MODEL BASED FAULT DETECTION OF COMPLEX PROCESSES - PRACTICAL EXAMPLES

Attilio Brighenti
S.A.T.E. S.R.L.
VENICE, ITALY
attilio.brighenti@sate-italy.com

Luca Fogar
S.A.T.E. S.R.L.
VENICE, ITALY
luca.fogar@sate-italy.com

Marco Nadalin
S.A.T.E. S.R.L.
VENICE, ITALY
marco.nadalin@sate-italy.com

ABSTRACT/SUMMARY

This paper describes the capabilities of black box models of detecting incipient faults on complex systems; an example with a centrifugal gas compression system is reported.

The algorithms proposed are able, on the one side, to detect even incipient faults, while on the other to prevent the raise of false alarms by a combined use of filtered data and a model validity check. The latter, in particular, is evaluated using a probability density function, corrected using the Liapunov settling time estimation, for detecting unknown working conditions.

The algorithms proposed are definitely not case dependent since they can easily be configured following a training procedure and, moreover, can be applied to a variety of cases (automotive, Metals industry, Oil & Gas industry, etc.) [1].

Keywords: industrial diagnostics, fault detection and identification, compression systems.

INTRODUCTION

Due to the reduction of installation costs, in the last few years the number of sensors installed on a single plant has started to grow rapidly and has reached such a high number that plant operators might find difficulties in controlling and monitoring all of them, with the risk of missing or misinterpreting important symptoms and the causes of alarms/warnings.

This is particularly true in the case of incipient faults, where the rate of signals' modifications is very small (in absolute value), or when faults can be detected only comparing and correlating among them different signals at the same time. Moreover, even if the operator detects an abnormal behaviour, he/she might find it very complex to recognize and locate its cause, thus to decide in time the proper corrective actions.

The use of models can definitely support the operator in these tasks, thanks to their capability of reproducing the normal behaviour of one or more output signals[1], as a function of a pre-defined set of input signals. By comparing the simulated output of a model with the measured ones the detection of faults is possible at their onset and evolution, well before they fully develop.

Generally speaking these models can be divided [2] into two categories:

1. **Transparent box models**: the model is obtained using the description of the parts constituting the system, which is described through the laws ruling the parts themselves. As a consequence, a physical background of the process to be modelled is mandatory for creating the model, and, moreover, the physical laws that rule it must be known. The applicability of these models stems from the knowledge of the actual parameters governing the systems dynamics;
2. **Black box models:** in some cases the user is not interested, or it is not possible to obtain a "physics based" model, and the main aim is to reach a synthetic description of the system, subsystems or components, which could describe the dynamics of some measurable signals in the system with a certain accuracy. In this case a physical background of the process or component to be modelled is not mandatory for creating the model, but, nonetheless, could be very useful (as will be shown later on) in leading the model creation process. These models are created by *training* certain general algorithms (e.g. State Space, Neural Networks, ARX, ARMAX, Signal analysis and statistics, etc.) on a subset of measured input and output signals and *validating and testing* the parameters obtained with a different subset.

An advantage of black box models over the transparent ones lies in their computing efficiency and formal simplification, so that both can be useful at different stages of the project/process lifetime. Of course hybrid approaches can be used, to describe subsystems whose physics and real parameters are fairly well known together with others whose are not.

In this paper both the two above approaches have been used, namely:

1. **a transparent model** (***COMPSYS™***) [4][5][6][7] has been used to generate a realistic set of signals that could be eventually measured in the real system to be monitored (both in normal operating conditions and in faulty conditions); signals for training and testing the black box models mentioned below have been extracted from these histories and are referred to as "pseudo-measured" data;
2. **a set of black box models** have been used for monitoring the signals generated with the transparent model and detecting the faulty conditions.

Usually both the *training* and *testing* signals shall be acquired using the Plant Data Acquisition System (briefly PDAS) during the commissioning phase, but the use of a transparent model (if available) could allow a lot of time and effort to be saved, since:

[1] In the following models with one input and one output will be called SISO models, models with many inputs and one output will be labelled MISO models, while models with multiple inputs and outputs will be mentioned as MIMO models.

1. a lot of experience can be gained on the system itself (e.g. to define which signals can be used as inputs, their ranges and how to acquire them in order to produce robust black box models), without interfering with the productive cycle (see section “Signals acquisition”), at least during the first development stage;

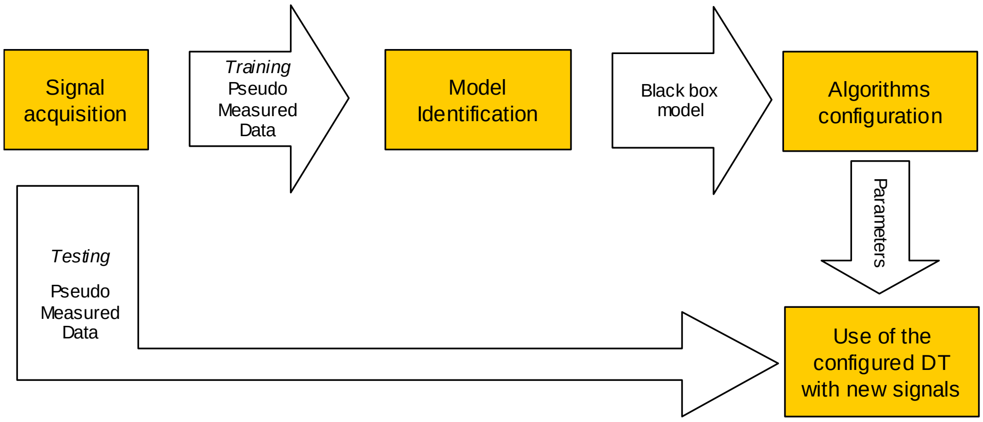

Figure 1 – FDI system configuration flow chart.

2. experiments on the real plant can be limited and studied off-line in order to produce robust results (see sections “Signals acquisition” and “Signals generated”);
3. one or more “first trial” black box models can be defined, and then used (maybe modified and improved) using the measured signals coming from the PDAS.

The task of creating these black box models might be very complicated and might require different capabilities, which may or may not belong to the same person. In particular three kinds of user/developer professional profiles can be recognized:

1. **System Identification Engineer (briefly SIE)**: has a deep knowledge in system identification techniques but does not necessarily have similar capabilities concerning the process to be monitored:
2. **Process Engineer (briefly PE)**: has a deep knowledge of the process to be monitored but a limited one in system identification techniques;
3. **Diagnostic Engineer (briefly DE)**: has a sufficient knowledge both in System Identification Techniques and in the process to be monitored, for interpreting data, but not enough experience to design and implementing these systems.

