Hierarchic Fault Diagnosis by Pattern-Recognition Approaches  
Applied to DAMADICS Benchmark  

A. R. C. Oliveira, J. M. G. Sá da Costa *  
Technical University of Lisbon, Instituto Superior Técnico, Portugal  
*(e-mail: sadacosta@dem.ist.utl.pt)

Abstract: This paper introduces a fault diagnosis system based on pattern recognition techniques, designed and tested using simulation model of a real system. The monitored system includes an industrial actuator valve, exposed to pre-defined faults, and an evaporator located downstream the valve. The proposed method relies on a hierarchic structure of diagnosis, with parallel processing through all branches. Different feature extraction methods are studied for retaining temporal information, and for dimensionality reduction at each branch of the hierarchy. Radial basis function networks are used as local classifiers, and fuzzy logic is employed to aggregate the results in a decision signal for each fault. Implementation and results for pre-defined abrupt fault scenarios are presented and discussed.

1. INTRODUCTION

Maintenance issues are a general concern in industrial world, and have been the object of intensive research. Malfunction or simply performance deterioration of equipment or processes may lead to quality and productivity problems forcing production interruption and, if not detected in early stages, frequently require expensive repairs. In worst scenarios, human risks or, at a larger scale, ecological disasters may be involved. In recent years, approaches to these problems have emphasized the component of preventive intervention, calling for both better regular maintenance and earlier fault detection. For that purpose, research has been done to develop automatic diagnosis systems. Such systems generally involve three main stages – detection, isolation and identification of fault characteristics – but so far most effort is devoted to the first two tasks, which are therefore referred to as Fault Detection and Isolation (FDI) systems, or simply Fault Diagnosis systems.

This paper presents a fault diagnosis system designed for diagnosing an industrial control valve, regulating the inlet flow to an evaporator, subsystem found in the Polish sugar factory Lublin, S.A., and studied under the European project DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems). Both the control valve and the evaporator are modelled, identified and implemented as Simulink models to generate data representing both normal and faulty behavior across a range of typical operation conditions. The model of the valve includes 19 pre-defined faults. These models are briefly introduced in section 2.

For the proposed FDI system, pattern recognition techniques are followed, stressing the perspective of fault diagnosis as a task of classifying different dynamic behaviours, and making extensive use of soft computing techniques. The option by pattern-recognition approaches, as opposed to the more common model-based approaches, is discussed in section 3, as well as the major design and implementation steps of the proposed FDI system. A hierarchic diagnosis structure is adopted, with specific feature extraction steps for each branch, and Radial Basis Function (RBF) networks are used as local classifiers. Finally, fuzzy logic aggregates the results for each fault. Results for the set of previously defined scenarios of abrupt faults are presented in section 4 and section 5 presents the final conclusions and suggestions for future work.

2. MONITORED SYSTEM

The monitored system comprises an industrial control valve pneumatically actuated and an evaporator located downstream the valve, where a mixture of sugar and water (juice) pass through, as shown in Figure 1. The pneumatic actuator with a PID controller determines the rod position (Xrod1) of the control valve determining the inlet flow (Fi) towards the evaporator. The evaporator is basically a heat exchanger consisting of an inlet and an outlet flow of juice and a separate outlet flow of evaporated water circulating in a long pipeline. The working principle of the evaporator is simple: a fluid mixture of water and sugar enters at the inlet with a given ratio sugar mass/total mass (Xs); this is designated the inlet flow Fi; part of the water content of the mixture evaporates due to the temperature raise achieved with the heat exchanger; evaporated water is removed through the vapour outlet flow (W); the resulting fluid mixture, with an increased ratio sugar mass/total mass (Xs), is returned at the outlet; this is designated the outlet flow Fw. All 19 pre-specified faults are located at the valve, so that, for diagnosis purposes the evaporator acts only as providing extra variables for a more robust and reliable diagnosis. A complete analytical model, reproducing normal and faulty operation of both valve and the evaporator were implemented as Simulink models in MatLab to generate the training and test data. The following 14 variables are selected for FDI purposes (see Figure 1): cv1, P1, P2, T1, Xrod1, Fe, Fw, T1o, T2, L, Xs, Tso. For more details, refer to Bartyš et al. (2006), Louro (2003) and Oliveira (2004).
3. FDI ARCHITECTURE AND DESIGN

