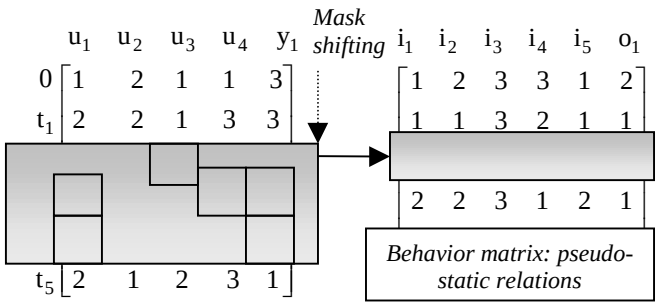



Fig. 1 Mask shifting on class dynamic matrix and static behavior matrix for the class qualitative information

Where:

$i_i$ = Input at the predefined instants
$o_i$ = Output at the current observed instant

As shown in Fig. 1, the first row can be read as:

***If*** input $i_1$ belongs to class 1 (e.g. 'low')
***and*** " $i_2$ " " 2 (e.g. 'medium')
…… …… ……. …
***The*** output
***n*** $o_1$ belongs to class 2 (e.g. 'quite small')

Particular attention should be paid on three main aspects: the I/O pattern selected to build the mask, the mask depth and the I/O domain mapping with qualitative memberships.
The I/O pattern should answer requisites of completeness and minimality. Completeness is obviously required to assure model identification as most accurate as possible. Minimality feature comes from the on-line running of the mechanism: the larger the data sets are the slower the forecast will be.
Particular attention must be paid, hence, in selecting both the time span for I/O data collection and the frequency for data sample collection. The time gap between two consequent samples has been, here, defined by matching the I/O expected and read series in the state space, looking for a well-distributed coverage. Whenever a periodic trend is expected within any monitored quantity, the time window dedicated to the time series data collection should be at least equal to the period. The mask depth - basically its row number- represents the model time dependence: the larger it is the more integral performance are added to the model, the lower the more time history independent the model is. According to the I/O domain qualitative mapping, both the number and type of membership functions must be settled. As for every Fuzzy Logic based algorithm, a large membership number increases the model complexity, giving, on the other hand, great flexibility. By containing that number the model will be less sensitive but more robust according to modeling errors. Both rule of thumb and optimization techniques can be applied to tune these two last parameters. According to the current work, they have been settled heuristically.

### 1.2 Forecast and Detection Techniques

Once the static causal model is created (see Fig. 1), the qualitative simulation engine can run to forecast system outputs starting from the current input set. As shown in
Fig. 3, input sets are fuzzified and the correspondent class, side and membership matrices are computed. Input set is defined by shifting the mask over the dynamics triple of matrices. The behavior matrix is then searched to identify similar input patterns within the base of causal knowledge. Similarity is computed according to the position quantity defined as:

$$\text{position}_{\text{input},i} = C_i + S_i*(1-MF_i) \qquad \text{eq. 4}$$

where:

$C_i$ = Class of current input
$S_i$ = Side of current input
$MF_i$ = Membership of current input

A weighted Euclidean distance of the current input vector according to the position of each input vector in the pseudo-static matrices is evaluated and the output of the nearest relationship is assumed as the forecast, according to the side S and the class C; the output membership value is evaluated as a weighted sum of the k nearest pseudo-static relationships. Eq.5 gives the different formulation for the forecast computation, with respect to the FIR technique proposed in [8].

$$d_j = \sqrt{\sum_{i=1}^{N} w_i (\text{position}_i - \text{position}_{ij}^{PS})^2} \qquad \text{eq. 5}$$

N=no.of inputs j= pseudo-static relationship counter

As explained, the $\underline{w}$ vector comes from the mask element optimization by the particular ANN application.
Having defined the output triple (C, S, MF) the crisp output is obtained by a simple defuzzification.
To accomplish failure detection tasks, the forecast outputs are compared with the actual ones, and an instantaneous error is obtained. By defining a time window during which errors are collected, an average error derives from a simple instantaneous error sum over the time window [6].
The average error is fundamental for failure detection not to be trapped in either fault alarms or instantaneous, but irrelevant variable deviations. To this end, the so-called *envelope* approach is proposed [4]. The idea is to compute, for each forecast, an interval of acceptability of the real trajectory value, called *envelope*.
The range boundaries the prediction can vary into are represented by the lowest and the highest prediction values within the k-patterns considered before. In that way a time-varying interval of forecasting acceptability is obtained and