In the following a detailed description of the models’ creation and fault detection procedure applied to a gas centrifugal compressor system is described, with an example based on a *state space* type black box model.

FDI SYSTEM CONFIGURATION

The configuration of the Fault Detection Identification system (briefly FDI) has been performed as per the following phases (see Figure 1):

1. **Signals acquisition**: collection of a set of signals suitable for training the model, i.e.

the so-called *training set* (section "Signals acquisition"), and different test sets, i.e. the so-called *testing set*s, for testing the fault detection capabilities of the algorithms (sections "Signals acquisition" and "Signals generated"). As previously stated (see section "Introduction"), in this work, data for system identification and fault detection have been produced using a transparent model (***COMPSYS™***) and not directly measured on a plant. However, ***COMPSYS™*** actually substitutes the real plant, and the fault detection procedure, which is described in this work, can be straightforwardly applied to real measured data. In order to highlight this aspect, in the following the ***COMPSYS™*** data will be referred to as "pseudo-measured" data;

2. **Model Identification**: identification of a State Space Model for simulating one or more output (section "Process Model Identification");
3. **Fault Detection Algorithms Configuration**: this step is constituted by two different sub-phases
    a. Creation of a function for evaluating the reliability of the State Space Model (section "Reliable Domain of the Model: model validity");
    b. Definition of the parameters needed for performing the fault detection (section "Diagnosis of faulty conditions");

During phases 2 and 3, a certain number of parameters (e.g. residuals' standard deviations, Butterworth filter parameters, prediction horizon, etc.), which are case dependent, are defined by the user.

These parameters will be used for *configuring*[2] all the tests (referred to as *detection test - DT* - in the following), before being able to use them with realistic signals (section "Detection test software service activation").

Signals acquisition

As previously stated, plants are equipped with a high number of sensors, which can measure a wide set of quantities, and, via the PDSA and the plant internal network, provide them to a variety of PLCs and PCs.

Usually these signals are used by process and equipment controllers and displayed on a variety of consoles or operator interfaces for a visual check by the operators, but, of course can be provided to and recorded in an internal database (briefly DB in the following).

In order to create a black box model, historical data should be obtained by the DB (or directly recorded from the PDAS), pre-processed (e.g. filtered, outliers replaced, data synchronized, etc.) and made available to the SIE (or DE).

According to the nature of the process to be monitored, the data acquisition issue, can reveal some limiting aspects [2]:

1. It might be possible to record real signals only during the normal operating conditions (no external disturbance can be injected into the system);

[2] All the algorithms developed within this work are general purpose ones, i.e. they are not specific to certain applications.

2. Small disturbances can be injected into the system but only by small changes of some quantities (valve opening, gas density modification, etc.); in this case the experiments through which the necessary *training* data are acquired can be planned, to obtain data which make the identified model more robust.

For the work described in this paper a transparent model has been used to simulate the measured data coming from the PDAS; therefore the above limitations do not apply and it has been possible to obtain sufficient *training* and *testing* sets. In particular the *testing* data has been created injecting a variety of faults (e.g. a valve leakage) into different points of the system (see sections "Possible faults examples (ASV leakage)"and "Signals generated").

Process Model Identification

Once the data has been made available to the SIE (or DE), the latter should perform the following steps for identifying a model suitable for fault diagnosis:

1. **Selection of the black box model family** (e.g. State Space, Neural Networks, ARX, Signal Based, etc.): the choice is driven by some physical considerations (e.g. if the relationships between input and output might be linear or non-linear, etc.) and also by the available computational time (e.g. a Neural Network approach is usually more time consuming than a State Space one).

   In this work a State Space Model has been chosen, namely a system ruled by the following equations (for a discrete time model):

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) + Ke(t) \\ y(t) = Cx(t) + Du(t) + e(t) \end{cases}$$

Equation 1

   where *A*,*B*,*C*,*D*,*K* are the so-called State Space Matrices, *u(t)* is the input vector, *y(t)* is the output vector, *x(t)* is the states vector and *e(t)* is a "white noise signal", with a specific variance.

2. **Selection of the model's input and output**: according to what previously stated, a huge amount of measured signals might be available and the choice of the important ones might be quite complex. Therefore a knowledge of the process to be monitored is essential and the support of a PE could be necessary;

3. **Definition of the model complexity**: it is unlikely that sufficient a-priori information concerning the model order that best describes the system is available, so that different models with different complexity should usually be evaluated. Then, according to some "quality criteria" (e.g. white noise pattern of the residuals), the most suitable one shall be chosen.

Reliable Domain of the Model: model validity

The black box model has been identified using a set of samples of the input/output relationship (the *training* set). Therefore the estimation of the output will be reliable [3] inside the region of the input space $X \subset \Re^n$, where n is the number of inputs, represented by the input samples of the training set. This region of the input space will be called the *reliable domain* of the model.

Using the input samples, the *probability density function* (PDF) of the inputs has been evaluated. Let $p_x[k]$ be the estimated PDF of the input vector $x[k]$. High values of $p_x[k]$ indicate a good representation of $x[k]$ in the training set, and, hence, a good characterization of the simulated output $y[k]$, under normal operating conditions. Vice versa, low values of $p_x[k]$ would indicate a poor representation of the input vector $x[k]$ in the training set, hence, a low level of reliability of the simulated output $y[k]$,

In this work the PDF function is normalized using its maximum value (thus the region with the highest PDF will have a value equal to 1) and the *validity* of each input sample, thus of the model, is computed using this normalized function. If the input vector falls outside the training set input space (even with only one component), a PDF equal to zero is assigned to this sample.

The reliability of the model is influenced also by the so-called system *settling time* (briefly ST in the following): during the model identification phase, the identification algorithm estimates the initial state (see Equation 1) vector of the system. In general this is different for different actual cases[3] but, since no a-priori information are usually available on the states, it is used also for the new cases. Therefore the model requires a certain amount of time to dampen this initial inaccuracy and start providing reliable outputs.

In order to estimate the ST the Liapunov theory [8], which relies on the following function (called *Liapunov function*), has been applied

$$V(x) = x^T Px \qquad \textbf{Equation 2}$$

where x is the states vector and P defined as

$$N = -2(PA)_{sym} \qquad \textbf{Equation 3}$$

with N an arbitrary positive defined or positive semi-defined symmetric matrix, and A is the state matrix identified during the training session.

The ST is estimated using Equation 4.