FDI model-based approaches deal with the increased complexity of treating dynamic systems by subdividing the diagnosis problem in two stages - residual generation and residual evaluation. Pattern recognition approaches, however, deal simultaneously with information concerning input variables – operating points and dynamics – plus the behaviour of output variables, for normal and faulty cases. Also, pattern recognition methods facilitate the design of hierarchic diagnosis approaches, with the advantage that:

- different subsets of behaviours can be treated at different levels; thus, the diagnosis of easily isolable faults is not compromised by a more complex global analysis, and even a classifier with a poorer performance may fail to separate overlapping classes, but perform satisfactorily on subsets of easily separable classes;
- moreover, by diving the overall problem into smaller, more tractable problems, it is likely to reduce the computational effort involved;
- finally, isolating the problematic groups of classes allows the design of specific feature extraction steps, enhancing the quality of information used in making the final decision.

It was also considered that the same fault may reveal very different effects in incipient mode (compared to abrupt mode), given the longer time that stabilization mechanisms have to adjust the dynamics. Therefore, diagnosis hierarchies for abrupt and incipient faults are designed separately. In this paper, only abrupt faults are addressed.

To investigate the various behaviours, looking for a proper hierarchy of diagnosis, a recent method that enables the creation of a hierarchic set of Self-organizing maps (Growing Hierarchical SOM – GHSOM, (Dittenbach et al., 2000) is investigated. A software package implementing GHSOM (Pampalk et al., 2001) is used, and an automatic process is devised to analyse the results:

- each node in each map is inspected, from lower to upper levels, looking for labels mapped together in the same node;
- labels found together in the same node generate a subset of grouped classes;
- new subsets are searched for at least one label in common with previous subsets, and the newly found subsets are preferably joined with equally small subsets, while nodes grouping a larger number of classes generate a new subset.

Finally, for each subset, the fraction of points of each label is calculated (relative to the total number of points in the subset, and relative to the total number of points of that class) and used to determine the relevant groups of classes. The final hierarchy for abrupt faults is described in Figure 2.

Regarding pre-processing steps, notice that part of the burden that is placed on the modelling task, when using model-based approaches, can be considered to fall on the feature extraction stage, when using pattern recognition approaches. Here, each monitored variable is first normalized to range [-1,1]. Next, feature extraction mechanisms for FDI have two concerns: retaining information about temporal evolution of the variables, and retaining relevant information from all variables, while reducing the final number of features.

Feature extraction steps are designed in two stages:

- extraction of temporal information from each monitored variable, individually:
  - for this purpose Karhunen-Loève transforms (PCA) are used, applied to 15 consecutive time samples (a 15s time window) of the variable, and the first 2 transformed features are retained; studies suggest (Milner, 1996) that this transform provides more informative features than typical temporal transforms, as evaluated for classification purposes;
- dimensionality reduction of the data space, constituted by the set of temporal transforms of each variable (at each stage of the hierarchy):
  - for this task, both non-discriminative and discriminative methods are investigated:
    - for each subset of faults, Curvilinear Component Analysis (CCA) was initially selected, to preserve
general information for identification (determining fault intensity); it was compared, in each case, with PCA, in terms of class separability (using Silhouette Index); Linear Discriminant Analysis, which focuses only on features relevant for fault isolation, was also investigated.

Special attention is paid to 2nd temporally transformed variables, whose behaviour is related to transient symptoms and similar to that of 1st derivatives; thus, they are treated separately in subsequent feature transform stages, in an attempt to approximate the advantages of the simplified residual evaluation of model-based approaches.