robustness according to modeling errors is acquired. Moreover, whenever the measured output falls out the *envelope* boundaries, the instantaneous error vector element is triggered to one. The average error vector, in its turn, having integral performance, is used to distinguish between incidental, false alarms and persistent deviations, as a filter: an instantaneous memory size and a threshold must be settled for the filter: the lower the size is, the more timely and less robust the mechanism is; the lower the threshold is the more sensitive, and the less robust the fault detection is; conversely, the higher the threshold the more robust and less sensitive the detection.
Whenever the average error – within the memory size window - overcomes the threshold, the detected deviation represents a symptom and the identification component is called.
The selected *envelope* technique on which the whole detection module is based, prevents to be trapped in a model affected either by not considered or expected errors: whenever a model has been identified, the forecast phase, the core of the alarm activation, does not compare and judge directly expected data from the FIR model with sensor readings, but a sort of 'safety zone' is built around the expected output computed by the model: the rougher the identified model is the larger that zone is to take into account for modeling errors, fault alarm events and whatever can lead to erroneous evaluations. It is important to notice that the 'safety zone', called the *envelope* is not prefixed by a user but it strictly depends on the distance between the I/O static patterns defined during the model identification and the current I/O data set.
Hence, the possible loss of accuracy in identifying the "reference model" is stemmed by providing the expected outputs with such a modeling uncertainty range used to decree whether the current sensor readings errors come either from a possible faulty scenario or from modeling inaccuracy.

## 2. The Identification technique

The identification mechanism is aimed to conclude which type of failure has occurred, that is to say to map symptoms into causes. Identification is a quite complex component to be implemented in the framework of an autonomous diagnosis system. The main issue of an autonomous identification module stays in the observations-into-failures on line mapping. The highest difficulty stays in the management of complexity that rises from uncertainty intrinsic in the process.
Two levels of uncertainty can be identified: the first one related to the behavioral knowledge representation. The behavioral knowledge is the causal dependency among inputs and outputs normally expressed into a declarative language. Within the KB not causal dependencies can be represented as implications, hence some faults may be present either if a symptom is present or absent. Hence, starting from observations, nothing can be surely said about faults not strictly related to (i.e. that do not entail) the current symptoms vector.
The second level of uncertainty deals with limited observable quantities. The aforementioned complexity levels are normally coped by human expertise in component plant behaviors and operational modes. That expertise allows recognizing particular observation patterns to rapidly rank possible failure scenarios, detecting unknown modes and record them for future identification, to admit multiple fault scenarios and uncertain fault situations. Those performances assure the human-driven FDI to be timely, flexible and safe respectively.
Hence, to automate the identification process, the same requisites should be satisfied. As already mentioned, the current paper proposes a Possibilistic Logic approach to deal with the uncertainty multiple levels, as explained in [10].
A behavioral knowledge of the system is necessary to build the system abductive models [11]; abduction enables the identification mechanism to restrict further the domain of possible failures – given a symptom scenario – with respect to a simple consistency based modeling.
The benefit of working with logical instead of redundancy models stays in the possibility of managing autonomous recovery performance by implementing no further models but by working on the same logical ones. The Possibilistic Logic handles causal propositions of the first-order to which are attached numbers between 0 and 1. These weights are lower bounds on so-called degrees of necessity or degrees of possibility of the corresponding formulas. The degree of necessity (or certainty) of a formula expresses to what extent the available evidence entails the truth of this formula. The degree of possibility expresses to what extent the truth of the formula is not incompatible with the available evidence.
At the mathematical level, degrees of possibility and necessity are closely related to fuzzy sets, and possibilistic reasoning is especially adapted to automated reasoning when the available information is pervaded with vagueness. [10], [12], [13]. Given a proposition p, a single membership is activated related to its degree of possibility to be true:

$\Pi(p)=\mu(p)$
$\mu(p)=1$ complete uncertainty on p as it is definitely possible
$\mu(p)=0$ it defines the certainty of $\neg p$
$\mu(p)=1 \mid \neq \mu(\neg p)=0$ the complete uncertainty on p does not entails certainty on $\neg p$

Conversely, a certainty measure is settled by eq.6a: the less the p proposition is possible, the higher the $\neg p$ proposition is certain. Certainty is, in its turn mapped into a fuzzy set by dedicated membership functions (see eq.6b)

$$N(p)=1-\Pi(\neg p) \text{ (a)} \qquad N(p)=\overline{\mu(\neg p)} \text{ (b)} \qquad \text{eq. 6}$$