---

[3] An example can be that of a two masses-springs-dampers system, where the states of the system are the position and the velocities of the system masses. The model training session can be performed with signals obtained by exciting the system with the masses initially at rest at a certain offset, e.g. x_1 and x_2. The initial state vector, therefore, is $[x_1\ x_2\ 0\ 0]$. In order to compare the model output with real ones, the model should be initialised at the same state of the latter and, in general, it is very unlikely that the initial states vector matches that of the identified one, since, for instance, the two masses might have a velocity different from zero or a position different from x_1 and x_2 when the model is started.

$$T_S \leq -\frac{1}{a}\ln\frac{V[x(T_S)]}{V[x(0)]}$$

Equation 4

where a is the minimum eigenvalue of the matrix NP^{-1}, $V[x(0)]$ is the initial state of the system, and $V[x(T_S)]$ is the value of the Liapunov function evaluated for the state corresponding to the settling time. This state can be obtained using the procedure described in [8].

The time TS represents the *maximum* time required by the model (when used with signals different from the training ones) to dampen the initial numerical transient response and to provide reliable results. Of course it might happen that this limit could be reduced using a visual evaluation of the simulated outputs, but maintaining this limit provides a more robust on-line tool (see section "Running the FDT software"), with lower possibility of generating false alarms.

Finally the model validity computed using the PDF is treated using the information coming from the ST evaluation using the following fuzzy logic rule:

1. For $0 \leq t < T_S$ the model validity obtained using the PDF will be multiplied by a linear function, which is equal to 0 at t=0 and to 1 at t=T_S;
2. For $t \geq T_S$ the model validity obtained using the PDF will be used as the final one.

Diagnosis of faulty conditions

In order to detect faulty conditions of the process, the model's *residuals* (i.e. the differences between the measured and simulated outputs) must be analysed, and, in particular, the following characteristics should be checked:

1. **Thresholds trespassing**: according to the system identification theory [2], the model's residuals should reveal a "white noise pattern", i.e.
   a. Have a mean equal to zero;
   b. Have a normal distribution.

   Condition b) implies that 99.73% of the measurements (model's errors) should fall within the $\pm 3\sigma$ interval during normal operating conditions, where

$$\sigma = \left(\frac{1}{n-1}\sum_{i=1}^{N}(x_i - \bar{x})\right)^{\frac{1}{2}}$$

Equation 5

   is the signals' **standard deviation** and

Equation 6

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{N} x_i$$

is the signals' **mean**, with *n* number of samples.

First of all the training set residuals are used for evaluating the standard deviations (one for each output signal, in a MIMO, multi-input-multi-output, case), in order to define the $\pm 3\sigma$ interval.

These upper and lower thresholds will be used during the fault detection test to check the signal of the residuals. Low pass filtering of the signals and outliers removal is convenient both before the training session and the on-line operation, to avoid odd alarms .

2. **Residual derivative analysis**: since the analysis of the $\pm 3\sigma$. thresholds trespassing can detect only faults which have already occurred, it has been decided to also use a trend analysis to produce an advance warning of future alarms.

   With this purpose, the derivative of the filtered residual signals is calculated and compared with a pre-defined value, which can be customized on a case to case basis.

   In particular this limit value is computed using the following equation:

Equation 7

$$f'_{\max}(\varepsilon) = \frac{3\sigma}{PH}$$

where PH (prediction horizon) is the time interval for which the trend should be checked, see Figure 2.

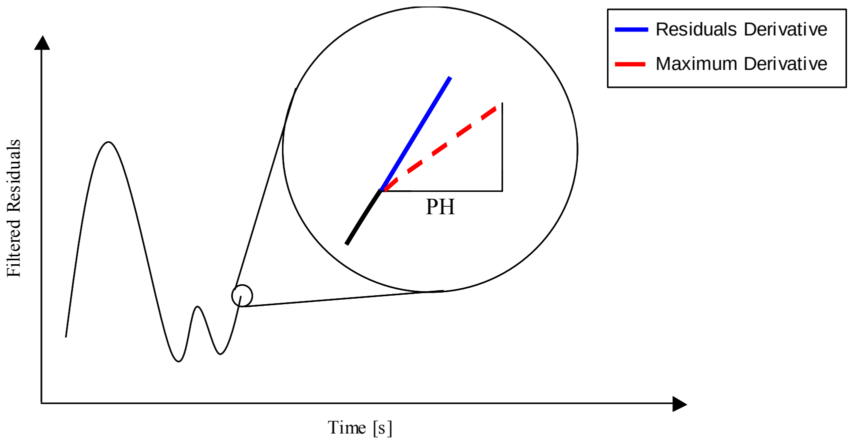

Figure 2 – Residuals' trend analysis.

As a matter of fact $f'_{\max}(\varepsilon)$ is the maximum allowed value of the derivative, i.e. the value that leads to an increase of the residuals equal to $+3\sigma$ within the prediction horizon.

The signal of the residuals' derivative is normalized using $f'_{\max}(\varepsilon)$, thus a normalized signal equal to 1 implies that the derivative is equal to the maximum allowed.

The two pieces of information, obtained using the two residuals characteristics described above, are coupled together to define the system behaviour (in the following, the latter behaviour of the system will be called *decision validity – DV*), using the three-dimensional fuzzy logic function of Figure 3.

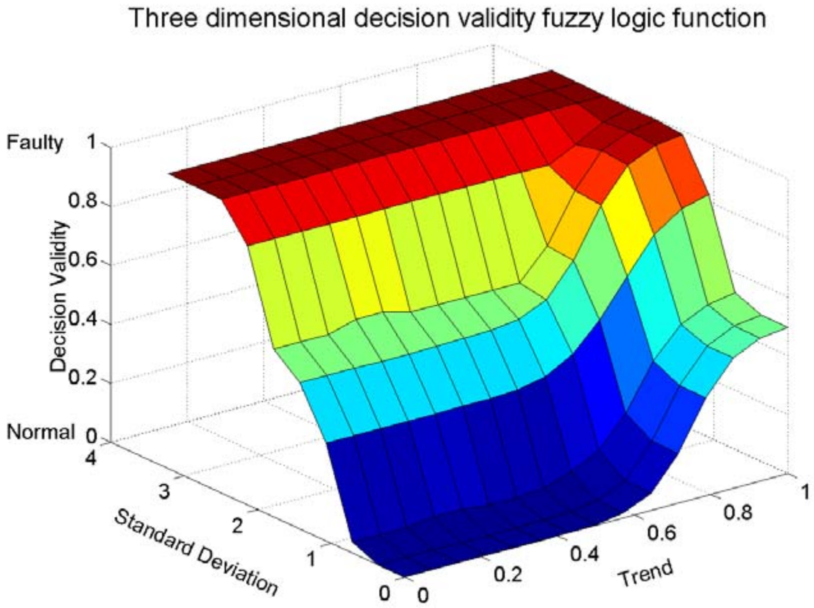

Figure 3 – Three dimensional decision validity fuzzy logic function.