PCA (Jolliffe, 1986) and LDA (Duda, 1983, Fukunaga, 2000) are well-known feature extraction methods. PCA is an attempt to approximate the advantages of the simplified optimal linear dimension reduction technique (in the mean-square sense); LDA finds the linear projection that maximizes the dispersion of data within classes. CCA (Demartines al., 1997, Lee, 2000, 2002), on the contrary, is relatively recent. It may be seen as a non-linear generalization of PCA, but is developed in different steps:

PCAs criterion similar to (1), which locates the point relative to the transformed space by applying an optimization and moving all other $y_j$ around, which yields faster optimization. Any point in the original space can be mapped to the transformed space by applying an optimization criterion similar to (1), which locates the point relative to the prototypes. The implementation of CCA found in SOM Toolbox 2.0 (Alhoniemi et al., 1999) was used.

Once proper features are selected, a RBFs classifier is designed for each subset of faults. The local response characteristics RBFs are particularly interesting in the frame of a hierarchic structure, in which classifiers should not respond significantly to data outside its training subset. To reduce the computational cost, the training strategy was developed in different steps:

- Number of basis functions: select the number of basis functions (centres) according to:
  $$N_c = \min\{10 \times \text{(n. of states)} \times \text{(n. of dimensions)}\}, \quad 200$$
- Position of centres: select the position of centres using Magnetic Neural Gas (supervised vector quantization algorithm, Bogacz et al., 1997).
- Widths of basis functions: select the widths of each basis function independently. The basic idea is, for each centre (i), to find data mapped to a vicinity of k neurons (in the sense that data is mapped to the closest neuron); class selection is performed for all possible values of k, and the best one is chosen.

where $X_{ij}$ is the distance between features vectors $i$ and $j$ in the input space, $Y_{ij}$ is the distance between features vectors $i$ and $j$ in the transformed space, the neighbourhood function is a bounded and monotonically decreasing function, defined in the output space, and the neighbourhood parameter controls the area of influence. Instead of standard stochastic gradient descent, the adjustment is done by pinning each prototype $y_i$ and moving all other $y_j$ around, which yields faster optimization. Any point in the original space can be mapped to the transformed space by applying an optimization criterion similar to (1), which locates the point relative to the prototypes. The implementation of CCA found in SOM Toolbox 2.0 (Alhoniemi et al., 1999) was used.

Once proper features are selected, a RBFs classifier is designed for each subset of faults. The local response characteristics RBFs are particularly interesting in the frame of a hierarchic structure, in which classifiers should not respond significantly to data outside its training subset. To reduce the computational cost, the training strategy was developed in different steps:

- Number of basis functions: select the number of basis functions (centres) according to:
  $$N_c = \min\{10 \times \text{(n. of states)} \times \text{(n. of dimensions)}\}, \quad 200$$
- Position of centres: select the position of centres using Magnetic Neural Gas (supervised vector quantization algorithm, Bogacz et al., 1997).
- Widths of basis functions: select the widths of each basis function independently. The basic idea is, for each centre (i), to find data mapped to a vicinity of k neurons (in the sense that data is mapped to the closest neuron); class selection is performed for all possible values of k, and the best one is chosen.

$$E = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} (X_{ij} - Y_{ij})^2 F(Y_{ij}, \lambda_j),$$

$$F(Y_{ij}, \lambda_j) = \text{neighbourhood function} \quad (1)$$

$$\lambda_j = \text{neighbourhood parameter}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} (X_{ij} - Y_{ij})^2 F(Y_{ij}, \lambda_j),$$

$$F(Y_{ij}, \lambda_j) = \text{neighbourhood function} \quad (1)$$

$$\lambda_j = \text{neighbourhood parameter}$$

Fig. 2. Diagnosis hierarchy found by analysing GHSOM mapping
information is introduced by the ratio of each label mapped to the centre of interest \( r_u,i \); separate co-variance matrices are estimated for each class, and widths are calculated using the co-variance matrices of each class \( u \) weighted by the respective ratio \( r_u,i \).