Uncertainty is hence modeled by the possibility memberships, while reasoning is based on the certainty membership values.
Possibilistic reasoning can be efficiently applied to causal nets related to fault identification, as they allow to deal with intractable scenarios such as having observation sets that may be related to a fault but they are not entailed by the fault or, having no observations that a fault causes but does not entail: these two situations express a sort of vagueness about the abduction the fault happens or not.
Possibilistic theory does not prune causes till the degree of necessity for the correspondent fault absence is equal to one ($N(\neg f)=1$). Hence, multiple causal branches are treated in parallel, and uncertainty may be kept until the final solution.
It has to be noticed that not to be able to identify a failure and hence giving a possible fault rank is a benefit with respect to identify the wrong faulty component.
Within the possibilistic approach, the certainty 'p' of event 'f' does not entail the certainty '1-p' of $\neg f$; in an uncertain domain that means that given the certainty on event f to be p(f), the certainty on its complement 'non-f' is not '1-p(f)'; hence, the more the 'f' event is uncertain, does not entails that the more the 'non-f' is certain. This is obviously a powerful

tool of that logic to correctly deal with reality: if it could rain with 100% level of uncertainty, it does not mean that the level of uncertainty for the 'no-rain' event has to be 0% (hence, it will be surely sunny). The weak point of the approach stays in the behavioral knowledge settlement. To better focus the search, causal relationships are clustered into knowledge islands, according to each considered failure logical representation. The root of the island is the considered failure [10]. Nodes of the KI are intermediate system component states, the last node being symptoms. Nodes are connected by branches that represent the causal net of the behavioral knowledge, each labeled with their degree of certainty, to represent the uncertainty related to the system behavior. Whenever symptom set are input from the detection module uncertainty degree is attached by a fuzzy mapping, to each observed quantity. Identification is based on consistency reasoning to prune firstly the F set of possible faults that are certainly inconsistent with observations; it makes use of abductive technique reasoning to further reduce the $\hat{F}$ set of consistent faults to those relevant with the certain current observations.

The causal net is modeled by an inference fuzzy motor, settled off-line, to define the degree of membership of each symptoms element to the absent or present set of each considered fault.

A second fuzzy inference motor is settled to establish the absence and presence degree of membership of quantities to the symptoms set. By intersecting, within each failure domain, certainly present symptoms fuzzy set from the behavioral model with certainly absent symptoms fuzzy set from the symptoms model, and certainly absent symptoms fuzzy set from the behavioral model with certainly present symptoms fuzzy set from the symptoms model consistency is preserved, and the set of possible faults is restricted.

By intersecting, the current fuzzy set of possible failure either with, intersection set of certainly present symptoms fuzzy set from the behavioral model with certainly present symptoms fuzzy set from the symptoms model, or certainly absent symptoms fuzzy set from the behavioral model with certainly absent symptoms fuzzy set from the symptoms model abduction is obtained, and the set of possible relevant faults is computed. The former operations are mathematically exposed below.

$$\hat{F} = \left\{ f \in F, M(f)^+ \cap M^- = 0 \wedge M(f)^- \cap M^+ = 0 \right\} \text{ consistence}$$

$$\hat{F}^* = \left\{ f \in \hat{F}, M(f)^+ \cap M+ \neq 0 \vee M(f)^- \cap M^- \neq 0 \right\} \text{ abduction} \qquad \text{eq. 7}$$

where:

| | | |
|---|---|---|
| $F$ | = | Considered fault set |
| $\hat{F}$ | = | Consistent possible failure set |
| $\hat{F}^*$ | = | Consistent possible relevant failure set |
| $M(f)^+$ | = | Fuzzy set of symptoms certainly present with failure f |
| $M(f)^-$ | = | Fuzzy set of symptoms certainly absent with failure f |
| $M^+$ | = | Fuzzy set of symptoms currently certainly present |
| $M^-$ | = | Fuzzy set of symptoms currently certainly absent |

## 3. Simulation results

Results obtained by applying the former exposed architecture are now given. The test bed selected for validation is the EPS of the GOCE spacecraft, currently under development at Alenia Spazio- Turin, Italy according to an ESA contract [14].