It must be underscored that, according to the function of Figure 3, a faulty alarm can be raised because either:

1. The absolute value of the filtered residuals signals approaches the ±3σ thresholds; or
2. The derivative of the filtered residuals signals approaches the maximum value allowed.

If both features appear simultaneously, the alarm level is increased.

Another quantity, which can be calculated for evaluating the system behaviour[4], is the so-called *asymmetry factor* (AF), defined as:

$$AF = \frac{A_{LEFT}}{A_{RIGHT}}$$

[4] In this work the AF has been used only for a qualitative check of the system behaviour and did not influence the evaluation of the DV.

where A_{LEFT} and A_{RIGHT} are, respectively, the area covered by the residuals' distribution on the left and right side of the signal's average. Residual falling outside the $+3\sigma$ interval are not taken into account. If the distribution is truly Gaussian, the AF is equal to 1.

Anomalous patterns definition rules (symptoms)

The output of a DT is a *symptom*, which can be defined as "a judgement on the plant/system behaviour", and it is built up using two values:

1. Decision Validity (DV): a judgement on the system operating condition (0 normal – 1 faulty);
2. Model Validity (MV): a judgement on the reliability of the identified model (0 not valid – 1 fully valid);

Detection test software service activation

As stated before (see section "FDI system Configuration"), a DT can be performed using a set of MATLAB®[5] functions, only once all the necessary parameters have been defined (namely the DT has been configured) during the *training* session.

If the test is called for the first time, the model validity is treated using both the PDF and the ST (see section "Reliable Domain of the Model: model validity"). Otherwise, if the DT is called more than once, two scenarios can take place:

1. **The time histories of the two calls overlap one another, at least for one sample**: in this case information concerning the previous states of the system can be obtained from previous runs, and, since the initial state is known, the ST correction of the MV can be avoided;
2. **The time histories of the two calls do not overlap one another**: in this case information concerning the previous states of the system are not available and the ST correction of the MV must be used.

These calls to the DT can be performed:

1. **On line**: i.e. the DT is called while the plant/system is operating, providing it with signals acquired shortly before or in real time. This implies that it would be possible to have a quick feedback on the system operating condition, thus to plan some corrective actions, if necessary;
2. **Off line**: i.e. using signals, which have been previously recorded and stored in a DB. In this way the system behaviour can be monitored a-posteriori, in order to understand, for instance, whether incipient faults are developing or when and how already detected faults have started to be visible, or to monitor processes, with intrinsic slow evolution, which do not require immediate intervention.

[5] MATLAB® is a product of TheMathworks Inc. (Mass. USA), used by the authors for the development of the models and the system identification procedures.

These two approaches are foreseen in an ongoing European project, MAGIC[6], whose aim consists in the development of an agent based diagnostic system running on a distributed computer architecture [9][10][11][12][13][14][15].

FDI OPERATIONAL PHASE: EXAMPLES

Natural gas compression system

Plant Description

The practical example taken for this work refers to a typical large compression facility, e.g. for a propane refrigeration unit, as shown in the process scheme of Figure 4.

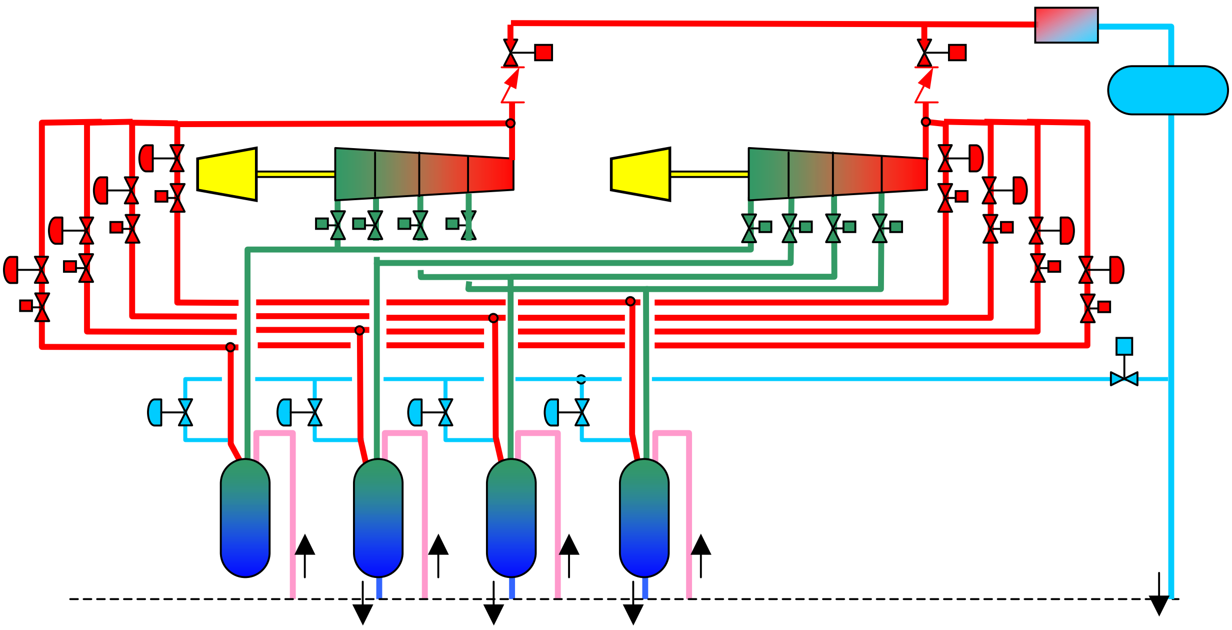

Figure 4 - Simplified PFD of a typical propane compressors system

Such a complex plant implies several controllers to maintain the process and machines efficiency as high as possible, associated to several MW power, and the plant safety.

Simulations, based on transparent models, are frequently used during the design phase of these systems to check the design and verify the controller's capability to prevent unwanted dangerous situations, particularly the occurrence of compressors' surge[7].

---

[6] Multi-AGents-Based Diagnostic Data Acquisition And Management In Complex Systems: the partners of this project are the University of Duisburg-Essen, the University of Karlsruhe, the Institute National Polytechnique De Grenoble, SMS-DEMAG A.G. and SATE S.r.l.