- Output weights: train the output layer weights using ridge regression \( (rbf2 \) toolbox, by Orr, 1999); a number of trial values for overall rescaling of widths, and initial guesses for the regularization parameter are tested;
- Generalized Cross-Validation (GCV) is the model-selection criteria used, and a bias unit is introduced.

The decisions of an upper level in the hierarchy can be used to disregard invalid decisions of sub-classifiers on data outside their scope. For decisions at root level, an additional measure of the confidence level is used: a decision concerning a test pattern located far from training data should have a lower degree of confidence than one located close to training patterns; the average response of hidden units in the RBF at root level decision is used for this purpose.

Three different fault signals are used for training the classifiers, depending on their position on the diagnosis hierarchy (see Figure 2):

- a state signal - is 1 whenever the fault (or normal state) is the current condition, and 0 when it is not;
- a transient signal - the transient signal is defined only for faulty states, as 1 in the first 15s of the fault, 0.3 in the remaining time that the fault is present, and 0 whenever the fault is not active,
- a strength signal - this signal corresponds to the strength or intensity of the fault. The same value used in simulations is used for the signal.

In order to retain the important information conveyed by transient detection, a fourth signal is defined (based on the transient signal), as:

- persistent transient signal

\[
PTS(t) = \max (TS(t), TS(t-1))
\]

where TS(t) is the transient signal at time t and PTS(t) is the persistent transient signal at time t.

These signals are used to train a fuzzy system for each fault, aggregating all results for that fault. Fuzzy systems are selected for their ability to perform the type of inferences necessary to combine the detection of initial transients and of developed faults, in a persistent fault signal, with a confidence level interpretation and strong generalization ability. The Fuzzy Modelling and Identification Toolbox (Babuska, 1998) was used to design the fuzzy systems. All signals relative to the fault (in all levels of the hierarchy) are used as inputs to the fuzzy system. Each state signal is multiplied by the correspondent state signal in the immediately upper level. Any other state and transient signals from the same final classifier of the fault are also fed as inputs to the fuzzy system.

The fuzzy models return two signals:

- a state signal, as to the fault being active;
- an intensity signal, which attempts to estimate the severity (or strength) of the fault.

Finally, the state signal of each fuzzy aggregator is converted to binary signal by applying a threshold, set by an optimization procedure. The goal is to maximize the true detection/isolation signals, at expenses of some amount of false alarms, if necessary. This is represented by the objective function (2) (distinguishing between false detection and false isolation):

\[
J(T) = \alpha \cdot fd(T) - \gamma \cdot fi(T) - \beta \cdot t(T)
\]

where \( fd(T) \) is the number of false detections (as a function of the threshold), \( fi(T) \) is the number of false isolations (as a function of the threshold) and \( \alpha, \beta, \gamma \) are coefficients that may be used to place different penalties on each term. For a real application, false detections are considered far more disturbing than false isolations. Accordingly, the above coefficients are set so as to force the number of false detection alarms to remain below 2% of true diagnosis (in training data). A direct search method (fminsearch, provided in MatLab's Optimization Toolbox, implementing the simplex method) is used to perform the optimization.

The overall FDI architecture just described is schematically represented in fig. 3 at the end of the paper.

4. DISCUSSION OF RESULTS

The performance of the designed Fault Diagnosis system is assessed using the indices defined by DAMADICS (Barty et al., 2006): true detection rate \( ti \), false detection rate \( fi \), true isolation rate \( ri \), false isolation rate \( fis \), detection time \( ti \), isolation time \( ti \), and diagnosis accuracy \( di \).

Two test simulations are used for each fault intensity defined for benchmark (0.25, 0.50, 0.75), in a total of 6 tests per fault. An overall summary of results is presented in Table 1.