Alenia Spazio has developed an EPS simulator devoted to generate I/O patterns both under nominal and faulty spacecraft modes. As an example, results related to the injection of one of the failures labeled as "hard to be detected" within the classic FDIR approach, are here analyzed. Actually by injecting each failure mode given in the related documentation the smart –FDIR module here proposed rapidly detect symptoms, without been trapped in false alarm and correctly identify the injected failure within few seconds after the injection occurred. Moreover, incipient failures, simultaneously to other abrupt occurred failures, are easily identified. Fig. 4 shows the model forecast and the real sensor readings for the Solar Array (SA) power supply having injected the loss of one string of Wing1 at 2000s. The reference model has been built by considering time series data coming from a whole orbit period (5370s); sampling is assumed to be available every 10s. Those two parameters have been settled by deeply analyzing the read and expected data matching within their space state domain. The real readings (blue) falls out of the *envelope* given by the model, as soon as the failure is injected, creating an instantaneous deviation between the expected and read values. This confirms the goodness of the FIR technique in identifying the model from time series data; the goodness of the "envelope" approach to classify deviations as symptom is also proved. That failure represents a real interesting case, as the power supply reduction of 1/27 (because of the failure) is definitely hardly detectable with traditional techniques. The signal decreases less than the 4% hence it is difficult to read it as a symptom. and Tab. 1 better highlight the power of the detection module in rapidly classifying the power supplied from the W1 as a possible symptom. Within the current simulation, the size and threshold parameters assumed default values (5 and 3 respectively). At the second sampling the readings already falls out of the envelop range and the instantaneous error starts being cumulated, as depicted in (a)

(b)

**Fig. 6**. As soon as the threshold is reached, at 22s after the failure happening and at the third sensor reading, the alarm is active (red stars), as clarified in Tab. 2 and zoomed in Fig. 6. The alarm keeps being correctly active as long as the sunlight period is on. As the monitored output is the W1 power supply, the eclipse phase makes the failure undetectable. That is why both the signal and the errors come back to a nominal mode (see - Fig. 6around 5000s).

Fig. 7 shows the behavioral model settled for the current failure (SOC=State of charge of the battery). In Fig. 8 necessity index F for three possible EPS failures to be considered as the current cause for the detected symptoms is represented: the higher the index is the more necessary the failure is. The system correctly identifies the current failure (third graph jumps to one at 2000s). Moreover, as a SA string loss will implies less power amount to be devoted to the battery recharge, correctly a possible incipient failure is detected (second graph): the battery voltage level starts lowering and, without any intervention it eventually will reach its lowest admissible limit: the related failure index starts smoothly, but monotonically growing to one (failure happening 'necessary') as soon as the SA string failure occurs.

## 4. Conclusions

The paper proposes a mixed architecture to face autonomous failure diagnosis within the space domain application. By

mixing two different techniques, benefits can be preserved while drawbacks can be limited. A model-based approach has been selected to assure the system a good level of flexibility. Time series reference model identification is applied in order to make the diagnosis system able to redefine itself whenever required by simply readings sensor I/O data. Moreover qualitative instead of quantitative modeling has been preferred to take into account all uncertainty aspects related to both observations and causal relationships. The *envelope* technique, from the FIR domain, revealed to be a robust approach to deal with false alarm avoidance and modeling error filtering. Some tuning parameters should be carefully settled in order to obtain expected results. The connection of the symptom classification with the failure identification module implemented through a possibilistic logic approach gave definitely good results according to required requisites: the smart-FDIR correctly identifies injected failures and gives information on possible interconnected effects on incipient failures risen from the currently identified. The application to a real space case - specifically the GOCE spacecraft – demonstrated that the system could be a good support within the currently applied classic approach, to support FDIR process in complex and uncertain scenarios.

## References


[1] D.Bernard, G.Dorais, C.Fry, E.Gamble, B.Kanefsky, N.Muscettola,P.Nayak, B.Pell,B.Smith, M.Wagner, B.Williams, "Design the Remote Agent Experiment for Spacecraft Autonomy" Proceedings of the IEEE Aerospace Conference,1998

[2] B.Pell D.Bernard, S.Chien, E.Gat, N.Muscettola,P.Nayak, M.Wagner,B.Williams,"An autonomous Spacecraft Agent Prototype", Proceedings of the First International Conference on Autonomous Agents, 1997

[3] A.Nebot, F.Cellier,M.Vallverdu, "Mixed quantitative/qualitative modelling and simulation of the cardiovascular system", Journal of Computer Methods and Programs in Biomedicine, 1997