[7] "Surge" is a phenomenon that, for axial and centrifugal compressors, is equivalent to the stall of airplane wings and implies the loss or reversal of the flow, the loss of pressure ratio generation by the machine, the potential damage to the blading and sharp pressure rise in the upstream piping. The higher the pressure ratio, the worse the consequences of surge are.

COMPSYS™ is a proprietary compression systems simulation environment, based on the above mentioned *transparent* modelling approach, and library, based on MATLAB®-SIMULINK®, which was qualified through various services performed on behalf of important compressors manufacturers and engineering companies [4][5][6][7].

The ***COMPSYS™*** simulator is able to describe:

- Real gas thermodynamics;
- 3D dynamic compressor maps (based on instantaneously corrected flow, polytrophic head and speed);
- Non linear gas dynamics through valves, orifices, head loss elements;
- Dynamics of heat exchangers, based on instantaneous convection coefficients and thermal flow calculation;
- Separation of gas components fractions below the dew point (if applicable);
- Drivers shaft dynamics, with feedback torque control, based on several modes of operation;
- Implementation of the specific actual controller schemes for
  - Antisurge;
  - Performance (speed control);
  - Load sharing;
  - Heat exchangers;
  - Process valves.

Possible faults examples (ASV leakage)

One possible fault, which could arise in a compression system like the one shown in Figure 4, is the offset of an antisurge valve actuator, yielding valve leakage, thus recirculation of hot or cooled gas from the discharge to the suction side of the compressors. This implies reduction of flow or loss of power and efficiency.

This is a significant, though not the sole, possible example of the application of FDI techniques to these complex systems, which is used in this work as a FDI test case.

Signals generated

The training signals have been generated as follows:

1. System's normal operating condition with the plant's nominal mass flow rate from t=0 s to t=20 s;
2. Application of a +10% inlet mass flow rate step from t = 20 s to t = 120s;
3. Return to the nominal mass flow rate and stabilization from t = 120 s to t = 220 s;
4. Application of a -10% inlet mass flow rate step from t = 220 s to t = 320s;
5. Return to the nominal mass flow rate and stabilization from t = 320 s to t = 1500 s.

Both the 1st and 2nd stage *testing* signals (**1STCBLEAK** and **2NDCBLEAK** respectively) have been generated as follows:

1. The cold by-pass valve is closed between 0 and 500 s (time t_1);
2. The cold by-pass valve starts opening according to a linear trend at 500 s;
3. The cold by-pass valve stops opening at 1000 s (time t_2), when it has reached 10% opening and starts closing according to a linear trend;
4. The cold by-pass valve returns closed position at 1100 s (time t_3).

It must be underscored that a 10% opening of the valve under consideration corresponds to 1% of the maximum mass flow rate through the same valve.

Identified model

According to the procedure described in section "Process Model Identification", the following black box (BB in the following) MISO models have been identified. Their main characteristics are summarized below:

1. Model monitoring the 1st stage mass flow rate (briefly **M1STMFR**):
    a. Input:
        i. Driver Speed;
        ii. 1st stage suction temperature;
        iii. 1st stage suction pressure;
        iv. 1st stage discharge pressure
    b. Output
        i. 1st stage mass flow rate;
    c. Order of the model equal to 2;
2. Model monitoring the 2nd stage mass flow rate (briefly **M2NDMFR**):
    a. Input:
        i. Driver Speed;
        ii. 2nd stage suction temperature;
        iii. 2nd stage suction pressure;
        iv. 2nd stage discharge pressure
    b. Output:
        i. 2nd stage mass flow rate;
    c. Order of the model equal to 2;

In order to evaluate the overall quality of the identified model, different characteristics have been taken into account. The results summarized below are relevant only to model **M1STMFR**, the other's being similar:

1. **The residuals' absolute values**: considering Figure 5 - comparison between the pseudo-measured and BB simulated signals - and Figure 6 - residuals' pattern - it can be noticed that the identified BB model is able to reproduce the output signal producing residuals contained within a narrow bandwidth (maximum residual equal to 3% of the peak to peak distance and minimum residual equal to –1.07% of the peak to peak distance).

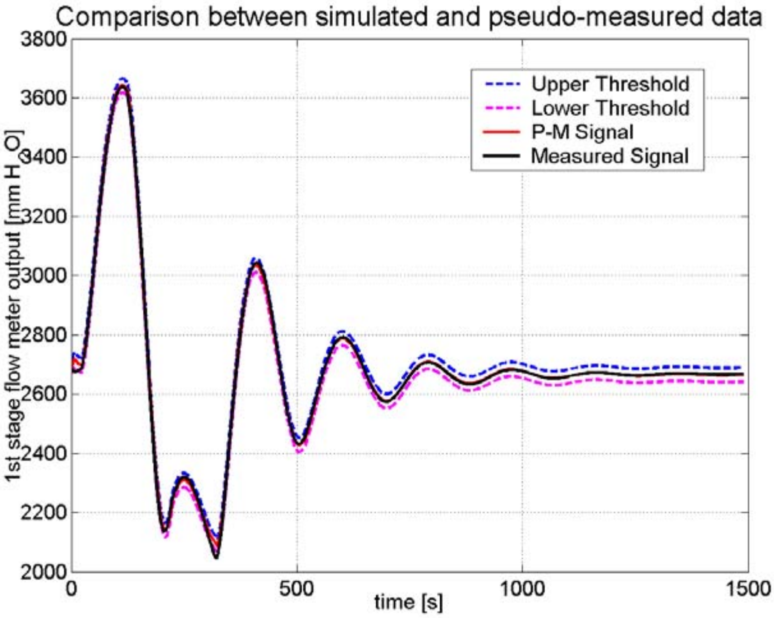

Figure 5 - Comparison between Pseudo-Measured (P-M) and simulated signals of the training set of BB model M1STMFR (flow rate measurement is expressed in mmH_2O).