Performing a step-by-step analysis of the results, the following remarks are due:

- The hierarchic structure is consistent with the results and with the description of faults.
- Karhunen-Loève transform provides an efficient method for extracting temporally informative features from a number of time samples.
- In most subsets of the hierarchy, CCA provides better unfolding of the feature vectors, in equal or less number of transformed features than PCA (in the training set); PCA performs better when the effect of faults is essentially linear.
- However, CCA fails to properly generalize to training patterns, in terms of class separation; this is found to be related to the disproportional range of some variables, compared to their relevance in defining set-point or fault behaviour. PCA is less severely affected because, unlike CCA, it extracts one variable at the time, so that a single variable does not compromise the entire projection space.
- The use of a discriminative feature extraction method – LDA – simplifies the basic classification task (detection/isolation), but reduces the quality of information for the identification task, and reveals clear weaknesses in some specific cases, discussed ahead. The results presented in Table 1 are obtained with this feature extraction method.
- Regarding the final thresholds, they are set by approaching the upper limit rendering equal performance. However, preliminary results show that fault signals \( f1, f10 \) and \( f16 \) are responsible for about 90% of all false alarms. Due to specific visibility conditions, these thresholds can be set to higher...
Table 1 – Overall results by fault signal (leaving out simulations with lowest degree of confidence)

<table>
<thead>
<tr>
<th>Fault signal</th>
<th>r_{id}</th>
<th>r_{fd}</th>
<th>r_{di}</th>
<th>r_{si}</th>
<th>t_{d} (s)</th>
<th>t_{i} (s)</th>
<th>r_{id} / (overall r_{id})</th>
<th>r_{si} / (overall r_{si})</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1</td>
<td>0.031</td>
<td>0.001</td>
<td>0.000</td>
<td>0.001</td>
<td>42</td>
<td>NaN</td>
<td>0.54</td>
<td>0.00</td>
</tr>
<tr>
<td>f2</td>
<td>0.988</td>
<td>0.009</td>
<td>0.988</td>
<td>0.044</td>
<td>7</td>
<td>7</td>
<td>17.11</td>
<td>220.46</td>
</tr>
<tr>
<td>f7</td>
<td>0.998</td>
<td>0.000</td>
<td>0.998</td>
<td>0.033</td>
<td>1</td>
<td>1</td>
<td>17.30</td>
<td>222.81</td>
</tr>
<tr>
<td>f8</td>
<td>0.121</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>67</td>
<td>NaN</td>
<td>2.10</td>
<td>0.00</td>
</tr>
<tr>
<td>f10</td>
<td>0.434</td>
<td>0.017</td>
<td>0.397</td>
<td>0.033</td>
<td>3</td>
<td>3</td>
<td>7.52</td>
<td>88.67</td>
</tr>
<tr>
<td>f11</td>
<td>0.409</td>
<td>0.000</td>
<td>0.001</td>
<td>0.000</td>
<td>12</td>
<td>12</td>
<td>7.08</td>
<td>0.19</td>
</tr>
<tr>
<td>f12</td>
<td>0.111</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>214</td>
<td>NaN</td>
<td>1.92</td>
<td>0.00</td>
</tr>
<tr>
<td>f13</td>
<td>0.880</td>
<td>0.001</td>
<td>0.576</td>
<td>0.025</td>
<td>3</td>
<td>6</td>
<td>15.25</td>
<td>128.58</td>
</tr>
<tr>
<td>f15</td>
<td>0.373</td>
<td>0.004</td>
<td>0.288</td>
<td>0.015</td>
<td>22</td>
<td>92</td>
<td>6.47</td>
<td>64.38</td>
</tr>
<tr>
<td>f16</td>
<td>0.994</td>
<td>0.027</td>
<td>0.080</td>
<td>0.010</td>
<td>128</td>
<td>8</td>
<td>1.64</td>
<td>17.80</td>
</tr>
<tr>
<td>f17</td>
<td>0.998</td>
<td>0.000</td>
<td>0.997</td>
<td>0.040</td>
<td>1</td>
<td>2</td>
<td>17.28</td>
<td>222.44</td>
</tr>
<tr>
<td>f18</td>
<td>0.834</td>
<td>0.000</td>
<td>0.828</td>
<td>0.046</td>
<td>3</td>
<td>4</td>
<td>14.44</td>
<td>184.69</td>
</tr>
<tr>
<td>f19</td>
<td>0.947</td>
<td>0.000</td>
<td>0.946</td>
<td>0.038</td>
<td>2</td>
<td>3</td>
<td>16.40</td>
<td>211.19</td>
</tr>
</tbody>
</table>