[4] A.Escobet, A.Nebot, F.Cellier, "Model acceptability measure for the identification of failure in qualitative fault monitoring systems", European Simulation Multi-conference, Warsaw-Poland, 1999

[5] A.Nebot, J.J.Valdes, M.Guiot, R.Alquezar "Fuzzy Inductive Reasoning Approaches to the identification of models of the Central Nervous System Control", ICSC Symposium on Engineering of Intelligent Systems, Tenerife, Spain, February 1998

[6] J.M.Tur, R.M.H.Garrido, "Fuzzy Inductive Reasoning Model-Based Fault Detection Applied to a commercial Aircraft",Simulation journal, vol.75, 2000

[7] D.Li, F.E.Cellier," Fuzzy Measures in inductive reasoning", proceedings of the winter simulation conference, New Orleans, LA, 1990

[8] F.Mugica, A.Nebot,"A specialization of the k-nearest neighbor classification rule for the prediction of dynamic system using FIR" Int. Conf. on system research Informatics &Cybernetics, Baden, Germany,1996

[9] Haykin, "Neural Networks: a comprehensive foundation"N.Y. Mac Millan College publishing Company

[10] D.Cayrac, D.Dubois,H Prade, "Handling uncertainty with possibility theory and fuzzy sets in a satellite fault diagnosis application", IEEE Transactions on Fuzzy Systems, vol.4, no.3 1996

[11] F.Harmelen,A. Teije "Using domain knowledge to select solutions in abductive diagnosis", European Conference on Artificial Intelligence, 1994

[12] D.Dubois, J.Lang, H.Prade, "Possibilistic Logic", Handbook of Logic in Artificial Intelligence and Logic Programming", Gabbay-Hogger-Robinson Ed. vol.3, 1994

[13] D.Dubois, H.Prade,"Fuzzy relation Equations and causal reasoning", Journal of Fuzzy Sets and Systems,pgg.119-134, 1995