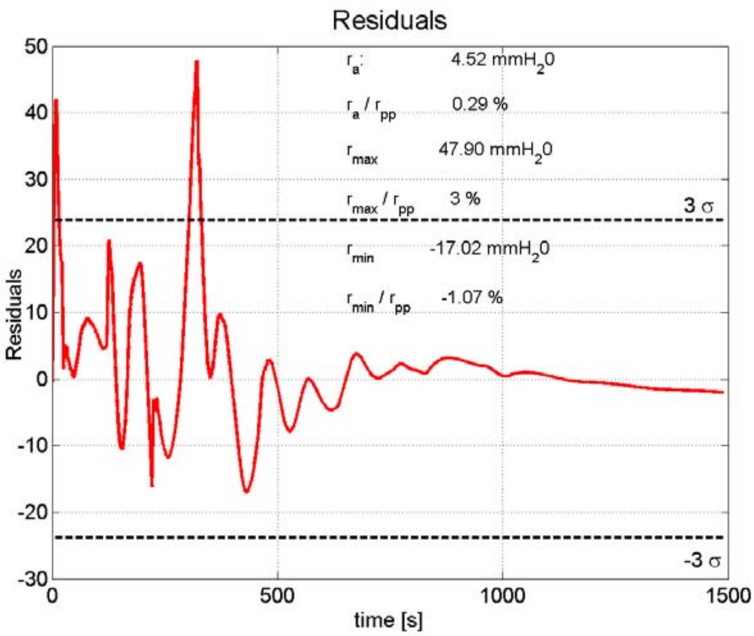

Figure 6 - Residuals of the training set of BB model M1STMFR (flow rate measurement is expressed in mmH_2O).

Moreover 97.47% of the residuals fall within the $\pm 3\sigma$ interval (this value is very close to the theoretical 99.73% of a truly Gaussian distribution), and when they do not, if the model validity (Figure 8) is taken into account, it can be noticed that the model has a very low validity (which means that few output values were available during the training in this particular input's region);

2. **The residuals' distribution**: considering Figure 7, it can be underscored that the residuals' distribution is symmetric (AF equal to 1.1081) around a mean value which is very close to zero, especially if considered related to the output peak to peak distance;
3. **The symptom values**: since the training set has been created using a normal operating condition, the DV should be always equal to 0, but, considering Figure 8, it can be notice that this is not strictly true. This can be explained considering that the symptom (which is the only quantity to be considered for fault diagnosis) is constituted by two values, which must be considered together, namely DV and MV. As a matter of fact, in cases where the DV approaches 1 (which should hint to the existence of a faulty condition), the MV decreases, which means that the simulated output is not very reliable. Therefore combining these two pieces of information, the operator can understand that the faulty alarm could be due, in this case, to the unreliability of the identified model during that particular operating condition, and not to a real developing fault.

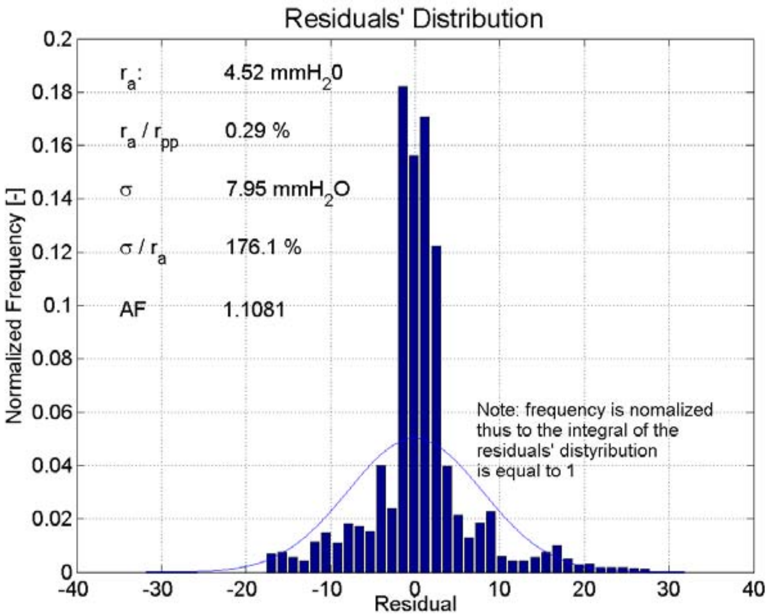

Figure 7 - Distribution of the residuals between the *training* set pseudo-measured and simulated signals of model M1STMFR (flow rate measurement residuals are expressed in mmH_2O).

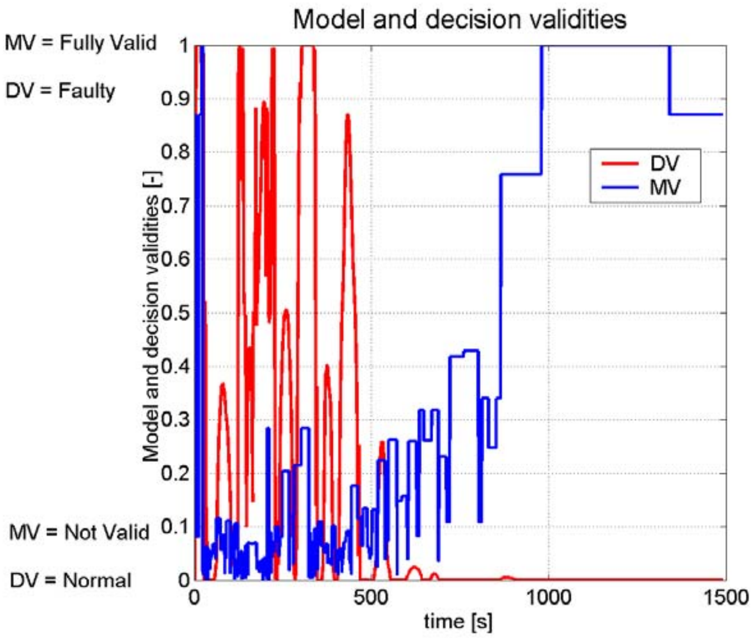

Figure 8 - Symptom obtained on the *training* set of BB model M1STMFR (DV and MV respectively).

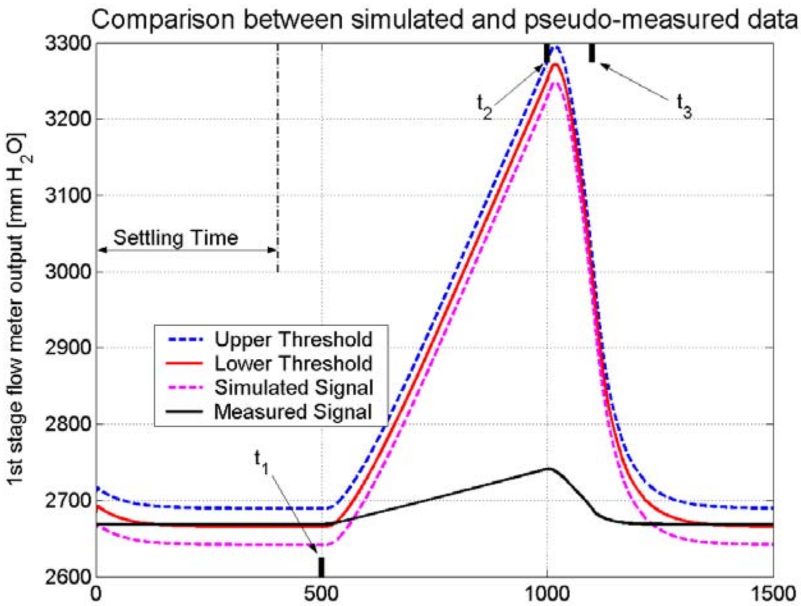

Figure 9 - Comparison between Pseudo-Measured and BB simulated signals of a testing set of BB model M1STMFR used with faulty condition 1STCBLEAK - fault starts developing in t_1; fault stops developing in t_2; fault ends in t_3.