values, decreasing false diagnosis in test data, and not affecting severely the detection of visible fault scenarios. A somewhat arbitrary increase of 20% is imposed on these thresholds.

- Finally, some simulations have an extremely high rate of false detections (r_{id}); the highest ones occur in simulations with very low degree of confidence. In these cases, the temperature at the valve (T_{i}), is much lower than those found in training data. A logical explanation is that the monitored data was collected while some actual fault was occurring in the real plant. The three simulations with lowest degree of confidence - simulations f1, f7, f8, with medium intensity - are disregarded in this analysis.

In a global analysis, faults f8 and f12 are, as expected by their theoretical behaviour, undetectable. In contrast, for faults f2, f7, f17, f13, f18 and f19 good results are achieved, even using a very simple feature extraction method, which revealed severe limitations in more difficult fault scenarios. These results are in agreement with the designed hierarchy and with fault behaviour.

Faults f1 and f16 are theoretically detectable in certain conditions, but the diagnosis system fails to achieve meaningful detection (even in the cases of clear fault visibility); the limitations of LDA have a large part of responsibility in the way that conditioned visibility of these faults affect final performance.

Fault f11 is detectable, but not isolable to the designed diagnosis system (possibly due to insufficient training data for proper generalization).

Fault f10 is undetectable for low fault intensities, but quickly detected and isolated for medium and high.

Fault f15 is correctly detected and isolated in all scenarios, although with relatively low detection and isolation rates, which can be attributed, both to actual ambiguity in temporary fault effects, and to limitations of the feature extraction method (LDA).

Finally, faults f16, f10, and f15 are responsible (in this order) by most false alarms, accounting for nearly 80% of false detections. These faults are separated, at root level of the hierarchy, to a subset of faults difficult to detect (grouped with normal state).

5. CONCLUSIONS AND FUTURE WORK

In this paper, a hierarchical FDI system is proposed based on pattern recognition approaches. Growing Hierarchical Self-Organizing Maps were successfully employed in the design of a diagnosis hierarchy, using an automated analysis of GHSOM mapping. Concerning feature extraction steps, although discriminative methods simplify the requirements for generalization, non-discriminative methods present more promising characteristics for the type of complex classification/identification task involved in Fault Diagnosis. A possibly advantageous compromise would be to use discriminative methods at the top level(s) of the hierarchy, and non-discriminative methods for more detailed classification stages. As to extracting temporally informative features, Karhunen-Loève transforms (PCA) performed very well, with very little computational cost. Concerning the specific method more closely studied for feature (CCA), despite its failure in the specific conditions it was employed, several tests suggest that, with proper normalization methods, CCA may be able to extract quality features for the diagnosis of dynamic systems. However, further research will be required to successfully apply the method. Normalization was found to be of crucial importance when applying non-discriminative - and, particularly, topology preserving – feature extraction methods. The results of both Radial Basis Function Networks and Fuzzy Systems were consistent with their expected characteristics and presented no particular problems. Notice that, although purely pattern recognition methods were preferred here, the same strategies could perform better if applied as a second stage, after a model-based step of residual generation.

Finally, with this work we try to complement the extensive work of FDI systems design to deal with the DAMADICS benchmark actuator valve problem by using a free model.
design technique, the pattern recognition approach. The overall results obtained with the proposed pattern recognition FDI system prove to be comparable with the other model-based FDI techniques.

REFERENCES


![Diagram](image-url)

**Fig. 3.** Diagnosis architecture diagram