[14] AA.VV."GOCE FDIR Concept Technical Note"Alenia 2002

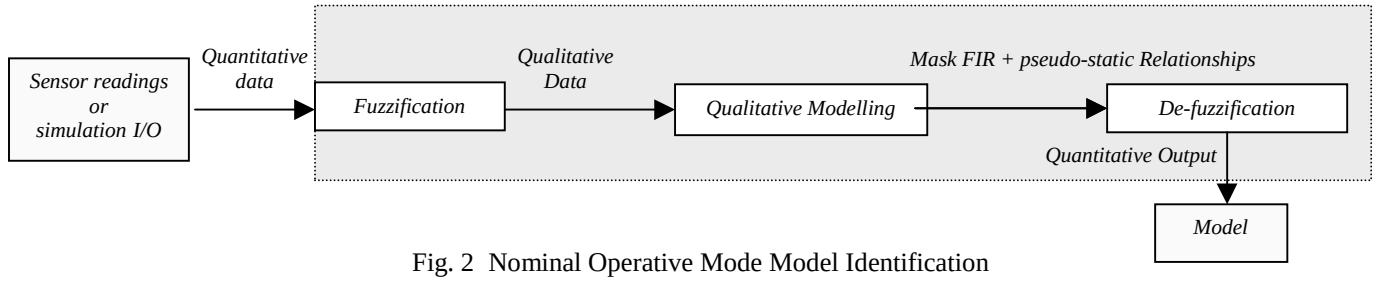




Fig. 2 Nominal Operative Mode Model Identification

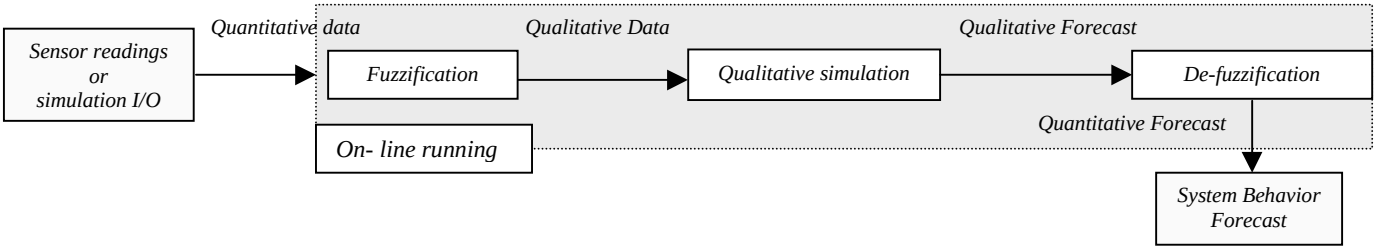


Fig. 3 Operative Mode System forecast

| Sampling instant (sec) | Expected by the "reference model" (Watts) | Read from the sensors (simulator outputs) (Watts) | Envelope Min | Envelope Max |
|---|---|---|---|---|
| 1992 | 426.075 | 426.070 | 425.298 | 426.855 |
| 2002 | 426.039 | **408.718** | **425.266** | **426.821** |
| 2012 | 426.005 | **408.686** | **425.236** | **426.786** |
| 2022 | 425.974 | **408.654** | **425.205** | **426.752** |
| 2032 | 425.935 | **408.625** | **425.173** | **426.687** |

Tab. 1 Expected-read outputs as soon as the failure is injected

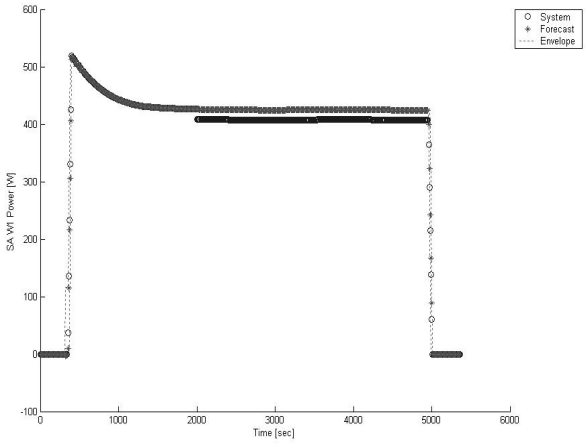



Fig. 4 W1 SA Power supply Voltage Forecast with failure injection at 2000s

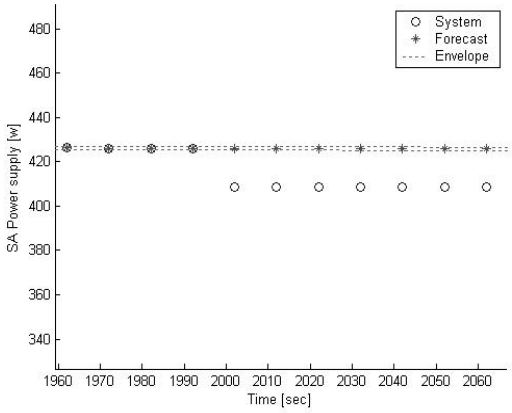
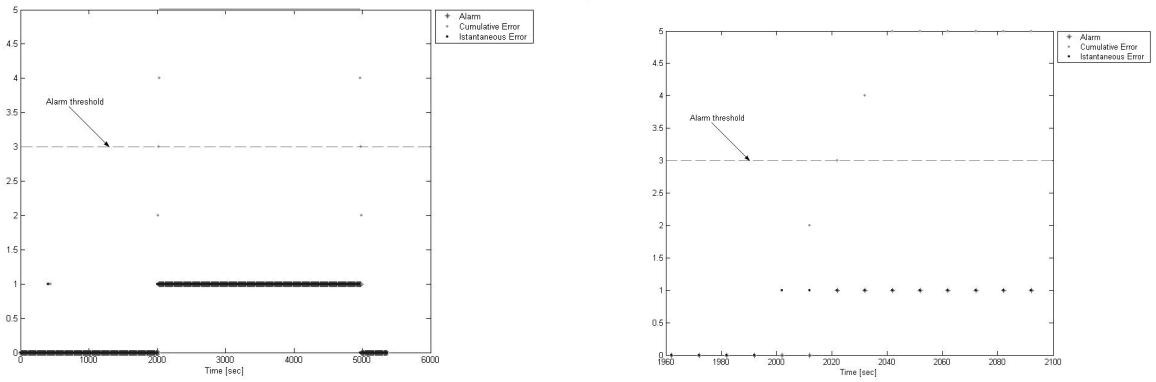



Fig. 5 Zoom failure injection instant: sensor readings out of modelled "envelope", symptoms detected

| Sampling instant (sec) | Instant Error | Cumulative Error | Alarm |
|---|---|---|---|
| 1992 | 0 | 0 | 0 |
| 2002 | 1 | 1 | 0 |
| 2012 | 1 | 2 | 0 |
| 2022 | 1 | **3** | **1 active** |
| 2032 | 1 | **4** | **1 active** |