Running the FDT software

The BB models described in section “Identified model” were applied to the faulty conditions reported in section “Signals generated”. In particular BB model **M1STMFR** was used with data obtained in the faulty condition **1STCBLEAK**. The relevant results are reported in Figure 9.

This figure, together with Figure 10, clearly shows that:

1. A certain amount of time is needed by the BB model output to become stable (however smaller then the pre-defined ST);
2. After the fault injection (time t_1) the BB model output starts deviating from the pseudo-measured values;
3. Once the fault starts being reduced (time t_2) the BB model output returns closer to the pseudo-measured one, till it starts providing values very close to the pseudo-measured ones.

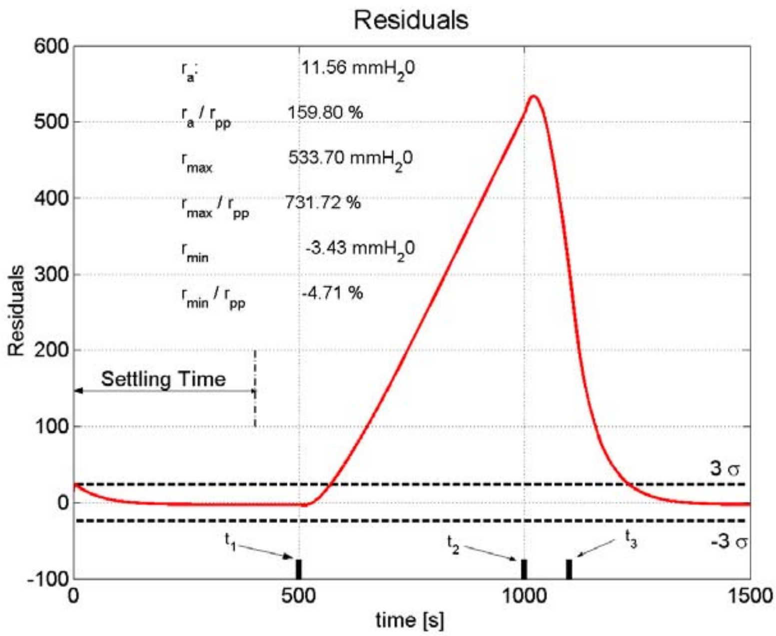

Figure 10 - Residuals of the testing set of model M1STMFR used with faulty condition 1STCBLEAK - fault starts developing in t_1; fault stops developing in t_2; fault ends in t_3.

Moreover, Figure 11 highlights that the residuals’ distribution, differently from the *training* set one of Figure 7, is clearly asymmetric (AF = 2.6388), and has a mean value far from zero, i.e. the residuals do not have a “white noise pattern”.

Therefore, in a real situation, by a visual comparison between the BB model’s and the measured output, it would be possible to detect that a fault occurred. In addition to this, the main aim of this work was to demonstrate that this check could be done automatically, which appears to have been fulfilled (Figure 12).

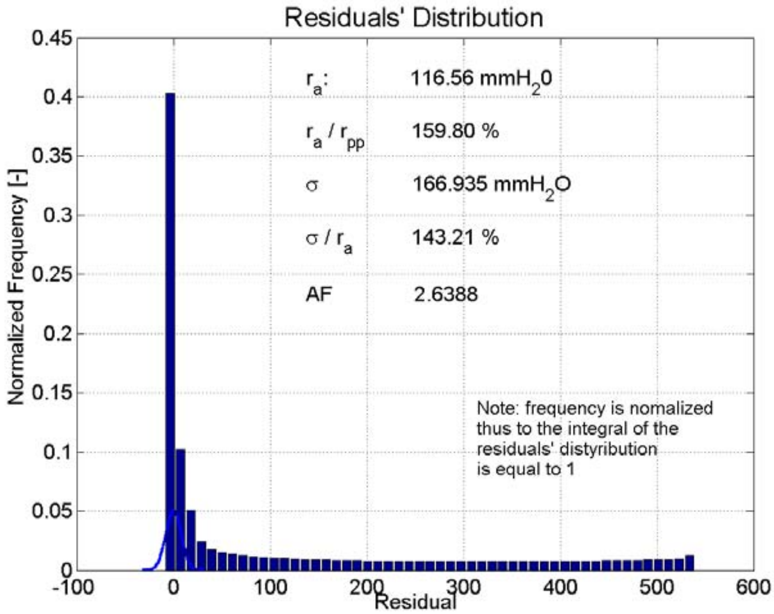

Figure 11 - Distribution of the residuals between the pseudo-measured and BB simulated signals of model M1STMFR used with faulty condition 1STCBLEAK. The flow rate measurement residuals are expressed in mmH_2O

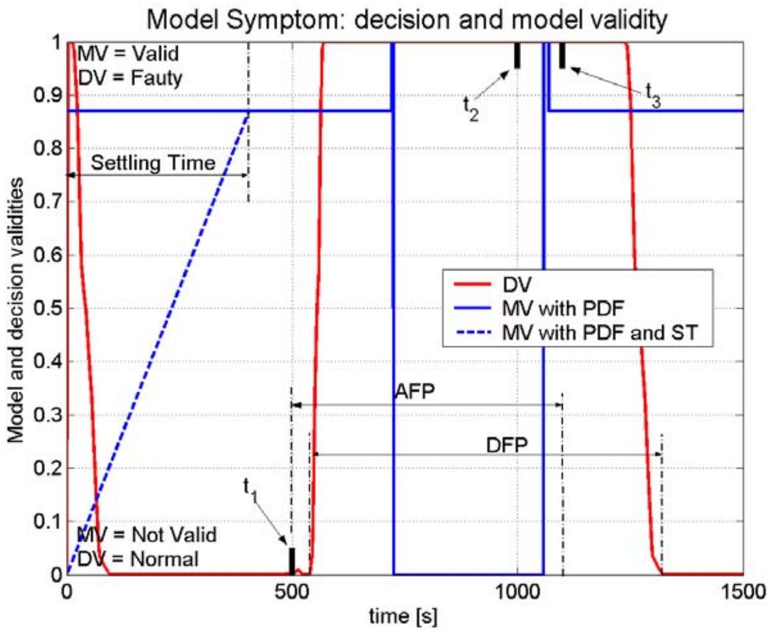

Figure 12 - Symptom (DV & MV) obtained on the testing set of model M1STMFR used with faulty condition 1STCBLEAK - fault starts developing in t_1; fault stops developing in t_2; fault ends in t_3. AFP: Actual Fault Period. DFP: Detected Fault Period.

Indeed the DT starts highlighting an abnormal operating condition 40 seconds after the fault started developing, when the valve opening is equal to 0.8% and the mass flow rate is 0.08% of its maximum, while it clearly raises an alarm after 70 s, when the valve opening is equal to 1.4% and the mass flow rate to 0.14%.

After 220 s the BB model starts loosing validity, since the fault is influencing the BB model's input, which then falls outside the training space.

Once the fault starts being reduced, and then completely removed, the BB model validity increases again. It requires a certain amount of time, which is comparable to the time needed at the beginning to start providing stable results, to detect that the fault has been removed, i.e. to provide a MV equal to 0. Therefore, the Detected Fault Period (DFP) is 30% longer than the Actual Fault Period (AFP in Figure 12).

THE *ENDURANCE* STUDY

To highlight the general applicability of the FDI approach described above, another work is briefly mentioned: the ***ENDURANCE*** study, which was performed by the leading author and others, on behalf of a renowned car manufacturer, between December 2000 and April 2001. They used the data history of a number of prototype cars under endurance tests, i.e. tests in driving conditions through standard urban and extra urban road or racing tracks. Some of these cars underwent real failures, such as the breakdown of the engine, or evident faulty or abnormal conditions, related to the lube oil or the cooling water systems.

Through the application of various types of identification techniques it was possible to:

1. identify parameters of the residuals statistics based on the critical subsystem BB models obtained by identification techniques that can be used as symptomatic variables for prognosis and diagnosis purpose;
2. detect faulty patterns and their trends several days before the test drivers warned of "unexplained" anomalies and/or the final engine failed (i.e. between 3000 and 11000 Km before breakdown);

CONCLUSIONS

The work reported in this paper, aimed at demonstrating the feasibility of creating an automatic fault detection system based on "black box" modeling, showing as an example the use of State Space models.

The fault detection algorithms (*detection test*) are based on the analysis of the differences between the outputs provided by an identified "black box" model and the measured ones, namely the model's residuals, considering and verifying the model validity under the running conditions.

The performances and reliability of the proposed approach have been improved coupling the fault detection algorithms with a model validity estimator, based on the probability density function evaluation, corrected according to the Liapunov theory, to account for initialization transients when necessary.

The *detection test* has been tested on a real world problem simulated by a "transparent" model (***COMPSYS™***). The diagnosis of a centrifugal gas compression system has been reported, demonstrating its capabilities of detecting even incipient faults.

SYMBOLS AND ACRONIMS

AF	Asymmetry Factor
AFP	Actual Fault Period
BB	Black Box
DB	Data Base
DE	Diagnostic Engineer
DFP	Detected Fault Period
DT	Detection Test
DV	Decision Validity
FDI	Fault Detection Isolation
MIMO	Multi Input Multi Output model
MISO	Multi Input Single Output model
MV	Model Validity
PDAS	Plant Data Acquisition System
PDF	Probability Density Function
PE	Process Engineer
PLC	Process Logic Control
SIE	System Identification Engineer
SISO	Single Input Single Output model
ST	Settling Time

REFERENCES

[1] J. Deckers, O.N. Jepsen S. Latzel, H. Metz, R. Stuecher, Condition monitoring and failure diagnostics in plants of the metals industry, Safeprocess 2003, Washington, USA, June 2003.

[2] S. Bittanti, Identificazione dei modelli e sistemi adattativi, Pitagora Edittrice Bologna, 2002.

[3] A. Muñoz, M.A. Sanz-Bobi, An incipient fault detection system based on the probabilistic radial basis function network: application to the diagnosis of the condenser of a coal power plant, Neurocomputing, 23 (1998) 177-194, Elsevier

[4] Brighenti A.: "COMPSYS - Compression plant simulator" (in italian), 2nd Italian MATLAB Conference, Bologna, 6-7 Nov. 1995

[5] Brighenti A. & others: “Application of Simulation in Multi-Agent Systems”, 1st Multi Agent Systems Conference, Wien (AU) – 2nd-4th Dec. 1999

[6] Brighenti A., Boatto P.: “Dynamic, gas dynamic and thermo dynamic simulations in the automotive field” – 3rd Italian MATLAB users Conference, Bologna 8-9 Feb. 2000

[7] Brighenti A., Pavan A.: “ACUSCOMP and ACUSYS – A powerful hybrid linear/non linear simulation suite to analyse pressure pulsations in piping”, ISMA 2002 International Conference on Noise and Vibration Engineering – Leuven (Belgium) 16-18 Sep. 2002

[8] J.J. D’Azzo, C.H. Houpis, Linear Control System Analysis and Design – Conventional and Modern, McGRAW-HILL, 1988.

[9] MAGIC project website: http://magic.uni-duisburg.de/

[10] Albert M., Brighenti A., Capobianco M. & others: “Multi-Agent Systems for Industrial Diagnostics” SAFEPROCESS 2003 - 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Washington, D.C., USA 9-11 June, 2003

[11] S. Ploix, S. Touaf, J.-M. Flaus, A logical framework for isolation in fault diagnosis, Safeprocess 2003, Washington, USA, June 2003.

[12] S. Ploix, S. Gentil, S. Laseq, Isolation decision for a multi-agent-based diagnostic system, Safeprocess 2003, Washington, USA, June 2003.

[13] B. Köppen-Seliger, T. Marcu, MAGIC: an integrated approach for diagnostic data management and operator support, Safeprocess 2003, Washington, USA, June 2003.

[14] M. Albert, T. Längle, H. Wörn, M. Capobianco, A. Brighenti, Multi-Agent systems for industrial Diagnostics, Safeprocess 2003, Washington, USA, June 2003.

[15] E. Atlas, C.C. Lacatusu, M. Capobianco, S. Gentil, M. Albert, T. Marcu, B. Köppen-Seliger, Process dependent choice of diagnostic methods, Safeprocess 2003, Washington, USA, June 2003.