Tab. 2 Sensor readings and alarm triggering with failure injected




(a) (b)

Fig. 6 Out of "envelope" error and alarm triggering(a)-zoom(b): symptom detection

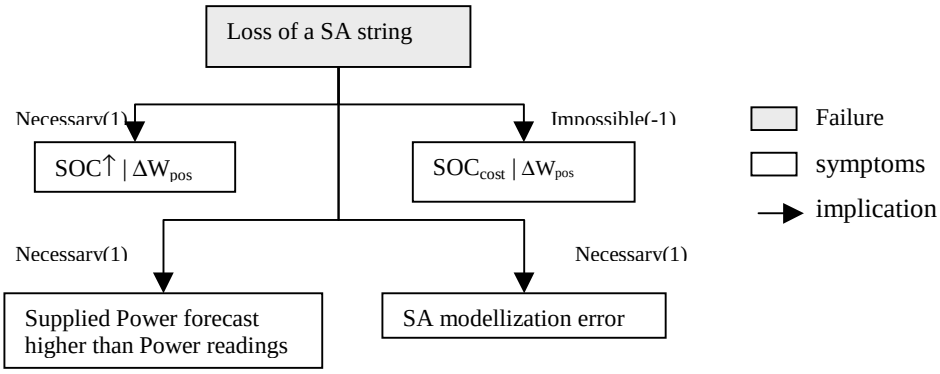



Fig. 7 Logical behaviour for the solar arrays string failure with certainty degree labels

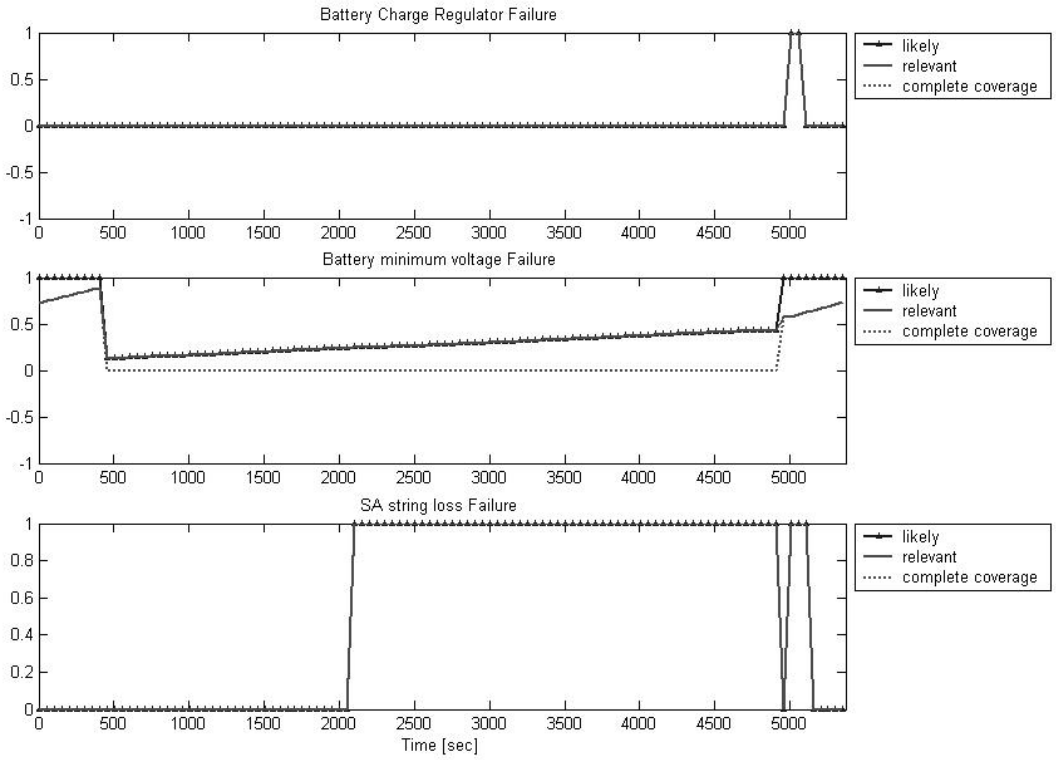



Fig. 8 SA string loss injected at 2000s: abrupt and incipient failure Identification